International Journal of Foundations of Computer Science

Languages vs. ω-Languages in Regular Infinite Games*

NAMIT CHATURVEDI† and JÖRG OLSCHEWSKI‡ and WOLFGANG THOMAS
Lehrstuhl für Informatik 7, RWTH Aachen University
52056 Aachen, Germany
{chaturvedi, olschewski, thomas}@automata.rwth-aachen.de
http://www.automata.rwth-aachen.de

Received (Day Month Year)
Accepted (Day Month Year)
Communicated by (xxxxxxxxxx)

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 Selivanov (2007) and Rabinovich et al. (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 strategy representations strictly exceed $\mathcal{K}$.

Keywords: Regular Languages; star-free languages; infinite games, winning strategies.

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. [8, 21]), 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, where the determinis-

*A preliminary version of this paper appeared in the proceedings of the 15th Conference on Developments in Language Theory (DLT 2011) [2].
†Supported by DFG Research Training Group (DFG-Graduiertenkolleg) 1298 AlgoSyn.
‡Supported by ESF-Eurocores LogICCC project GASICS.

tic model of automata is needed. Another canonical transformation of $*$-languages into ω-languages is the *extension* of words of a $*$-language U:

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

Boolean combinations of such languages with regular U are recognized by deterministic weak Muller automata (also known as Staiger-Wagner automata [16]). We write BC(lim(REG)), respectively BC(ext(REG)), for the Boolean combinations of sets $\lim(U)$, respectively $\text{ext}(U)$, 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 $\text{ext}(U)$ 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 $\text{BC}(\lim(\mathcal{K}))$ and $\text{BC}(\text{ext}(\mathcal{K}))$, 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 are "winning conditions", and $*$-languages represent "winning strategies". We shall study the question whether games with a winning condition in classes such as $\text{BC}(\text{ext}(\mathcal{K}))$ or $\text{BC}(\lim(\mathcal{K}))$ can be "solved" with winning strategies representable in $\mathcal{K}$.

If the formalism of logic is used for specification, the transition to definitions of winning conditions can be accomplished without operators such as ext or lim; indeed, the same logic can be used for defining ω-languages and $*$-languages. We address the connection between definability of winning conditions and winning strategies in this simple setting as well (see Section 6).

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 $\text{BC}(\text{ext}(\mathcal{K}))$ are referred to as *weak games* while those whose winning conditions belong to $\text{BC}(\lim(\mathcal{K}))$ 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] (see also [5, 21]) says that for a winning condition defined by any 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*. An analogous result for the class SF of star-free languages was shown in [12] (see also [13]) and later, by a different method, also in [11]: 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 one, games with winning conditions in classes $\mathrm{BC}(\mathrm{ext}(\mathrm{DD}_1))$ and $\mathrm{BC}(\lim(\mathrm{DD}_1))$ are, in general, not determined 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 $w_0\Sigma^*w_1 \dots \Sigma^*w_n$, 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. Section 6 couples winning conditions and winning strategies more tightly with the help of a suitable logic. We conclude with some open questions and perspectives.

2. Technical Preliminaries

2.1. *Languages, Automata, Games*

We use standard notations [6] regarding languages and automata. As a model of automata with output we use Moore machines, which transform words over an alphabet Σ into words over 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 run over α belongs to $\mathcal{F}$; for Muller automata one refers instead to the set of states visited infinitely often.

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 $\alpha = (a_i, x_i)_{i\in\mathbb{N}} \in \Sigma^\omega$ 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 $\tau(\alpha[0,i))(a_i) = x_i$. For two strategies σ and τ there is a (uniquely determined) word, or *play*, $\alpha(\sigma,\tau)$ that is consistent with both σ and τ. This play is called *winning for Player 2* if $\alpha(\sigma,\tau) \in L$.

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 $K_{a\mapsto x} = \{w \mid \tau(w)(a) = x\}$ is in $\mathcal{K}$ for every $(a,x) \in \Sigma$. Given a game L and a class $\mathcal{K}$ of $*$-languages, we say that *L is determined with winning strategies in $\mathcal{K}$* if one of the players has a winning strategy in $\mathcal{K}$. 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 in 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 $\mathrm{BC}(\mathrm{ext}(\mathcal{K}))$, where $\mathcal{K} \subseteq \mathrm{REG}$, it is known that SW-automata (over $\Sigma := \Sigma_1 \times \Sigma_2$) can be used; for ω-languages in a class $\mathrm{BC}(\lim(\mathcal{K}))$ we may take Muller automata.

