Solving Pushdown Games with a Σ_3 Winning Condition*

Thierry Cachat, Jacques Duparc, and Wolfgang Thomas

Lehrstuhl für Informatik VII, RWTH, D-52056 Aachen
Fax: (49) 241-80-22215 {cachat,duparc,thomas}@informatik.rwth-aachen.de

Abstract We study infinite two-player games over pushdown graphs with a winning condition that refers explicitly to the infinity of the game graph: A play is won by player 0 if some vertex is visited infinity often during the play. We show that the set of winning plays is a proper Σ_3-set in the Borel hierarchy, thus transcending the Boolean closure of Σ_2-sets which arises with the standard automata theoretic winning conditions (such as the Muller, Rabin, or parity condition). We also show that this Σ_3-game over pushdown graphs can be solved effectively (by a computation of the winning region of player 0 and his memoryless winning strategy). This seems to be a first example of an effectively solvable game beyond the second level of the Borel hierarchy.

1 Introduction

The theory of infinite two-person games, originally developed in descriptive set theory, has found enormous interest in recent years also in theoretical computer science. Whereas in the framework of set theory, the mere existence of winning strategies is the central question, the applications in computer science are concerned with algorithmic aspects. In the past ten years, this development led to interesting connections with the verification and automatic synthesis of reactive programs (see, e.g., [13,16]). It turned out that central problems in the verification of state-based systems can be studied in the game theoretical framework (an example is the model-checking problem for the modal μ-calculus), and that the construction of discrete controllers can be viewed as the synthesis of winning strategies in certain infinite games.

The standard setting of these applications are the finite-state games. Here one deals with a finite game graph where each vertex is associated to one of the two players (called 0 and 1). A play is an infinite sequence of vertices which arises when a token is moved through the graph, where in each step the token is moved by the player to whom the current vertex is associated. The winning condition (say for player 0) is given by an automata theoretic acceptance condition applied to plays. A prominent example is the Muller condition which is specified by a family $\mathcal{F}$ of vertex sets and which requires that the vertices visited infinitely often

* A preliminary version of this article appeared in the proceedings of CSL 2002, LNCS 2471, pp. 322–336

in the considered play form a set in $\mathcal{F}$. The core result on finite-state games is the Büchi-Landweber Theorem ([2]). It says that for a game on a finite graph with Muller winning condition one can compute the "winning region" of player 0 (i.e., the set of vertices from which player 0 has a winning strategy) and that the corresponding winning strategies are executable by finite automata. Many more results have been shown, in particular on the so-called parity games, where even memoryless strategies suffice [5,14].

The Muller and parity winning conditions (as well as related ones like Rabin and Streett conditions) define sets of plays which are located at a very low level of the Borel hierarchy, namely in $B(\boldsymbol{\Sigma}_2)$, the Boolean closure of the Borel class $\boldsymbol{\Sigma}_2$. This restriction to winning conditions of low set theoretical complexity is justified by two reasons: First, most winning conditions which are motivated by practical applications (safety, liveness, assume-guarantee properties, fairness, etc.), and Boolean combinations thereof, all define sets in $B(\boldsymbol{\Sigma}_2)$. Secondly, by Büchi's and McNaughton's results on the transformation of monadic second-order logic formulas into deterministic Muller automata, any winning condition which is formalizable in linear time temporal logic or in monadic second-order logic (S1S) over infinite strings defines a $B(\boldsymbol{\Sigma}_2)$-set. (One transforms a logical formula φ into an equivalent deterministic Muller automaton, say with transition graph G_φ, and proceeds from a game graph G and a winning condition defined by φ to $G \times G_\varphi$ as game graph equipped with the Muller winning condition applied to the second components of vertices.) In this connection, Büchi claims in [3, p. 1173] as a general thesis that any set of ω-sequences with an "honestly finite presentation" (by some form of "finite-state recursion") belongs to $B(\boldsymbol{\Sigma}_2)$.

Recently, the Büchi-Landweber Theorem was extended to infinite game graphs, and in particular to the transition graphs of pushdown automata [10,11,16]. For example, it was shown by Walukiewicz [16] that parity games over pushdown graphs can be solved effectively. But the restriction to the parity condition is now only justifiable by pragmatic aspects, and it is well conceivable that higher levels of the Borel hierarchy are reachable by natural winning conditions exploiting the infinity of pushdown transition graphs.

