



HAL
open science

Games with Winning Conditions of High Borel Complexity

Olivier Serre

► **To cite this version:**

Olivier Serre. Games with Winning Conditions of High Borel Complexity. Automata, Languages and Programming: 31st International Colloquium, ICALP 2004, Turku, Finland, July 12-16, 2004, 2004, Turku, Finland. pp.1150-1162. hal-00012134

HAL Id: hal-00012134

<https://hal.science/hal-00012134>

Submitted on 17 Oct 2005

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Games With Winning Conditions of High Borel Complexity^{*}

Olivier Serre

LIAFA, Université Paris VII, 2, place Jussieu, case 7014, F-75251 Paris Cedex 05

Abstract. We first consider infinite two-player games on pushdown graphs. In previous work, Cachat, Duparc and Thomas [4] have presented a winning decidable condition that is Σ_3 -complete in the Borel hierarchy. This was the first example of a decidable winning condition of such Borel complexity. We extend this result by giving a family of decidable winning conditions of arbitrary high finite Borel complexity. From this family, we deduce a family of decidable winning conditions of arbitrary finite Borel complexity for games played on finite graphs. The problem of deciding the winner for these winning conditions is shown to be non-elementary complete.

Keywords: Pushdown Automata, Two-player Games, Borel Complexity.

1 Introduction

Infinite two-player games have been intensively studied in the last few years. One of the main motivations is the strong relation that exists with verification questions and controller synthesis. For instance, μ -calculus model checking for finite graphs (respectively for pushdown graphs) is polynomially equivalent to the problem of deciding the winner in a game played on a finite graph [6] (respectively on a pushdown graph [14]). In addition, constructing a winning strategy is the same as synthesizing a discrete controller [1].

One important branch of game theory is developed in the framework of descriptive set theory in which the central question is determinacy, that is the existence of a winning strategy. One of the deepest results is due to Martin [9] and states that, for Borel winning conditions, games are determined. In computer science, the games considered are in general equipped with winning conditions of low Borel complexity and therefore trivially determined. Nevertheless, deciding the winner is, in many cases, a difficult problem. Since we are mostly interested in decidable games, it is natural to ask whether there exist decidable games of arbitrary high finite Borel complexity.

^{*} This research has been partially supported by the European Community Research Training Network “Games and Automata for Synthesis and Validation” (GAMES), (contract HPRN-CT-2002-00283), see www.games.rwth-aachen.de.

For finite graphs and for the natural conditions appearing in verification and model-checking, efficient algorithms are known to decide the winner and to compute the associated winning strategies [11, 15, 7, 13]. These winning conditions all belong to a low level of Borel hierarchy, namely to the boolean closure of the Borel class Σ_2 .

In [12], Thomas proposes to study games with winning conditions of Borel level larger than 2. In the paper we answer this question by exhibiting a family of winning conditions on pushdown games that have an arbitrary high Borel complexity while remaining decidable. As a corollary, one obtains a similar result for games on finite graphs. In addition, these games have effective winning strategies.

The first results concerning high level Borel conditions come from pushdown games. In this model, the game graph is the infinite graph of the configurations of a pushdown process. Walukiewicz has shown that parity games on such graphs can be effectively solved [14]. For this model, higher level winning conditions exploiting the infinity of the stack become natural. In [4], Cachat, Duparc and Thomas have considered the following condition: Eve wins a play if and only if some configuration is infinitely often visited. They have shown that it is a decidable winning condition belonging to Σ_3 . More recently, Bouquet, Serre and Walukiewicz have considered in [2] winning conditions that are boolean combinations of a Büchi condition with a condition called *unboundedness* that requires the stack to be unbounded. These winning conditions are closely related with the one of [4] and remain decidable [2]. A natural question was therefore to consider higher level winning conditions.

In this paper, we give a uniform answer to the question of [12] by providing a family of winning conditions of increasing finite Borel complexity. The main idea is to require the stack to converge to some limit and then to have additional conditions on the limit. To solve classical conditions on pushdown games one method consists in reducing the problem to a game on a finite graph [14, 2]. We will adapt this method and reduce the problem of deciding the winner in a pushdown game to the problem of deciding the winner in another pushdown game, equipped with a lower winning condition. Then the proof goes by induction.

From the proofs we also infer the effectiveness of the winning strategies. Whereas for previously studied winning conditions on pushdown games, the set of winning positions was regular, it is no longer the case here. The exact nature of these sets remains open. We further show that determining the winner for these high level Borel winning conditions is non-elementary complete. We also show that Eve has, from a winning position, a persistent strategy, that is a strategy using memory but such that the move given from some node is always the same for a given play.