In the literature [5], two-player games are usually considered 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 expanding 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, where in a single move from a node, 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.

In the rest of this paper, we refer to some established results about games [5]. For a concise presentation, see [22]. 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 each 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". The former two correspond to games L of the form $\mathrm{ext}(K)$ or $\lim(K)$ respectively, for some $*$-language K. The latter *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 parity 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 parity condition while expanding the game graph by an extra "memory component" [22]. 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 $h \in \{1, \ldots, k\}$. Over the expanded state-space it suffices to satisfy the mentioned parity condition.

2.2. *Regular Expressions and Classes of Star-free Languages*

In the subsequent definitions we recall some basic subclasses of the star-free languages; for more background see [9, 18].

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 $a_1, a_2, \ldots, a_n \in \Sigma$. We denote the class of piecewise testable languages by PT, and the class of *positive* Boolean combinations of basic PT-sets (in which only $\cup$ and $\cap$ are used) 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 [3], 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 $K_1 \cdot K_2 \cdot \ldots \cdot K_\ell$ (over a given alphabet Σ) with $K_1, \ldots, K_\ell \in \mathrm{DD}_n$. As a special case let us mention the languages of *dot-depth one*; they are the Boolean combinations of *basic* DD_1-*sets* $w_0 \Sigma^* w_1 \Sigma^* \cdots w_{n-1} \Sigma^* w_n$ where $w_0, w_1, \ldots, w_n \in \Sigma^*$. Also note that $\mathrm{PT} \subsetneq \mathrm{DD}_1$. In analogy to the class pos-PT we define pos-DD_1 as the class of positive Boolean combinations of basic DD_1-sets.

For $|\Sigma| > 1$, the dot-depth hierarchy is strict, and it exhausts the class SF of star-free languages. The study of these classes is based on corresponding congruences on Σ^*. We recall these congruences for the case of languages of dot-depth one.

For $k, m \in \mathbb{N}$ and an m-tuple $\nu = (w_1, \ldots, w_m), |w_i| = k + 1$, we say that ν *appears* in $u \in \Sigma^*$ if for $1 \leq i \leq m,\ u$ can be written as $u = u_i w_i v_i$ with suitable words u_i, v_i such that $1 \leq i < j \leq m$ implies $|u_i| < |u_j|$. We say that ν *appears* in $\alpha \in \Sigma^\omega$ if for $1 \leq i \leq m,\ \alpha$ can be written as $u = u_i w_i \alpha_i$ with suitable words

u_i, α_i such that $1 \leq i < j \leq m$ 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 $u, v \in \Sigma^*$ 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 $m, k \in \mathbb{N}$* [14]. In the definition of $\sim_{m,k}$ we refer to possibly overlapping infixes; this does not affect statement $(*)$.

Closely related with the dot-depth hierarchy of star-free languages is the Straubing-Thérien hierarchy [9]. We denote the class of languages at level n of this hierarchy as ST_n. This hierarchy is recursively defined over a given alphabet Σ, by $\mathrm{ST}_0 := \{\emptyset, \Sigma^*\}$, and ST_{n+1} as the class of all Boolean combinations of languages $K_0 a_1 K_1 \dots a_n K_n$ where $a_i \in \Sigma$ for $i = 1, \dots, n$ and $K_i \in \mathrm{ST}_n$ for $i = 0, \dots, n$. In particular, we have $\mathrm{ST}_1 = \mathrm{PT}$. Analogous to the dot-depth hierarchy, the Straubing-Thérien hierarchy is strict, infinite, and exhausts the class of all star-free languages. It is known that for all $n \in \mathbb{N}, \mathrm{ST}_n \subsetneq \mathrm{DD}_n$, and for all $n > 0$, $\mathrm{DD}_n \subsetneq \mathrm{ST}_{n+1}$ [9].

Henceforth, we prefer to speak of the language class PT instead of ST_1 to emphasize the fact that is also an independently interesting subclass of DD_1 languages.