In the present paper we propose such a winning condition, by the requirement that (in a winning play) there should be one vertex occurring infinitely often. Syntactically, this is formulated as a condition on a play ρ using a $\boldsymbol{\Sigma}_3$-prefix of unbounded quantifiers:

> "there is a vertex v such that for all time instances t there is $t' > t$ such that v is visited at t' in the play ρ under consideration"

In Section 3 below we show that for a suitable deterministic pushdown automaton the corresponding set of winning plays forms indeed a $\boldsymbol{\Sigma}_3$-complete set in the Borel hierarchy. The completeness proof needs some prerequisites of set theory, in particular on continuous reductions and the Wadge game [15]. In Section 2, these preparations are collected.

In Section 4 we show that the $\boldsymbol{\Sigma}_3$-winning condition does not prohibit an algorithmic solution of the corresponding games. Building on the approach of

[4] for Büchi games over pushdown graphs, we present an algorithm to decide whether a given vertex of a pushdown transition graph is in the winning region of player 0; and from this, also a memoryless winning strategy can be extracted.

In the final section we discuss some related acceptance conditions (studied in ongoing work) which involve a specified set F of vertices and requires that some $v \in F$ is visited infinitely.

The main result of this paper may be considered as a first tiny step in a far-reaching proposal of Büchi ([3, p.1171-72]). He considers constructive game presentations by "state-recursions", as they arise in automata theoretic games, and he asks to extend the construction of winning strategies in the form of "recursions" (i.e., algorithmic procedures) from the case of $B(\boldsymbol{\Sigma}_2)$-games to appropriate games on arbitrary levels of the Borel hierarchy.

2 Borel Hierarchy and Wadge Game

Given a finite alphabet Σ, we consider the set Σ^ω of all infinite words over Σ as a topological space by equipping it with the Cantor topology, where the open sets are those of the form $W \cdot \Sigma^\omega$ for some set $W \subseteq \Sigma^*$ of finite words. The *finite Borel Hierarchy* is a sequence $\boldsymbol{\Sigma}_1, \boldsymbol{\Pi}_1, \boldsymbol{\Sigma}_2, \boldsymbol{\Pi}_2, \ldots$ of classes of ω-languages over Σ, inductively defined by:

– $\boldsymbol{\Sigma}_1 = \{\textit{Open sets}\} = \{W \cdot \Sigma^\omega : W \subseteq \Sigma^*\}$
– $\boldsymbol{\Pi}_n = \left\{A^{\complement} : A \in \boldsymbol{\Sigma}_n\right\}$ (for $n \geq 1$)
– $\boldsymbol{\Sigma}_{n+1} = \left\{\bigcup_{i \in \mathbb{N}} A_i : \forall i \in \mathbb{N} \ \ A_i \in \boldsymbol{\Pi}_n\right\}$ (for $n \geq 0$)

Let $B(\boldsymbol{\Sigma}_n)$ be the class of Boolean combinations of $\boldsymbol{\Sigma}_n$-sets. The Borel classes are arranged in the form

$$\begin{tikzcd}
\boldsymbol{\Sigma}_1 \arrow[dr] & & \boldsymbol{\Sigma}_2 \arrow[dr] & & \\
& B(\boldsymbol{\Sigma}_1) \arrow[ur] \arrow[dr] & & B(\boldsymbol{\Sigma}_2) \arrow[ur] \arrow[dr] & & \cdots \\
\boldsymbol{\Pi}_1 \arrow[ur] & & \boldsymbol{\Pi}_2 \arrow[ur] & &
\end{tikzcd}$$

where each arrow denotes strict inclusion. A set that is in $\boldsymbol{\Sigma}_k$ but not in $\boldsymbol{\Pi}_k$ is called a *true-$\boldsymbol{\Sigma}_k$-set*. (For background see e.g. [9].)

Recall that a function $\phi : \Sigma_A^\omega \longrightarrow \Sigma_B^\omega$ is continuous if every inverse image of an open set is open. In other words, for any $W_B \subseteq \Sigma_B^*$ there exists some $W_A \subseteq \Sigma_A^*$ such that $x \in W_A \cdot \Sigma^\omega \iff \phi(x) \in W_B \cdot \Sigma^\omega$. Now, given $A \subseteq \Sigma_A^\omega$ and $B \subseteq \Sigma_B^\omega$, we say A continuously reduces to B (denoted $A \leq_W B$ since originally studied by Wadge [15]) if there is a continuous mapping ϕ such that $x \in A \iff \phi(x) \in B$. This ordering should be regarded as a measure of topological complexity. Intuitively $A \leq_W B$ means that A is less complicated than B with regard to the topological structure.