The paper is organized as follows. In Section 2, we start with basic definitions on games and introduce the family of winning conditions that we will consider in the rest of the paper. In Section 3, we precise the Borel complexity of these winning conditions. In Section 4, we give the decidability results and construc-

tions of these games. In Section 5 we show that deciding the winner for such winning conditions is non-elementary complete. Finally, in Section 6 we discuss several points. Due to the page limit, the paper contains only proof sketches.

2 Definitions

Definition 1 *An alphabet A is a finite or infinite set of letters. A^* denotes the set of finite words on A , A^ω the set of infinite words on A and A^∞ the set $A^* \cup A^\omega$. The empty word is denoted by ε . For a word u , we denote its (possibly infinite) length by $|u|$.*

Definition 2 (Prefix) *Let $u \in A^*$ and $v \in A^\infty$. Then u is a prefix of v , denoted $u \sqsubseteq v$ if there exists some word $w \in A^\infty$ such that $v = u \cdot w$. For any word $u \in A^\infty$ there exists a unique prefix of length k for all $k \leq |u|$. This prefix is denoted by $u \upharpoonright k$.*

Definition 3 (Limit of a sequence of finite words) *Let $(u_i)_{i \geq 1} \in (A^*)^\mathbb{N}$ be an infinite sequence of finite words. The limit $\lim_{i \geq 1} u_i$ of $(u_i)_{i \geq 1}$, is the maximal word satisfying the following: for each j there exists an index r such that the j -th letter of $\lim_{i \geq 1} u_i$ equals the j -th letter of u_p for every $p \geq r$. Note that the $\lim_{i \geq 1} u_i$ can be either finite or infinite.*

We recall now some basic definitions on games. For more details and basic results, we refer to [11, 15].

Infinite two-player game. Let $G = (V, E)$ be a possibly infinite graph. Let $V = V_E \cup V_A$ be a partition of the nodes among two players, Eve and Adam. A *play* from some node v_0 proceeds as follows: if $v_0 \in V_E$, Eve chooses a successor v_1 such that $(v_0, v_1) \in E$. Otherwise, it is Adam's turn to choose a successor v_1 . If there is no such v_1 , then the play ends in v_0 , otherwise the player to whom v_1 belongs tries to move to some v_2 and so on. Therefore a play starting from v_0 is a finite or infinite sequence $v_0 v_1 v_2 \dots$ such that $(v_i, v_{i+1}) \in E$ for all i . In the case where the play is finite, we require that there is no $v \in V$ such that $(v_n, v) \in E$, if v_n was the last node of the play. A *partial play* is a prefix of a play.

A *winning condition* for Eve in G is a subset $\Omega \subseteq V^\omega$. An infinite play is winning for Eve with respect to the winning condition Ω if it belongs to Ω . A finite play is won by the player that has made the last move (in other words, the player that cannot move loses the play). In the rest of the paper, we will suppose that all plays are infinite as the games we consider can easily be reduced to equivalent games where all plays are infinite. An *infinite two-player game* is a tuple $\mathcal{G} = (G, V_E, V_A, \Omega)$ where $G = (V, E)$ is a graph, $V = V_E \cup V_A$ and Ω is a winning condition for Eve in G .

Strategies, winning positions. Let $\mathcal{G} = (G, V_E, V_A, \Omega)$ be an infinite two-player game. A strategy for Eve is a partial function $\varphi : V^* V_E \rightarrow V$ such that

for any partial play $\lambda \in V^+$ ending in some node v and such that $\varphi(\lambda) = v'$, we have $(v, v') \in E$. In other words, a strategy is a function that associates with any partial play a valid move in G . Eve *respects a strategy* φ in a play $\lambda = v_0v_1v_2 \dots$ if for all $v_i \in V_E$, $v_{i+1} = \varphi(v_0v_1 \dots v_i)$. A strategy φ is a *winning strategy* for Eve from some position v_0 if Eve wins all the plays starting from v_0 and where she respects φ . A position v is a *winning position* for Eve if she has a winning strategy from it. Symmetrically one defines the strategies and winning positions for Adam. We will denote by W_E and W_A the respective sets of winning positions for Eve and Adam. Martin's determinacy theorem [9] tells that $V = W_E \cup W_A$ whenever $\Omega \subseteq V^\omega$ is a Borel set.

2.1 Games on Pushdown Graphs and the Winning Condition

$$\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$$

Pushdown process. A *pushdown process* is a tuple $\mathcal{P} = (Q, \Gamma, \perp, \hookrightarrow)$ where Q is the finite set of states, Γ is the finite set of stack symbols, $\perp \in \Gamma$ is a special stack symbol (bottom) and \hookrightarrow is the transition relation. A configuration of \mathcal{P} is a pair (q, u) with $q \in Q$ and $u \in (\Gamma \setminus \{\perp\})^* \perp$ (the top stack symbol is the leftmost symbol of u). The bottom stack symbol \perp is never put nor removed from the stack. The transition relation that can be applied to the set of configurations of \mathcal{P} have one of the following form (note that we push or pop only one letter at each transition.):

- *Push*(q, b): $(p, a) \hookrightarrow (q, ba)$, where $p, q \in Q$, $a \in \Gamma$ and $b \in (\Gamma \setminus \{\perp\})$.
- *Pop*(q): $(p, a) \hookrightarrow (q, \varepsilon)$, where $p, q \in Q$ and $a \in (\Gamma \setminus \{\perp\})$.