2.3. *Logic and Classes of Star-free Languages*

Many of the classes of special regular languages have natural characterizations in the framework of logic (see e.g. [18]). The starting point is the classical result of Büchi, Elgot, and Trakhtenbrot on the expressive equivalence of finite automata and monadic second-order logic. We deal here with sublogics for classes of star-free languages; in this case we have an equivalence with sublogics of first-order logic over finite words. A word $w = a_1 \dots a_n \in \Sigma^*$ is then identified with a finite linear ordering $(\mathrm{dom}(w), <, (P_a)_{a\in\Sigma})$ where $\mathrm{dom}(w) = \{1, \dots, n\}$, $<$ is the standard ordering on this set, and $P_a = \{i \in \mathrm{dom}(w) \mid a_i = a\}$. In an extended signature, over nonempty words, we also include the elements min and max (the first and last element of $\mathrm{dom}(w)$) and the successor and predecessor functions suc and pre respectively, such that $\mathrm{suc}(\max) = \max$ and $\mathrm{pre}(\min) = \min$. The corresponding first-order logic (with equality $=$) is built up as usual with the symbols $<$, P_a, and in the extended signature also with min, max, suc, and pre. We refer to these logics by $\mathrm{FO}[<]$ and $\mathrm{FO}[<, \mathrm{suc}, \mathrm{pre}, \min, \max]$ respectively.

We recall that a language $K \subseteq \Sigma^+$ belongs to DD_n iff it can be defined by a first-order sentence that is a Boolean combination of "Σ_n-sentences" (first-order sentences in prenex normal form with n alternating quantifier blocks starting with an existential block) in $\mathrm{FO}[<, \mathrm{suc}, \mathrm{pre}, \min, \max]$ [20]. Similarly, a language $K \subseteq \Sigma^*$ belongs to ST_n iff it is definable by a Σ_n-sentence in $\mathrm{FO}[<]$ [7].

3. Winning Strategies in Restricted Weak Games

We start with games in $\mathrm{BC}(\mathrm{ext}(\text{pos-DD}_1))$ which coincides with the class of Boolean combinations of sets $\mathrm{ext}(K)$ where K is a basic DD_1-set, or in other words: Boolean combinations of sets $w_0\Sigma^* w_1 \Sigma^* \cdots w_{n-1}\Sigma^* w_n \Sigma^\omega$.

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, analogous to the characterization presented in Section 2.2, we have: *An ω-language $L \subseteq \Sigma^\omega$ is in $\mathrm{BC}(\mathrm{ext}(\text{pos-DD}_1))$ iff it can be written as a union of $\Sigma^\omega/{\sim_{m,k}}$ equivalence classes for some $m, k \in \mathbb{N}$.* We rely on this observation when we present the following result.

Theorem 1. *Games in $\mathrm{BC}(\mathrm{ext}(\text{pos-DD}_1))$ are determined with winning strategies in DD_1.*

Proof. We can write an ω-language L in $\mathrm{BC}(\mathrm{ext}(\text{pos-DD}_1))$ 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 $(a, x) \in \Sigma$, we have edges from $[w]$ to $[w(a, x)]$. Note that this relation 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 starting 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 prefix of w of the same length and $\mu_{m,k}(\alpha) = \mu_{m,k}(w)$; we set $\chi([w]) = 2\cdot|\mu_{m,k}(w)|-1$ 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).

By a well-known result on parity games by Emerson and Jutla, and Mostowski (see [5]), the parity game is determined, and the winning player has a positional winning strategy from the starting node 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 $\sigma(w) = \lambda([w])$. 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; basic piecewise testable languages are languages of the form $\Sigma^* a_1 \Sigma^* a_2 \ldots a_n \Sigma^*$. 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 [15] 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 it only retain edges of $\mathcal{A}$ that are labeled with elements of Γ. Note that $G(\mathcal{A}, \Gamma)$ may consist of several unconnected components. Moreover, order all states in Q by letting $p \leq q$ iff q is reachable from p.

Proposition 2 ([15]) *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 $L \in \mathrm{BC(ext(pos\text{-}PT))}$, there exists a regular language $K \in \mathrm{PT}$ such that $L = \lim(K)$. This is shown easily by induction over Boolean combinations (cf. [10]). The minimal DFA $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ accepting K can be considered as an ω-automaton accepting L. We thus obtain a game graph with a Büchi winning condition. Since Büchi games are determined with positional winning strategies (see e.g. [5]), 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 the set of 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 BC(ext(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 property of the associated transition graphs. For games in BC(ext(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 [17] 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 that for pos-PT.