One among many properties about this ordering is that for each integer n, if A is *$\boldsymbol{\Sigma}_n$-complete* (i.e. both $A \in \boldsymbol{\Sigma}_n$ and $B \leq_W A$ holds for all $B \in \boldsymbol{\Sigma}_n$), then A is a true $\boldsymbol{\Sigma}_n$-set, which means it does not belong to $\boldsymbol{\Pi}_n$.

The main device in working with this measure of complexity is a game that links the existence of a winning strategy for a player to the existence of a continuous function that witnesses the relation $A \leq_W B$:

Definition 1 (Wadge game) *Given $A \subseteq \Sigma_A^\omega$, $B \subseteq \Sigma_B^\omega$, $\mathbf{W}(A\ ,\ B)$ is an infinite two player game between players I and II where players take turns, I plays letters in Σ_A, and II plays finite words over the alphabet Σ_B. At the end of an infinite play (in ω moves), I has produced an ω-sequence $x \in \Sigma_A^\omega$ of letters and II has produced an ω-sequence of finite words which concatenated give rise to a finite or ω-word $y \in \Sigma_B^* \cup \Sigma_B^\omega$. The winning condition on the resulting play, denoted here $x\hat{\ }y$, is the following:*

$$\textit{II} \text{ wins the play } x\hat{\ }y \quad \Longleftrightarrow_{def} \ y \textit{ is infinite } \wedge \ (x \in A \longleftrightarrow y \in B)$$

Proposition 2 ([15]) *II has a winning strategy in $\mathbf{W}(A\ ,\ B) \iff A \leq_W B$.*

Example 3 Consider the set $\mathbb{J}$ of all infinite words over the alphabet $\{0,1\}$ that have infinitely many 0. We show that $\mathbb{J}$ is $\boldsymbol{\Pi}_2$-complete.
To verify $\mathbb{J} \in \boldsymbol{\Pi}_2$ we note that the complement $\mathbb{J}^c$ belongs to $\boldsymbol{\Sigma}_2$:

$$x \notin \mathbb{J} \iff x \in \bigcup_{n \in \mathbb{N}} \{0,1\}^n \cdot 1^\omega \ .$$

Note that $\{0,1\}^n \cdot 1^\omega$ is the complement of $\{0,1\}^n \cdot \{0,1\}^* \cdot 0 \cdot \{0,1\}^\omega$ and hence a $\boldsymbol{\Pi}_1$-set, whence $\bigcup_{n\in\mathbb{N}}\{0,1\}^n \cdot 1^\omega$ is a $\boldsymbol{\Sigma}_2$-set. To show $\boldsymbol{\Pi}_2$-completeness, let A be any set in $\boldsymbol{\Pi}_2$, $A = \bigcap_{n\in\mathbb{N}} W_n \cdot \Sigma^\omega$, with $W_n \subseteq \Sigma^*$. We describe a winning strategy for player *II* in the game $\mathbf{W}(A\ ,\ \mathbb{J})$:
Set $i := 0$,
do
  if I's current position u does not have any prefix in W_i,
  then play the letter 1, i remains the same,
  else play the letter 0, $i := i + 1$,
od

Clearly, this strategy is winning for *II* since it induces an infinite word y that contains infinitely many 0 if and only if the infinite word x played by I belongs to each and every open set $W_n \cdot \Sigma^\omega$; hence $x \in A \iff y \in \mathbb{J}$.

3 Pushdown Automata with a $\boldsymbol{\Sigma}_3$-Acceptance condition

We consider *deterministic* pushdown automata of the form $\mathcal{P} = (\Sigma, \Gamma, Q, \delta, q_i)$, where Σ is the finite input alphabet, Γ is the finite stack alphabet, Q is the set of control states, q_i is the initial state, and δ is the partial transition function from $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma$ to $Q \times \Gamma^*$ with the usual restriction on choice between ε-move and Σ-moves (for all $q \in Q$ and $\alpha \in \Gamma$, either $\delta(q, \varepsilon, \alpha)$ is undefined and $\forall a \in \Sigma\ \ \delta(q, a, \alpha)$ is defined, or $\delta(q, \varepsilon, \alpha)$ is defined and $\forall a \in \Sigma\ \ \delta(q, a, \alpha)$ is undefined). A configuration (or "global state") is a pair $(q, w) \in Q \times \Gamma^*$, often written as the word qw, consisting of control state q and stack content w.