By $(p, v) \rightarrow (q, w)$ we mean that from the configuration (p, v) the pushdown process can go in one step to (q, w) . Note that the emptiness test of the stack corresponds to a push rule with $a = \perp$. This naturally leads to associate with any pushdown process an infinite graph as follows:

Pushdown graph. With any pushdown process \mathcal{P} , one can associate a *pushdown graph* defined as the directed graph having the set of configurations of \mathcal{P} as nodes and the edges of which are given by the relation \rightarrow .

To define a *two-player game* on a pushdown graph $G = (V, \rightarrow)$ associated with a pushdown process \mathcal{P} , we need a partition $Q_E \cup Q_A$ of the set of states Q . From this partition follows a partition $V_E \cup V_A$ of the nodes V of G among the two players: the nodes of Eve are those whose control state belongs to Q_E and the others are Adam's nodes.

From now on we assume that we are given a pushdown process $\mathcal{P} = (Q, \Gamma, \perp, \hookrightarrow)$ and a partition $Q_E \cup Q_A$ of Q .

For a node $v = (p, u)$ of V we define $|v|$ to be the length of u . In a play $v_1v_2v_3 \dots$, the stack is *strictly unbounded* if the stack size converges to $+\infty$. More formally we require that for all $k \geq 0$, there exists some i such that: for all $j \geq i$, $|v_j| > k$.

If the stack in a play $v_1v_2 \dots$ is strictly unbounded, we will consider its *limit* $u = a_1a_2a_3 \dots$, $a_i \in \Gamma$, which is formally defined by: for all $k \geq 1$, there

exists some i such that for all $j \geq i$, there exists $w_j \in \Gamma^*$ such that $v_j = (p_j, w_j \cdot a_1 \cdots a_k)$.

Let us now recall the classical notion of deterministic pushdown automaton:

Deterministic pushdown automaton. A *deterministic pushdown automaton* with input from Σ^∞ is a tuple $\mathcal{A} = (Q, \Gamma, \Sigma, \perp, q^{in}, \delta)$, where Q is the finite set of states, Γ is the finite set of stack symbols, $\perp \in \Gamma$ is a special stack symbol (bottom), Σ is the input alphabet, $q^{in} \in Q$ is the initial state and $\delta : Q \times \Gamma \times \Sigma \rightarrow M$ is the transition function where $M = \{pop(q) \mid q \in Q\} \cup \{push(q, a) \mid q \in Q, a \in \Gamma\}$. A *run* of \mathcal{A} on an infinite word $u = a_0 \cdot a_1 \cdots$ starts from the configuration (q^{in}, \perp) . \mathcal{A} starts reading a_0 and applies the rule corresponding to $\delta(q^{in}, \perp, a_0)$ (that is the control state and the stack contents are changed), then it reads a_1 and so on. One can in addition equip such an automaton with a classical acceptance condition, for instance a Büchi condition.

Now let us consider a collection $\mathcal{A}_1, \dots, \mathcal{A}_n$ of deterministic pushdown automata. Let \mathcal{B} be a deterministic pushdown automaton equipped with a Büchi acceptance condition. We require the following properties on the input and stack alphabets of $\mathcal{A}_1, \dots, \mathcal{A}_n, \mathcal{B}$:

- \mathcal{A}_1 has Γ as input alphabet.
- For all $i \geq 1$, the stack alphabet of \mathcal{A}_i is the input alphabet of \mathcal{A}_{i+1} .
- The stack alphabet of \mathcal{A}_n is the input alphabet of \mathcal{B} .

For given $\mathcal{A}_1, \dots, \mathcal{A}_n, \mathcal{B}$ we define the winning condition $\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$ as follows. Eve wins an infinite play in a pushdown graph G associated with a pushdown process \mathcal{P} , if and only if the following conditions are satisfied:

- The stack of \mathcal{P} is strictly unbounded and therefore the sequence of stack contents converges to some limit $u_0 \in \Gamma^\omega$.
- For all $1 \leq i \leq n$, when \mathcal{A}_i reads u_{i-1} , its stack is strictly unbounded and the sequence of stack contents converges to some limit u_i .
- \mathcal{B} accepts u_n .

In particular, Eve wins an infinite play in G with the winning condition $\Omega_{\mathcal{B}}$ if and only if the stack is strictly unbounded and converges to some limit accepted by \mathcal{B} . For instance, if \mathcal{B} accepts all infinite words, $\Omega_{\mathcal{B}}$ is the winning condition for Adam considered in [4].