4. Winning Strategies in Weak Games

So far we considered winning specifications that were constructed only from positive combinations of $*$-languages. In this section, we study the general setting of weak games. This means that for every class of $*$-languages, we allow arbitrary Boolean combinations of languages from this class as seeds for winning specifications.

Theorem 4. *There are games in* BC(ext(PT))*, and therefore in* BC(ext(DD_1)) *and in* BC(ext(ST_1))*, that do not admit* DD_1 *winning strategies.*

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

$$\big[\alpha \in \mathrm{ext}(K_d) \to \big(\alpha \in \mathrm{ext}(K_1) \leftrightarrow \alpha \in \mathrm{ext}(K_2)\big)\big] \wedge \bigwedge_{x \in \{a,b,c\}} \alpha \in \overline{\mathrm{ext}(K_x)}$$

All the $*$-languages above are in PT, so L is in BC(ext(PT)). Since PT $=$ $ST_1 \subsetneq DD_1$, it follows that L is also in BC(ext(DD_1)) and in BC(ext(ST_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, $K_{d1} := \{w \mid \tau(w)(d) = 1\}$ is a dot-depth one language as a finite union of K_{θ_i} languages. So it is a finite union of equivalence classes $[w_i]_{\sim_{m,k}}$.

Let $w \in \{a, b, c\}^*$ be a word in which all possible m-tuples of length $k + 1$ over the alphabet $\{a, b, c\}$ appear. Such a word exists, since there are only finitely many of such tuples, and concatenating words w_1 and w_2 with sets T_1 and T_2 of tuples yields a word containing $T_1 \cup T_2$. Let Player 1 play a strategy σ_1 that chooses $a^k b \cdot w \cdot d \cdot a^\omega$. 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_1. So the word $a^k b w \times \{0\}$ is in K_{d1}.

Now let Player 1 play the strategy σ_2 which chooses $a^k c \cdot w \cdot d \cdot a^\omega$. The word $a^k c \cdot w$ contains all possible m-tuples of length $k+1$ over the alphabet $\{a,b,c\}$, as well. Then we have $a^k bw \sim_{m,k} a^k cw$ and thus $a^k cw \times \{0\} \in K_{d1}$. 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. *For each $i \in \mathbb{N}$, games in* $\mathrm{BC}(\mathrm{ext}(\mathrm{DD}_i))$ *are determined with winning strategies in* DD_{i+1}.

Proof. Any regular language $K_\ell \in \mathrm{DD}_i$ is accepted by a DFA $A_\ell = (Q, \Sigma, q_0, \delta, F)$ such that for every $q \in Q$, the language $[w]_q \coloneqq \{w \in \Sigma^* \mid q_0 \xrightarrow{w} q\}$ is in DD_i. Then, preserving the transition structure of A_ℓ, we obtain a Staiger-Wagner automaton $\mathcal{A}_\ell$ which accepts the ω-language $L_\ell = \mathrm{ext}(K_\ell)$. For any Boolean combination of languages $\mathrm{ext}(K_\ell)$, we obtain a Staiger-Wagner automaton $\mathcal{A}$ by constructing the product automaton from the automata for $\mathrm{ext}(K_\ell)$. Since DD_i is closed under Boolean combinations, for $\mathcal{A}$, it again holds that $[w]_q \in \mathrm{DD}_i$ for each state q.

Therefore, for a winning condition $L \in \mathrm{BC}(\mathrm{ext}(\mathrm{DD}_i))$, we obtain 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, it will immediately follow that 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.

To reason about $K_{(q,R)}$, it is convenient to apply the logical characterization of DD_i-languages as mentioned in Section 2.3. 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 in $\mathrm{FO}[<, \mathrm{suc}, \mathrm{pre}, \min, \max]$. We now have to express the restriction that a play prefix leads to state q *and* to the AR $R \subseteq Q$ for the state space Q. This is formalized by the sentence

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

where, for every $r \in Q$, the formula $\psi_r[x]$ is obtained from ψ_r by *relativisation* with respect to the segment $[0,x]$ (see [4]). Formally this means that we introduce x as a new free variable and rewrite every subformula of ψ_r of the form $\exists y \lambda(y)$ as $\exists y (y \leq x \wedge \lambda(y))$, every subformula of the form $\forall y \lambda(y)$ as $\forall y (y \leq x \rightarrow \lambda(y))$, and every occurrence of max as x. Thus $\exists x \psi_r[x]$ describes all words that have a prefix that satisfies ψ_r.