Given $a \in \Sigma \cup \{\varepsilon\}$; $q, q' \in Q$; $\mu, \nu \in \Gamma^*$; $\alpha \in \Gamma$; we write $a:\ (q, \alpha \cdot \mu) \underset{\mathcal{P}}{\vdash} (q', \nu \cdot \mu)$ if $\delta(q, a, \alpha) = (q', \nu)$. Finally we denote the transitive closure of $\underset{\mathcal{P}}{\vdash}$ by $\overset{*}{\underset{\mathcal{P}}{\vdash}}$. So $u:\ (q, \nu) \overset{*}{\underset{\mathcal{P}}{\vdash}} (q', \nu')$ holds if the input word u leads $\mathcal{P}$ from the configuration (q, ν) to (q', ν').

Let us equip these pushdown automata with the following acceptance condition: $\mathcal{P}$ accepts $x \in \Sigma^\omega$ iff

$$\exists q \in Q\ \exists \mu \in \Gamma^*\ \forall n\ \exists\ m > n \quad x \restriction m:\ (q_i, \perp) \overset{*}{\underset{\mathcal{P}}{\vdash}} (q, \mu)\ ,$$

(where $x \restriction m$ is the initial segment of x up to position m) and let $L(\mathcal{P})$ be the set of words $x \in \Sigma^\omega$ accepted by $\mathcal{P}$. To say it in words, x is accepted if there is a configuration that occurs infinitely many times while reading x. Or, considering the fact both Q and Γ are finite, a word x is accepted by $\mathcal{P}$ iff, while reading x, for some n the stack content goes back infinitely many times to a word of length n.

By its very definition, it is easy to see that $L(\mathcal{P})$ belongs to $\boldsymbol{\Sigma}_3$: let $A_{q,\mu,n}$ denote the set of finite words u of length precisely n such that, after reading u (from the initial configuration), $\mathcal{P}$ is in configuration (q, μ). We have

$$L(\mathcal{P}) = \underbrace{\bigcup_{\substack{q \in Q \\ \mu \in \Gamma^*}} \underbrace{\bigcap_{n\in\mathbb{N}} \underbrace{\bigcup_{k\in\mathbb{N}} \underbrace{A_{q,\mu,n+k} \cdot \Sigma^\omega}_{\in \boldsymbol{\Sigma}_1 \cap \boldsymbol{\Pi}_1}}_{\in \boldsymbol{\Sigma}_1}}_{\in \boldsymbol{\Pi}_2}}_{\in \boldsymbol{\Sigma}_3}$$

Let us verify that this representation cannot be improved w.r.t. nesting of $\boldsymbol{\Sigma}$ and $\boldsymbol{\Pi}$.

Proposition 4 *There exists a DPDA $\mathcal{P}$ such that $L(\mathcal{P})$ is $\boldsymbol{\Sigma}_3$-complete.*

Proof of proposition 4: We consider a DPDA $\mathcal{P}$ which adds a "0" on top of the stack when it reads a 0, and when it reads 1 it deletes one letter, unless the stack is already empty, in which case it does nothing. Formally, let $\mathcal{P} = (\Sigma,\ \Gamma,\ Q,\ \delta,\ q)$ be the DPDA defined by $\Sigma = \{0, 1\}$, $\Gamma = \{\perp, 0\}$, $Q = \{q\}$, and δ fixed as follows:

- $\delta(q, 0, \perp) = (q, 0 \cdot \perp)$
- $\delta(q, 1, \perp) = (q, \perp)$
- $\delta(q, 0, 0) = (q, 0 \cdot 0)$
- $\delta(q, 1, 0) = (q, \epsilon)$

The figure shows the configuration graph of $\mathcal{P}$:

$$\begin{tikzcd} q\perp \arrow[loop left, "1"] \arrow[r, bend left, "0"] & q\,0\perp \arrow[l, bend left, "1"] \arrow[r, bend left, "0"] & q\,00\perp \arrow[l, bend left, "1"] \arrow[r, bend left, "0"] & q\,000\perp \arrow[l, bend left, "1"] \arrow[r, bend left, "0"] & \cdots \arrow[l, bend left, "1"] \end{tikzcd}$$