2.2 Games on Finite Graphs and the Winning Condition

$$\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$$

We give now a version of the winning condition $\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$ for a game played on a finite game graph $G = (V = V_E \cup V_A, E)$.

Let us consider a collection $\mathcal{A}_1, \dots, \mathcal{A}_n$ of deterministic pushdown automata. Let \mathcal{B} be a deterministic pushdown automaton equipped with a Büchi acceptance condition. We require the same properties on the input and stack alphabets of $\mathcal{A}_1, \dots, \mathcal{A}_n, \mathcal{B}$ as in the preceding case. In addition, we require that \mathcal{A}_1 (we set $\mathcal{A}_1 = \mathcal{B}$ if $n = 0$) has V as input alphabet.

$\mathcal{A}_1, \dots, \mathcal{A}_n, \mathcal{B}$ induce a winning condition on G denoted by $\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$ and defined as follows. Eve wins a play in G if and only if the following conditions are satisfied:

- The play is infinite and is therefore a word $u_0 \in V^\omega$.
- For all $1 \leq i \leq n$, when \mathcal{A}_i reads u_{i-1} , its stack is strictly unbounded and the sequence of stack contents converges to some limit u_i .
- \mathcal{B} accepts u_n .

In particular, Eve wins an infinite play in G with the winning condition $\Omega_{\mathcal{B}}$ if and only if the play seen as an element of V^ω is accepted by \mathcal{B} . For instance, such a condition can require that a play in G is won if and only if any partial play contains more nodes from V_E than V_A .

Let $G = (V_E \cup V_A, E)$ be a finite game graph equipped with the winning condition $\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$ (with $n \geq 1$). Let $\mathcal{A}_1 = (Q_1, V, \Gamma_1, q_{\mathcal{A}_1}^{in}, \delta)$. Let us define a pushdown process $\mathcal{P} = (Q, \Gamma, \perp, \hookrightarrow)$ where $Q = V \times Q_1$, $\Gamma = \Gamma_1$ and \hookrightarrow is such that:

- $((v, q), a) \hookrightarrow ((v', q'), \varepsilon)$ if and only if $(v, v') \in E$ and $\delta(q, v, a) = pop(q')$.
- $((v, q), a) \hookrightarrow ((v', q'), ba)$ if and only if $(v, v') \in E$ and $\delta(q, v, a) = push(q', b)$.

Let $G \otimes \mathcal{A}_1$ be the pushdown graph induced by \mathcal{P} equipped with the partition $Q = Q_E \cup Q_A$ where $Q_E = V_E \times Q_1$ and $Q_A = V_A \times Q_1$.

The following lemma reduces the game on the finite graph to a pushdown game with a simpler winning condition:

Lemma 1. *Let $G = (V, E)$ be a finite game graph. Eve wins in G from some position $v \in V$ with the winning condition $\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$ if and only if she wins in $G \otimes \mathcal{A}_1$ from $((v, q_{\mathcal{A}_1}^{in}), \perp)$ with the winning condition $\Omega_{\mathcal{A}_2 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$ (respectively Büchi if $n = 0$).*

3 Borel Complexity

3.1 Borel Hierarchy

Let Σ be a (possibly infinite) alphabet. We consider the set Σ^ω of infinite words on the alphabet Σ and we equip it with the usual Cantor topology where the open sets are those of the form $W \cdot \Sigma^\omega$ where $W \subseteq \Sigma^*$ is a language of finite words on the alphabet Σ . The *finite Borel hierarchy* $(\Sigma_1, \Pi_1), (\Sigma_2, \Pi_2), \dots$ is inductively defined as follows:

- $\Sigma_1 = \{W \cdot \Sigma^\omega \mid W \subseteq \Sigma^*\}$ is the set of open sets.
- For all $n \geq 1$, $\Pi_n = \{\overline{S} \mid S \in \Sigma_n\}$ consists of the complements of Σ_n -sets.
- For all $n \geq 1$, $\Sigma_{n+1} = \{\bigcup_{i \in \mathbb{N}} S_i \mid \forall i \in \mathbb{N}, S_i \in \Pi_n\}$ is the set of countable union of Π_n -sets.

3.2 Borel Complexity of a Winning Condition

Let Ω be a winning condition. Ω is a Σ_n -winning condition if and only if Ω is a Σ_n -set. In the following, we may consider winning conditions given in a more abstract way than subsets of infinite words. For instance, we may consider

a Büchi winning condition, that is a condition where we require to infinitely visit some final nodes. Such a condition can be defined independently of the graph (except that implicitly, a subset of final nodes has to be defined). Such an abstract condition is a Σ_n -winning condition if there exists some graph for which the set of winning plays is a Σ_n -set. Note also that we do not focus on the set of effective winning plays but on the set of winning plays (some may not be effectively possible). In the same way, one defines Π_n -winning conditions.