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}. □

Theorem 6. *For each $i \in \mathbb{N}$, games in* $\mathrm{BC}(\mathrm{ext}(\mathrm{ST}_i))$ *are determined with winning strategies in* ST_{i+1}.

Proof. Analogously to Theorem 5, for a language $L \in \mathrm{BC}(\mathrm{ext}(\mathrm{ST}_i))$, we can obtain a game graph with Staiger-Wagner winning condition where every node is a ST_i equivalence class (and hence a $*$-language in ST_i). For each node, this language can be described by a Boolean combination of Σ_i-sentences in FO[<]. The remainder of the proof is similar to that of Theorem 5. □

5. Winning Strategies in Strong Games

In [11, 12] the general result by Büchi and Landweber was refined to the class of star-free games, i.e., games with star-free winning conditions are determined with star-free winning strategies. In this section, we refine this result even further, as the games considered here constitute the class of all star-free games.

Theorem 7. *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 [10]). The remainder of this proof is identical to that of Theorem 3. □

Theorem 8. *There are games in the class* $\mathrm{BC}(\lim(\mathrm{DD}_0))$ *that do not admit* DD_1 *winning strategies.*

Proof. Let $\Sigma := \{a, b\} \times \{0, 1\}$. Consider DD_0 languages $K_{a,0} := \Sigma^*(a, 0)$, $K_{a,1} := \Sigma^*(a, 1)$, $K_{b,0} := \Sigma^*(b, 0)$, and $K_{b,1} := \Sigma^*(b, 1)$. The winning condition is an ω-language

$$L := \overline{\lim(K_{a,0})} \cap \left[\overline{\lim(K_{b,0})} \cup \lim(K_{b,1})\right] \cap \left[\lim(K_{b,0}) \cup \overline{\lim(K_{b,1})}\right],$$

which states that a word $\alpha \in \Sigma^\omega$ belongs to L if and only if, firstly, α contains only finitely many occurrences of $(a, 0)$, and, secondly, $(b, 0)$ occurs infinitely often in α if and only if $(b, 1)$ does. Thus $L \in \mathrm{BC}(\lim(\mathrm{DD}_0))$.

Player 2 has a winning strategy which can be described in two parts as follows. First, whenever Player 1 plays a, respond with 1. Second, remember the previous response to Player 1's choice of b, and flip that response if Player 1 chooses to play b again. This ensures that for any play $\alpha \in \Sigma^\omega$ consistent with this strategy, firstly $\alpha \notin \lim(K_{a,0})$, and secondly $(b, 0)$ and $(b, 1)$ appear alternately in α. Indeed, Player 2's strategy maps finite play prefixes to functions $\theta_0, \theta_1 \in \Theta$ where,

$$\theta_0 = \{a \mapsto 1, b \mapsto 0\}, K_{\theta_0} = \Sigma^*(b, 1) \cup \left(\Sigma^*(b, 1) \cdot \overline{\Sigma^*(b, 0)\Sigma^* \cup \Sigma^*(b, 1)\Sigma^*} \cdot (a, 1)\right)$$
$$\theta_1 = \{a \mapsto 1, b \mapsto 1\}, K_{\theta_1} = \Sigma^*(b, 0) \cup \left(\Sigma^*(b, 0) \cdot \overline{\Sigma^*(b, 0)\Sigma^* \cup \Sigma^*(b, 1)\Sigma^*} \cdot (a, 1)\right)$$

It is straightforward to verify that $K_{\theta_0}, K_{\theta_1} \in \mathrm{DD}_2$. Now we show that Player 2 has no DD_1 winning strategy. On the contrary, assume that Player 2 has such a strategy $\tau\colon \Sigma^* \to \Theta$. In particular, for $\theta \in \Theta$ there is a language $K_\theta \in \mathrm{DD}_1$ such that for every play prefix $u \in K_\theta, \tau(u) = \theta$, and K_θ is described by a finite set of m_θ-tuples of words of length $k_\theta+1$ for $m_\theta, k_\theta \in \mathbb{N}$. We define $k_\tau := \max\{k_\theta \mid \theta \in \Theta\}$.