In order to prove that $L(\mathcal{P})$ is $\boldsymbol{\Sigma}_3$-complete, we need to show that for any $A \in \boldsymbol{\Sigma}_3$ the relation $A \leq_W L(\mathcal{P})$ holds. For this purpose, let A be a subset of Σ^ω such that $A = \bigcup_{n \in \mathbb{N}} A_n$ where each A_n belongs to $\boldsymbol{\Pi}_2$. Let $\mathbb{J}$ be the $\boldsymbol{\Pi}_2$-complete set defined above in Example 3. For each n, let $\boldsymbol{\sigma}_n$ be a winning strategy for *II* in the game $\mathbf{W}(A_n\ ,\ \mathbb{J})$. Let also $\phi: \mathbb{N} \longrightarrow \mathbb{N} \times \mathbb{N}$ be any bijection that satisfies $\phi(k) = (n, m) \Longrightarrow m \leq k$. We describe a winning strategy for *II* in $\mathbf{W}(A\ ,\ L(\mathcal{P}))$. We write $x_0, x_1, x_2, \ldots$ for the letters chosen by I and $y_0, y_1, y_2, \ldots$ for the finite words chosen by *II*.
Assume $\phi(k) = (n, m)$. Then player *II*'s k^{th} move y_k is defined as follows:

- if $\boldsymbol{\sigma}_n(x_0, x_1, \ldots, x_m)$ contains the letter 0, then y_k is the *shortest* sequence of 0 or 1 such that

$$y_0 \cdot y_1 \cdot \dots \cdot y_k : \ (q, \bot) \overset{*}{\underset{\mathcal{P}}{\vdash}} (q, 0^n \cdot \bot)$$

- if $\boldsymbol{\sigma}_n(x_0, x_1, \ldots, x_m)$ does not contain 0, then $y_k = 0$

This strategy is well defined since $m \leq k$ always holds, therefore $(x_0, x_1, \ldots, x_m)$ is a subsequence of $(x_0, x_1, \ldots, x_k)$. This strategy is winning for *II* because, if I and *II* have respectively played x and y, we can we verify that $x \in A$ iff $y \in L(\mathcal{P})$ as follows:

If $x \in A$ then $x \in A_n$ for some n. Since $\boldsymbol{\sigma}_n$ is winning for *II* in $\mathbf{W}(A_n\ ,\ \mathbb{J})$, there exist infinitely many m such that $\boldsymbol{\sigma}_n(x_0, x_1, \ldots, x_m)$ contains 0. Therefore, by construction, the word $0^n \cdot \bot$ appears infinitely many times as stack content. Thus $y \in L(\mathcal{P})$.

If $x \notin A$ then $x \notin A_n$ holds for any n. Since $\boldsymbol{\sigma}_n$ is winning for *II* in $\mathbf{W}(A_n\ ,\ \mathbb{J})$, there exist only finitely many m such that $\boldsymbol{\sigma}_n(x_0, x_1, \ldots, x_m)$ contains 0. So, for each integer n let k_n be the smallest integer such that

$$\forall k \geq k_n \forall i \leq n \ \ \forall m \in \mathbb{N} \ \Big(\phi(k) = (i, m) \implies \boldsymbol{\sigma}_i(x_0, x_1, \ldots, x_m) \text{ contains no } 0\Big).$$

By construction, after $k + n$ moves, no word $0^i \cdot \bot$ for any $i \leq n$ will appear as stack content. This shows that any configuration of $\mathcal{P}$ occurs only finitely many times, hence $y \notin L(\mathcal{P})$. ⊣

In the next section we are more interested in the set $\mathcal{R}(\mathcal{P})$ of successful runs of $\mathcal{P}$ than in $L(\mathcal{P})$. Let us note that also $\mathcal{R}(\mathcal{P})$ is a true $\boldsymbol{\Sigma}_3$-set:

Proposition 5 *Let $\mathcal{P}$ be as in the preceding proposition. Then the set $\mathcal{R}(\mathcal{P}) \subseteq (Q \cdot \Gamma^*)^\omega$ of accepting runs of $\mathcal{P}$ is $\boldsymbol{\Sigma}_3$-complete.*

Proof of proposition 5: It is easy to see that $\mathcal{R}(\mathcal{P}) \in \boldsymbol{\Sigma}_3$; see the explanation of the acceptance condition in the introduction. In order to verify $\boldsymbol{\Sigma}_3$-completeness, consider the function $\phi : \Sigma^\omega \longrightarrow (Q \cdot \Gamma^*)^\omega$ which associates to $x \in \Sigma^\omega$ the $\mathcal{P}$-run ρ_x on x. Obviously, ϕ is continuous (one does not even need the Wadge game to verify this), and we have $x \in L(\mathcal{P}) \Longleftrightarrow \rho_x \in \mathcal{R}(\mathcal{P})$. Thus $L(\mathcal{P}) \leq_W \mathcal{R}(\mathcal{P})$. ⊣

It should be noted that for nondeterministic pushdown automata the situation is much different: As shown by Finkel [6,7], nondeterministic pushdown automata equipped with the Büchi acceptance condition can recognize Borel-sets of any finite rank and even non-Borel sets.