Example 1. Consider a Büchi winning condition (Eve wins if and only if she infinitely visits nodes belonging to some subset $F \subseteq V$). Such a condition is a Π_2 -winning condition, as the set of winning plays for Eve is $\bigcap_{n \geq 0} [(V^j V^* F) V^\omega]$.

Consider now a strict unboundedness winning condition for pushdown games. The corresponding condition for Adam is the one considered in [4]: Adam wins if and only if some configuration (equivalently some stack size) is infinitely repeated. Therefore, if one denotes by V_n the set of configurations of stack size n , the set of winning plays for Adam is $\bigcup_{n \geq 0} \bigcap_{m \geq 0} [(V^m V^* V_n) V^\omega]$. Therefore, the set of winning plays for Adam is a Σ_3 -set and thus the strict unboundedness winning condition is a Π_3 -winning condition for Eve.

Borel sets can be equipped with a reduction notion, inducing a notion of completeness [5]:

Definition 4 (Wadge Reduction, Complete Sets) *We say that $X \subseteq \Sigma_X^\omega$ Wadge reduces to $Y \subseteq \Sigma_Y^\omega$, denoted $X \leq_W Y$, if and only if there exists a continuous function $f : \Sigma_X^\omega \rightarrow \Sigma_Y^\omega$ such that $X = f^{-1}(Y)$. If $X \leq_W Y$ and $Y \leq_W X$ then we say that X and Y are Wadge equivalent and we denote it by $X \equiv_W Y$. A set $S \in \Sigma_n$ is Σ_n -complete if and only if $X \leq_W S$ for all $X \in \Sigma_n$.*

Example 2. Let $\Sigma = \{a, b\}$. Let $X \subseteq \Sigma^\omega$ be the set of infinite words that contain infinitely many a . X is a Π_2 -complete set. Effectively $X = \bigcap_{i \geq 0} \Sigma^{\geq i} a \Sigma^\omega$: X is a Π_2 set. Let Y be a Π_2 set on some alphabet Γ : $Y = \bigcap_{i \geq 0} Y_i$ for some family $(Y_i)_{i \geq 0}$ of open sets, where $Y_i = Z_i \Gamma^\omega$. We define a function $f : \Gamma^\omega \rightarrow \{a, b\}^\omega$ by setting $f(x) = a_1 \cdot a_2 \cdot \dots$, where $a_i = a$ if and only if $x \upharpoonright i \in Z_k \Gamma^*$, where k is the number of letters a_1, a_2, \dots, a_{i-1} equals to a . We have that $Y = f^{-1}(X)$ and that f is continuous.

3.3 Borel Complexity of $\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$

In [5], Duparc has introduced an operation on sets that allows to define sets of arbitrary high Borel complexity in a uniform way. Once reading some special letter, the letter that was just before it is erased. Applying this operation to some set, gives, under several conditions, a set having a higher Borel complexity. This operation can be iterated and therefore allows to construct sets of arbitrary Borel complexity. The winning conditions $\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$ can simulate the n -times iterated version of the eraser operator. Intuitively, popping the top symbol of a stack is the same that erasing in a word the last letter.

We have the two following results concerning the Borel complexity of $\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$:

Theorem 1 For all n , there exists a collection of deterministic pushdown automata $\mathcal{A}_1, \dots, \mathcal{A}_n$ and a deterministic Büchi pushdown automaton \mathcal{B} such that $\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$ is a Π_{n+3} -complete winning condition for pushdown games.

Theorem 2 For all n , there exists a collection of deterministic pushdown automata $\mathcal{A}_1, \dots, \mathcal{A}_n$ and a deterministic Büchi pushdown automaton \mathcal{B} such that $\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$ is a Π_{n+2} -complete winning condition for games on finite graphs.

4 Decidability

In this section we show that pushdown games equipped with the winning condition $\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$ are decidable.

Theorem 3 Let $\mathcal{P} = (Q, \Gamma, \perp, \leftrightarrow)$ be a pushdown process, let $Q_E \cup Q_A$ be a partition of Q and let $G = (V, \rightarrow)$ be the associated pushdown graph. For any collection $\mathcal{A}_1, \dots, \mathcal{A}_n$ of deterministic pushdown automata, for any deterministic Büchi pushdown automaton \mathcal{B} , and for all $q \in Q$, it is decidable whether Eve has a winning strategy from (q, \perp) in the pushdown game $(G, V_E, V_A, \Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}})$.

The proof follows from iterating the result of Proposition 1 below and finally using the known algorithms on Büchi pushdown games [14].