For Player 1, we fix the strategy σ that involves playing ba^{k_τ} ad infinitum. Now we consider the play $\alpha \in \Sigma^\omega$ that is consistent with σ and τ. Since τ is a winning strategy, α must be winning for Player 2.

Since $\alpha \notin \lim(K_{a,0})$, there must be a smallest position $\ell_a \in \mathbb{N}$ such that for any $j \geq \ell_a$, $\alpha[\ell_a, j]$ does not contain the letter $(a, 0)$. For each $\theta \in \Theta$, since there are only finitely many m_θ-tuples of words of length $k_\theta + 1$, there exists a smallest position $\ell_\theta \in \mathbb{N}$ after which no new m_θ-tuple is seen in α. That is, the same tuples appear in the prefix $\alpha[0, j]$ for every $j \geq \ell_\theta$.

Finally, we define ℓ_τ as the smallest multiple of $k_\tau + 1$ greater than or equal to $\max\{\ell_a, \ell_\theta \mid \theta \in \Theta\}$, and consider the prefix $w_\tau := \alpha[0, \ell_\tau]$. Without loss of generality, we assume that $\tau(w_\tau) = \theta$ and $\theta(b) = 0$. It is easy to see that $w_\tau \sim_{m_\theta,k_\theta} w_\tau(b, 0)(a, 1)^{k_\tau} \sim_{m_\theta,k_\theta} w_\tau(b, 0)(a, 1)^{k_\tau}(b, 0)(a, 1)^{k_\tau} \cdots \in K_\theta$. Therefore, after w_τ, τ always responds to b's with 0's, leading to infinite occurrences of $(b, 0)$ but not of $(b, 1)$. This contradicts α being a winning play for Player 2. □

While one may construct strong games at any level of the dot-depth hierarchy such that there are no winning strategies at the same level, 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 9. *For each $i \in \mathbb{N}$, games in* $\mathrm{BC}(\lim(\mathrm{DD}_i))$ *are 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 L when we equip the graph constructed above with a Muller winning condition. As explained in Section 2.1, we transform this game graph via the LAR construction into 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 parity game graph, the winner has a positional winning strategy.

We know that 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 (cf. Section 2.3). Using 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 $R = (\mathit{perm}, h)$. Let the sentence φ_R express the fact that a play prefix has led to R. It is evident that $K_{(q,R)}$ is the set of words that satisfy the formula φ_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 pre(max) the most recent play prefixes are in r, s, q in this order. This can be described as

$$\begin{aligned}\varphi_R = \psi_q[\max] \wedge \psi_r[\mathrm{pre}(\max)] \\ \wedge \exists x, y, z\big(\max > x > y > z \wedge \psi_r[x] \wedge \psi_s[y] \wedge \psi_q[z] \\ \wedge \forall x_1(\max > x_1 > x \rightarrow \psi_r[x_1]) \\ \wedge \forall y_1(x > y_1 > y \rightarrow \psi_s[y_1])\big),\end{aligned}$$

where, for every $q \in Q$, the formula $\psi_q[x]$ is obtained from ψ_q by relativisation with respect to the segment $[0, x]$.

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

Theorem 10. *For each* $i \in \mathbb{N}$*, games in* $\mathrm{BC}(\lim(\mathrm{ST}_i))$ *are determined with winning strategies in* ST_{i+2}.

Proof. As with Theorem 6, for a language $L \in \mathrm{BC}(\lim(\mathrm{ST}_i))$, we start with constructing a game graph with Muller winning conditions where every node is a ST_i equivalence class. For each node, this language can be described by a Boolean combination of Σ_i-sentences in FO[<]. Then the sentence describing the LAR as in the proof of Theorem 9 can be written analogously, without the constant "max", as follows. Note that this constant can be defined in terms of $\leq$, and, for the sake of readability, we use variables x_{max} and x_{max-1} to stand in place of max and pre(max) respectively.