4 Effective Solvability

4.1 Outline

In the present section we use pushdown automata for the specification of infinite games (between two players 0 and 1) rather than for the definition of ω-languages. The acceptance condition considered in the previous section is now employed as a winning condition for Player 0. Our aim is to show that for any such pushdown game one can compute the winning region of Player 0 (the set of those configurations from which Player 0 can force a win) and, moreover, a positional winning strategy.

Let us first introduce the game-theoretic setting. A pushdown game graph is specified by a variant of the pushdown automata considered in the previous section, which we call *pushdown game systems.* The input alphabet Σ and the initial state q_0 are canceled, but a partition $Q = Q_0 \uplus Q_1$ of the state set Q into sets Q_0, Q_1 is introduced. Note that by the deletion of Σ the transitions become unlabeled, and thus there is not a deterministic transition function any more but a transition relation: A *pushdown game system* (PDS) is of the form $\mathscr{P} = (\Gamma, Q_0, Q_1, \Delta)$, where Γ is the finite stack alphabet, $Q = Q_0 \uplus Q_1$ the finite state set, and $\Delta \subseteq Q \times \Gamma \times Q \times \Gamma^*$ the finite transition relation. Of course, given a pushdown game system one may obtain a normal DPDA by introducing an initial state and a sufficiently large input alphabet Σ, which would allow to regain a deterministic (partial) transition function.

A pushdown game system $\mathscr{P}$ determines a *pushdown game graph* $G_{\mathscr{P}} = (V, E)$ with vertex set $V = Q\Gamma^*$ and the edge set E consisting of the pairs $(p\gamma\mu, q\nu\mu) \in V \times V$ such that $(p, \gamma, q, \nu) \in \Delta$. Define $V_0 = Q_0\Gamma^*$ and $V_1 = Q_1\Gamma^*$. A *play* over (V, E) from $v \in V$ is a sequence $u_0, u_1, u_2, \cdots$ built up by the two players 0,1 as follows: We have $u_0 = v$; given $u_i \in V_0$, Player 0 chooses u_{i+1} such that $(u_i, u_{i+1}) \in E$, and given $u_i \in V_1$, Player 1 chooses u_{i+1} with $(u_i, u_{i+1}) \in E$. The play is won by Player 0 iff

there is a configuration from V that appears infinitely often in the play, (1)

equivalently, iff for some length n a configuration of length n is visited infinitely often. Our aim is to compute the set W_0 of winning positions of Player 0: the positions from which he can win whatever Player 1 does.

As a preparatory step, we recall the definition of winning regions of somewhat simpler games: reachability games, where Player 0 has to reach a configuration of a given "target set" T just *once* in order to win, and Büchi games where Player 0 has to ensure that infinitely often configurations in T are visited. We recall the corresponding definitions (see, e.g., [13]) which rely on the fact that our game graphs are of bounded degree. Given a set $T \subseteq V$, the 0-attractor of

T is the set of configurations from which Player 0 can force the play to reach T. It is inductively defined by:

$$\begin{aligned} Attr_0^0(T) &= T \ , \\ Attr_0^{i+1}(T) &= Attr_0^i(T) \cup \{u \in V_0 \mid \exists v, (u,v) \in E, \ v \in Attr_0^i(T)\} \\ &\quad \cup \{u \in V_1 \mid \forall v, (u,v) \in E \Rightarrow v \in Attr_0^i(T)\} \ , \\ Attr_0(T) &= \textstyle\bigcup_{i \in \mathbb{N}} Attr_0^i(T) \ . \end{aligned}$$

Here $Attr_0^i(T)$ is the set of configurations from which Player 0 can force a visit in T in at most i steps. If we slightly modify the definition, we get $Attr_0^+(T)$: the set of configurations from which Player 0 can force the play to reach T in *at least* one move, whatever Player 1 does.

$$\begin{aligned} X_0(T) &= \emptyset \ , \\ X_{i+1}(T) &= X_i(T) \cup \{u \in V_0 \mid \exists v, (u,v) \in E, \ v \in T \cup X_i(T)\} \\ &\quad \cup \{u \in V_1 \mid |u| > 1, \ \forall v, (u,v) \in E \Rightarrow v \in T \cup X_i(T)\} \ , \\ Attr_0^+(T) &= \textstyle\bigcup_{i \geqslant 0} X_i(T) \ . \end{aligned}$$

For technical reasons concerning the definition of $Attr_0^+(T)$, it is convenient to allow deadlocks by the empty stack in the game graph and to declare here Player 1 as the winner of any play terminating with empty stack.