Consider automata $\mathcal{A}_1, \dots, \mathcal{A}_n, \mathcal{B}$ inducing a winning condition $\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$. We reduce the problem of solving a game with the condition $\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$ to that of solving an exponentially larger pushdown game $\tilde{G} \times \mathcal{A}$ equipped with the simpler winning condition $\Omega_{\mathcal{A}_2 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$ if $n > 0$, resp. a Büchi condition if $n = 0$.

We define \mathcal{A} to be \mathcal{A}_1 if $n > 0$ and to be \mathcal{B} otherwise. In addition, $\Gamma_{\mathcal{A}}$ will denote the stack alphabet of \mathcal{A} , $q_{\mathcal{A}}^{in}$ its initial state and $\delta_{\mathcal{A}}$ its transition function. The game $\tilde{G} \times \mathcal{A}$ is presented in Figure 1.

For readability we will use the abbreviations *CS* and *CM* for *Choose_sets* and *Choose_move*. We give here an intuitive meaning of this graph. Intuitively, a node $[(p, a, R), t, u]$ represents a position (p, av) in G , such that Eve can play so that, if the top symbol a is eventually popped, it leads to a node (r, v) for some $r \in R$. In addition reading v^R (the mirror image of v) by \mathcal{A} leads to the configuration (t, u) of \mathcal{A} . Thus, $\tilde{G} \times \mathcal{A}$ encodes an on-the-fly computation of \mathcal{A} . Such a node belongs to Eve if and only if $p \in Q_E$.

CS and *CM* nodes are for push moves: once the player to whom belongs $[(p, a, R), t, u]$ has decided to push b and to change the state to p' , which is represented by the node $[CS(p, a, R, p', b), t, u]$, Eve, by moving to some node $[CM(p, a, R, p', b, S), t, u]$, announces the set S of states that she can ensure to reach if b is eventually popped. Then, if Adam wants b to be popped he moves to some state $[(s, a, R), t, u]$ for some $s \in S$. Otherwise, if he does not want to pop b or if he believes that S is incorrect, he moves to $[(p', b, S), t', u']$ (the edge he follows is called a *push edge*). Intuitively in that case, a is fixed forever and therefore, the on-the-fly computation of \mathcal{A} must be updated: (t', u') is the

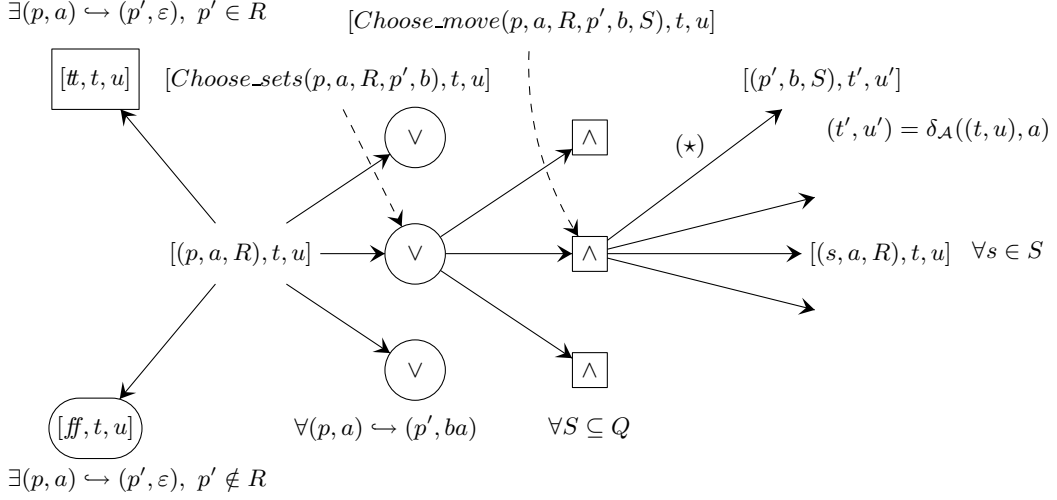


Fig. 1. $\tilde{G} \times \mathcal{A}$: oval nodes belong to Eve, square for Adam.

configuration reached in \mathcal{A} from (t, u) by reading a . In the special case where $n = 0$, $\mathcal{A} = \mathcal{B}$ is equipped with a Büchi condition. Therefore, a push edge is marked final (\star on the figure) if and only if t' is a final state of \mathcal{B} .

We have the following key result:

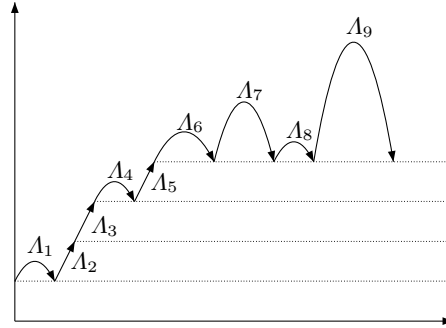
Proposition 1 *Eve wins from (q, \perp) in the game played on G with the winning condition $\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$ if and only if she wins from $[(q, \perp, \emptyset), q_{\mathcal{A}}^{in}, \perp]$ in the pushdown game $\tilde{G} \times \mathcal{A}$ equipped with the winning condition $\Omega_{\mathcal{A}_2 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$ if $n > 0$ (resp., equipped with the Büchi condition if $n = 0$).*

Sketch of proof. Let A be some play in the pushdown game.