$$\begin{aligned}\varphi_R = \exists x_{\max}, x_{\max-1}\big(\forall y(y \leq x_{\max}) \wedge \forall y(y < x_{\max} \rightarrow y \leq x_{\max-1}) \\ \wedge \psi_q[x_{\max}] \wedge \psi_r[x_{\max-1}] \\ \wedge \exists x, y, z\big(x_{\max} > x > y > z \wedge \psi_r[x] \wedge \psi_s[y] \wedge \psi_q[z] \\ \wedge \forall x_1(x_{\max} > x_1 > x \rightarrow \psi_r[x_1]) \\ \wedge \forall y_1(x > y_1 > y \rightarrow \psi_s[y_1])\big)\big).\end{aligned}$$

This sentence is a Boolean combination of Σ_{i+2} sentences, and we infer the existence of a winning strategy in ST_{i+2} as in Theorem 9. □

6. On the Straubing-Thérien Hierarchy

The results of the previous sections referred to frameworks in which definitions of $*$-languages and of ω-languages are related via operators that provide a bridge, in particular, via the operators ext and lim.

In the framework of logic a more direct relation is possible if the signature is flexible enough to deal with both finite and infinite words. In the logic FO[<] one can use precisely the same formulas for defining $*$-languages and ω-languages; the only change involved is that of the format of models – in the first case as labeled finite orderings, in the second case as infinite labeled orderings. It is then unnecessary to invest an operator such as ext or lim as a connection between definitions of $*$-languages and ω-languages.

In the present section we directly relate definability of games and winning strategies in the logical setting, taking here the case of Boolean combinations of Σ_n-sentences in the Straubing-Thérien hierachy (i.e., Boolean combinations of first-order sentences in the signature with order and letter predicates, and with quantifier prefix of n quantifier blocks, starting with an existential block of quantifiers). The class of infinite games definable by a Boolean combination of this form sentences we denote by ST_n^ω. We treat these games by reducing them to a form that is available from the previous section.

More precisely, we show that each game in ST_n^ω can be presented as a game in $\mathrm{BC}(\lim(\mathrm{ST}_n))$ [cf. Theorem 13 below] and then apply Theorem 10. Since all proofs are routine adaptations of the classical case treated in [19], we do not present a proof with all details.

Theorem 11. *For $i \in \mathbb{N}$, games in ST_i^ω are determined with strategies in ST_{i+2}.*

To show the above theorem, we apply a variant of the main result of [19]. The core of the argument is an analysis of sentences in prenex normal form that have a (first-order) quantifier prefix

$$\mathrm{Q}\overline{x_1}\mathrm{Q}\overline{x_2}\ldots\mathrm{Q}\overline{x_n}$$

where Q is one of the quantifiers $\exists$, $\forall$ and $\overline{x_i}$ a tuple of variables say of length m_i. (We write here, for example, $\exists x, y, z$ as a shorthand for $\exists x \exists y \exists z$.) Call such a sentence a $(m_1, \ldots, m_n)$-sentence or, in short, an $\overline{m}$-sentence.

In the classical framework of Ehrenfeucht-Fraissé games, one distinguishes models by the m-equivalence $\equiv_m$, which holds between two models if they cannot be distinguished by sentences of quantifier-depth m. In our setting, models are labeled linear orderings with the signature $\{<, (P_a)_{a\in\Sigma}\}$. We use here a slightly different equivalence, namely for a tuple $\overline{m} = (m_1, \ldots, m_n)$ the equivalence $\equiv_{\overline{m}}$ holds between two models if they cannot be distinguished by $\overline{m}$-sentences.

The characterization of $\equiv_{\overline{m}}$ by Ehrenfeucht-Fraïssé games is now obtained in the obvious way, using – for $\overline{m} = (m_1, \ldots, m_n)$ – the game in which the players pick m_i elements (from the same structure) each in round i, rather than just one element. It is then clear, as for $\equiv_m$, that the following congruence property holds (written here for finite labeled orderings u, v):

Lemma 12. *If $u \equiv_{\overline{m}} u'$ and $v \equiv_{\overline{m}} v'$, then $uv \equiv_{\overline{m}} u'v'$.*

Lemma 13. *A Σ_n-sentence (interpreted over ω-words from Σ^ω) can be written as a Boolean combination of sentences*

$$\forall x \exists y \psi(y)$$

where $\psi(y)$ is a Σ_n-formula bounded in y, *which means that each variable that is quantified in $\psi(y)$ is bounded by y, i.e., it occurs only in the form $\exists z(z \leq y \wedge \ldots)$ or $\forall z(z \leq y \rightarrow \ldots)$.*