We are now able to define $\text{Büchi}_0(T)$, the set of those configurations from which Player 0 can force to reach T infinitely many times (to win the "Büchi game for T"):

$$\begin{aligned} \text{Büchi}_0^0(T) &= V \ , \\ \text{Büchi}_0^{i+1}(T) &= Attr_0^+(\text{Büchi}_0^i(T) \cap T) \ , \\ \text{Büchi}_0(T) &= \textstyle\bigcap_{i \in \mathbb{N}} \text{Büchi}_0^i(T) \ . \end{aligned}$$

We note $\Gamma^{\leqslant M}$ the language $\{\epsilon\} \cup \Gamma^1 \cup \cdots \cup \Gamma^M$. The effective solution of pushdown games with winning condition (1) is based on the following straightforward representation of the winning region W_0 of player 0:

Proposition 6 *Over a game graph induced by a pushdown game system, the winning region W_0 of Player 0 w.r.t. winning condition (1) is* $W_0 = \bigcup_{M>0} \mathit{Büchi}_0(Q\Gamma^{\leqslant M})$.

Let us refine this into an algorithmic description of W_0. In [4] it is shown that if the set T is regular (the configurations of the pushdown game graph are considered as words), then one can compute a finite automaton recognizing $Attr_0(T)$, respectively $Attr_0^+(T)$, which hence are again regular. Using the regularity of $Attr_0^+(T)$ one can compute a finite automaton recognizing $\text{Büchi}_0(T)$. Of course $\Gamma^{\leqslant M}$ is regular for $M \geqslant 0$, so $\text{Büchi}_0(Q\Gamma^{\leqslant M})$ can be computed. To compute the set W_0 of Proposition 6, we finally have to overcome the problem that W_0 is an infinite union. We shall prove that

$$W_0 = Attr_0(\text{Büchi}_0(Q\Gamma^{\leqslant N}) \bullet \Gamma^*)$$

where $N = 1 + |\Gamma||Q| \max\{|\nu| - 1 \mid (p, \gamma, q, \nu) \in \Delta\}$, and the set $\text{Büchi}_0(Q\Gamma^{\leqslant N}) \bullet \Gamma^*$ will be defined later (it is almost $\text{Büchi}_0(Q\Gamma^{\leqslant N}) \cdot \Gamma^*$). The idea is that if

Player 1 can make the stack increase by more than N letters, then he can make it increase indefinitely (without returning to previous stack contents an unbounded number of times) and thus wins.

4.2 Details

We first recall the constructions of [4]. Given a regular set T of configurations, it is recognized by a finite automaton $\mathscr{A}_T$ over the alphabet $Q \uplus \Gamma$. Then a finite construction, originally presented in [1] in the framework of alternating pushdown systems, transforms $\mathscr{A}_T$ into $\mathscr{A}_{Attr(T)}$, an *alternating* finite automaton that recognizes $Attr_0(T)$. The state space remains the same during the construction, the algorithm just adds new transitions. By an obvious modification of the algorithm, it is possible to construct a finite automaton $\mathscr{A}_{Attr+(T)}$, recognizing $Attr_0^+(T)$.

We describe here the format of these automata and explain how to use them for the construction of an automaton recognizing $\text{Büchi}_0(T)$. The automata to recognize sets of configurations are alternating finite word automata with a special convention about initial states: Given a PDS $\mathscr{P} = (\Gamma, Q_0, Q_1, \Delta)$, a $\mathscr{P}$-*automaton* is a finite automaton $\mathscr{A} = (P, \Gamma, \longrightarrow, Q, F)$, where $P \supseteq Q$ is its finite set of states, $\longrightarrow \;\subseteq\; P \times (\Gamma \cup \{\epsilon\}) \times 2^P$ the set of transitions, $Q \subseteq P$ the set of initial states (note that these are the control locations of $\mathscr{P}$), and $F \subseteq P$ a set of final states. A transition $r \xrightarrow{\gamma} S$ indicates a move from state r via letter $\gamma \in \Gamma$ simultaneously to all states of S, i.e. by a universal branching of runs. Existential branchings are captured by nondeterminism. (So, a transition like $r \xrightarrow{\gamma} (r_1 \wedge r_2) \vee (r_3 \wedge r_4)$ is represented here by *two* transitions $r \xrightarrow{\gamma} \{r_1, r_2\}$ and $r \xrightarrow{\gamma} \{r_3, r_4\}$.) For each $p \in P$ and $w \in \Gamma^*$, the automaton $\mathscr{A}$ accepts a configuration $pw \in Q\Gamma^*$ iff there exists a successful $\mathscr{A}$-run on w from the initial state p. Successful runs are defined in the standard way, using computation trees for the representation of simultaneously active states; the acceptance condition requires that some computation tree exists which at every leaf ends in a final state. By $q \xrightarrow{w}_* S$ we indicate that such a computation tree exists on input qw such that its leaf states form the set S.