A can be represented graphically by the evolution of the stack during the play. Such a representation can be decomposed in a unique way into bumps (from some level the stack increases and eventually return to the level) and push moves that leaves some level forever. Such a decomposition is called a *P/B Factorization*. Figure 2 gives an example of such a factorization for a prefix of some play.

A play in $\tilde{G} \times \mathcal{A}$ can be considered as a P/B factorization where in addition one performs an on-the-fly computation of \mathcal{A} on some prefix of the limit (once some push move had been made the level will no longer be visited and therefore some prefix of the limit is known).

Fig. 2. P/B Factorization



In the case $n = 0$, the strict unboundedness and the acceptance of the limit is ensured by the fact that a final state only appears after a push move (implies strict unboundedness) inducing a prefix leading to a final state in \mathcal{B} . From these remarks follows easily the direct implication. In the case $n > 0$, the winning condition $\Omega_{\mathcal{A}_2 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$ implies unboundedness (\mathcal{A} 's stack has to be unbounded which implies that an infinite number of push edges were taken in $\tilde{G} \times \mathcal{A}$). Cascade acceptance by $\mathcal{A}_1, \dots, \mathcal{A}_n, \mathcal{B}$ is due to the on-the-fly computation of \mathcal{A}_1 that implies a cascade acceptance by $\mathcal{A}_2, \dots, \mathcal{A}_n, \mathcal{B}$.

For the converse implication, one notes that winning in $\tilde{G} \times \mathcal{A}$, allows to have a winning strategy in the original pushdown game. For this, one needs to use a stack as memory that stores partial plays in $\tilde{G} \times \mathcal{A}$ and allows to reconstruct the bumps (recall that $\tilde{G} \times \mathcal{A}$ only represents the factorization). This direction is much more technical than the preceding one. This construction also implies effectivity of the winning strategy in the original game.

The result for games on finite graphs, is shown similarly, using Lemma 1 and Theorem 3.

5 Complexity

We first start with some definitions:

Definition 5 Let $h, N \geq 0$. $tow(h, N)$ is defined inductively by:

- $tow(0, N) = N$.
- $tow(h, N) = 2^{tow(h-1, N)}$ for $h \geq 1$.

Definition 6 (h -DEXPTIME) Let consider a problem P and a deterministic Turing Machine deciding in $\mathcal{O}(tow(h, N))$ steps whether some instance of the problem P is true, where N is polynomial in the size of the instance. Then the problem P belongs to the class h -DEXPTIME.

We have the following results for pushdown games equipped with winning conditions of the form $\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_k \triangleright \mathcal{B}}$:

Proposition 2 Let $\mathcal{A}_1, \dots, \mathcal{A}_k$ be a collection of deterministic pushdown automata, and let \mathcal{B} be a deterministic Büchi pushdown automaton. Let \mathcal{G} be a pushdown game equipped with the winning condition $\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_k \triangleright \mathcal{B}}$. Deciding the winner in such a game is an $(k + 2)$ -DEXPTIME problem.

Proof. By induction on k and noting that the construction given in the proof of Proposition 1 induces an exponential blow up, and that deciding the winner in a Büchi pushdown game is a DEXPTIME problem.

Proposition 3 The problem of deciding the winner in a pushdown game equipped with a winning condition of the form $\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_k \triangleright \mathcal{B}}$, with $k \geq 0$, is a $(k + 1)$ -DEXPTIME hard problem.

Therefore we have the following result:

Theorem 4 *The problem of deciding the winner in a pushdown game (resp. a finite game graph) equipped with a winning condition of the form $\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_k \triangleright \mathcal{B}}$ is non-elementary complete (in the size of the winning condition).*

6 Winning Positions and Strategies

We first give some results on the form of the set of winning positions for pushdown games equipped with the winning condition $\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$.

In [10, 3], it is shown that the set of winning positions in a parity pushdown game is a regular language. In fact, using the same techniques, one can prove a similar result for various winning conditions, as for instance for unboundedness or strict unboundedness.