Proof. The proof is a copy of that for Theorem 4.4 in [19]. □

Proof. (Theorem 11) Follows from Lemma 13 and Theorem 10. □

7. 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 [1,11,12] 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 one and work with Boolean combinations of sets $\text{ext}(K)$, then determinacy with piecewise testable winning strategies and, respectively, dot-depth one strategies, holds.
(2) Games with winning conditions in $\text{BC}(\text{ext}(\mathcal{K}))$, where $\mathcal{K}$ is now the full class of piecewise testable languages or languages of dot-depth one, are determined only with winning strategies beyond $\mathcal{K}$, namely in DD_2.
(3) This situation is no better when games in $\text{BC}(\lim(\text{DD}_1))$ are considered; there are winning strategies in DD_3 but not in DD_1. For $\text{BC}(\lim(\text{PT}))$ 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 Theorems 9 to 11 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] N. Chaturvedi, J. Olschewski, and W. Thomas. Languages vs. ω-languages in regular infinite games. In G. Mauri and A. Leporati, editors, *Developments in Language Theory*, volume 6795 of *LNCS*, pages 180–191. Springer, 2011.
[3] R. S. Cohen and J. A. Brzozowski. Dot-depth of star-free events. *Journal of Computer and System Sciences*, 5:1–16, February 1971.
[4] H. Ebbinghaus, J. Flum, and W. Thomas. *Mathematical logic*. Springer, 1994.
[5] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games*, volume 2500 of *LNCS*. Springer, 2002.
[6] 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.
[7] D. Perrin and J.-E. Pin. First-order logic and star-free sets. *Journal of Computer and System Sciences*, 32:393–406, June 1986.
[8] D. Perrin and J.-É. Pin. *Infinite Words*. Elsevier, Amsterdam, 2004.
[9] J.-É. Pin. *Varieties of Formal Languages*. North Oxford, London and Plenum, New-York, 1986.
[10] J.-É. Pin. Positive varieties and infinite words. In C. L. Lucchesi and A. V. Moura, editors, *LATIN*, volume 1380 of *LNCS*, pages 76–87. Springer, 1998.
[11] A. Rabinovich and W. Thomas. Logical refinements of Church's problem. In J. Duparc and T. Henzinger, editors, *Computer Science Logic*, volume 4646 of *LNCS*, pages 69–83. Springer, 2007.
[12] V. L. Selivanov. Fine hierarchy of regular aperiodic ω-languages. In T. Harju, J. Karhumäki, and A. Lepistö, editors, *Developments in Language Theory*, volume 4588 of *LNCS*, pages 399–410. Springer, 2007.
[13] V. L. Selivanov. Fine hierarchy of regular aperiodic ω-languages. *International Journal of Foundations of Computer Science*, 19(3):649–675, 2008.
[14] I. Simon. *Hierarchies of events with dot-depth one*. PhD thesis, University of Waterloo, 1972.
[15] I. Simon. Piecewise testable events. In H. Brakhage, editor, *Automata Theory and Formal Languages*, volume 33 of *LNCS*, pages 214–222. Springer, 1975.
[16] L. Staiger and K. W. Wagner. Automatentheoretische und automatenfreie Charakterisierungen topologischer Klassen regulärer Folgenmengen. *Elektronische Informationsverarbeitung und Kybernetik*, 10(7):379–392, 1974.
[17] J. Stern. Characterizations of some classes of regular events. *Theoretical Computer Science*, 35:17–42, 1985.
[18] H. Straubing. *Finite Automata, Formal Logic, and Circuit Complexity*. Birkhäuser Verlag, Basel, Switzerland, 1994.
[19] W. Thomas. A combinatorial approach to the theory of ω-automata. *Information and Control*, 48(3):261–283, 1981.
[20] W. Thomas. Classifying regular events in symbolic logic. *Journal of Computer and System Sciences*, 25(3):360–376, 1982.
[21] W. Thomas. Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages, Vol. 3*, pages 389–455, New York, NY, USA, 1997. Springer.
[22] 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 *LNCS*, pages 635–655. Springer, 2008.