For the conditions we study in this paper, the set of winning positions may not be regular. For instance, every deterministic context-free language may occur as a winning set:

Proposition 4 *Let \mathcal{A} be some deterministic pushdown automaton on finite words. There exists a deterministic Büchi automaton \mathcal{B} , a pushdown process $\mathcal{P} = (Q, \Gamma, \perp, \hookrightarrow)$, a state $q \in Q$ and a partition $Q = Q_E \cup Q_A$ such that, in the induced pushdown game equipped with the winning condition $\Omega_{\mathcal{B}}$, the set $\{u \mid (q, u) \in W_E\}$ is exactly the set of words recognized by \mathcal{A} .*

First, let us recall what a persistent strategy is:

Definition 7 (Persistent strategy) [8] *A strategy φ for Eve is persistent if for each play $v_1 v_2 \dots v_k$ played by Eve according to this strategy, if $v_i = v_j$, for some $1 \leq i, j < k$, and v_i is a vertex where Eve is to move, then $v_{i+1} = v_{j+1}$.*

In other words, a persistent strategy may require memory but once a choice is made, it is done forever. For the winning conditions of the form $\Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}}$, we show that Eve has a persistent winning strategy. More precisely:

Theorem 5 *Let $\mathcal{P} = (Q, \Gamma, \perp, \hookrightarrow)$ be a pushdown process, let $Q_E \cup Q_A$ be a partition of Q and let $G = (V, \rightarrow)$ be the associated pushdown graph. For any collection $\mathcal{A}_1, \dots, \mathcal{A}_n$ of deterministic pushdown automata, for any deterministic Büchi pushdown automaton \mathcal{B} , Eve has a persistent winning strategy from any winning position in the pushdown game $\mathcal{G} = (G, V_E, V_A, \Omega_{\mathcal{A}_1 \triangleright \dots \triangleright \mathcal{A}_n \triangleright \mathcal{B}})$.*

7 Conclusion

We have provided a family of winning conditions that have an arbitrary high Borel complexity while remaining decidable for pushdown games and games

on finite graphs. Deciding the winner for such a winning condition is a non-elementary complete problem. In addition, for pushdown games, it gives an example of decidable winning conditions inducing non regular sets of winning positions. The exact form of the winning sets remains open. Finally, we have shown that there are persistent winning strategies for pushdown games equipped with these winning conditions. The existence of positional strategies remains open.

Acknowledgments. I gratefully acknowledge Jacques Duparc for suggesting me to study this family of conditions. His advices and knowledge of Borel complexity were very important in this research. I would also like to express my thanks to Anca Muscholl for her help while writing this paper.

References

1. A. Arnold, A. Vincent, and I. Walukiewicz. Games for synthesis of controllers with partial observation. *Theoretical Computer Science*, 303(1):7–34, 2003.
2. A. Bouquet, O. Serre, and I. Walukiewicz. Pushdown games with the unboundeness and regular conditions. In *Proceedings of FST TCS 2003*, volume 2914 of *Lecture Notes in Computer Science*, pages 88–99. Springer, 2003.
3. T. Cachat. Uniform solution of parity games on prefix-recognizable graphs. volume 68 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2002.
4. T. Cachat, J. Duparc, and W. Thomas. Solving pushdown games with a Σ_3 winning condition. In *Proceedings of CSL 2002*, volume 2471 of *Lecture Notes in Computer Science*, pages 322–336. Springer, 2002.
5. J. Duparc. Wadge hierarchy and Veblen hierarchy. part I: Borel sets of finite rank. *Journal of Symbolic Logic*, 66(1):56–86, 2001.
6. E.A. Emerson, C.S. Jutla, and A.P. Sistla. On model-checking for the mu-calculus and its fragments. *Theoretical Computer Science*, 258(1-2):491–522, 2001.
7. M. Jurdziński. Small progress measures for solving parity games. In *Proceeding of STACS 2000*, volume 1770 of *Lecture Notes in Computer Science*, pages 290–301. Springer-Verlag, 2000.
8. J. Marcinkowski and T. Truderung. Optimal complexity bounds for positive LTL games. In *Proceedings of CSL 2002*, volume 2471 of *Lecture Notes in Computer Science*, pages 262–275. Springer, 2002.
9. D.A. Martin. Borel determinacy. *Annals of Mathematics*, 102(363-371), 1975.
10. O. Serre. Note on winning positions on pushdown games with ω -regular conditions. *Information Processing Letters*, 85:285–291, 2003.
11. W. Thomas. On the synthesis of strategies in infinite games. In *Proceedings of STACS '95*, volume 900 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 1995.
12. W. Thomas. Infinite games and verification (extended abstract of a tutorial). In *Proceedings of CAV 2002*, volume 2404 of *Lecture Notes in Computer Science*, pages 58–64. Springer, 2002.
13. J. Vöge and M. Jurdziński. A discrete strategy improvement algorithm for solving parity games. In *Proceedings of CAV 2000*, volume 1855 of *Lecture Notes in Computer Science*, pages 202–215. Springer-Verlag, 2000.
14. I. Walukiewicz. Pushdown processes: games and model checking. *Information and Computation*, 157:234–263, 2000.
15. W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200(1-2):135–183, 1998.