Let us explain the transformation of a $\mathscr{P}$-automaton $\mathscr{A}$ recognizing T into a $\mathscr{P}$-automaton recognizing $\text{Büchi}_0(T)$. We consider the case $T = Q\Gamma^{\leqslant M}$ for a given number M and set

$$Y_0^M = Q\Gamma^{\leqslant M} \ , \ Y_{i+1}^M = Attr_0^+(Y_i^M) \cap Q\Gamma^{\leqslant M} \ , \text{ and } \ Y_\infty^M = \bigcap_{i \geqslant 0} Y_i^M \ .$$

Then $\text{Büchi}_0(Q\Gamma^{\leqslant M}) = Attr_0(Y_\infty^M)$.

In the sequel the relation E is written in infix-notation with the symbol "$\hookrightarrow$": so we have $(u, v) \in E \iff u \hookrightarrow v$ and also $(p, \gamma, q, \nu) \in \Delta \iff p\gamma \hookrightarrow q\nu$. Consider the PDS $\mathscr{P} = (\Gamma, Q_0, Q_1, \Delta)$ with $Q = Q_0 \cup Q_1$. The construction of the automaton recognizing Y_∞^M starts with a $\mathscr{P}$-automaton $\mathscr{B}_0$ which recognizes $Q\Gamma^{\leqslant M}$: its state set is $Q \cup \{f_0, \cdots, f_M\}$, with transitions $f_i \xrightarrow{\Gamma} f_{i+1}$ for $i < M$, each f_i being a final state, and the states of $Q \cup \{f_0\}$ are merged into a unique state named f_0, *i.e.*, f_0 is initial.

In stages or "generations" $i = 1, 2, 3, \cdots$ new copies of Q are added. We write (q, i) or short q^i for the copy of a node $q \in Q$ added in stage i. So the state space will by a subset of $(Q \times \mathbb{N}) \cup \{f_0, \ldots, f_M\}$ (where $q^0 = f_0$ for all $q \in Q$). We write Q^i for the set $Q \times \{i\}$. Two auxiliary operations are needed which refer to this indexing by stages:

Definition 7 *For a finite set $S \subseteq (Q \times \mathbb{N}) \cup \{f_0, \ldots, f_M\}$ let*

$$\phi(S) = \{q^i \mid q^{i+1} \in S\} \ \cup \ (S \cap \{f_0, \cdots, f_M\}) \ ,$$

with the convention that q^0 is f_0 for all q.

Definition 8 *For $i > 0$ and a set $S \subseteq (Q \times [1, i]) \cup \{f_0, \cdots, f_M\}$, let*

$$\pi^i(S) = \{q^i \mid \exists i \geqslant k > 0, q^k \in S\} \cup (S \cap \{f_0, \cdots, f_M\}) \ .$$

This is the projection of the set S on the generation i (except for $\{f_0, \cdots, f_M\}$).

Algorithm 9 *To compute an automaton recognizing Y_∞^M*
Input: *PDS $\mathscr{P} = (\Gamma, Q_0, Q_1, \Delta)$ and $M > 0$*
Output: *a $\mathscr{P}$-automaton $\mathscr{C}$ that recognizes Y_∞^M*

Initialization: Set $\mathscr{C} := \mathscr{B}_0$ recognizing $Q\Gamma^{\leqslant M} = Z_0$, with states q^0 (for $q \in Q$) and $f_0, \ldots, f_M$, where for all $q \in Q$, q^0 is set to be f_0. (Recall that for all $\gamma \in \Gamma, f_i \xrightarrow{\gamma} f_{i+1}$, and the f_i's are the final states.)
$i := 0$.

repeat
  $i := i + 1$ (i is number of the current generation)
  Add the states q^i, for each $q \in Q$, using them as initial states.
  Add an ϵ-transition from q^i to q^{i-1} for each $q \in Q$
  {obtain an automaton still recognizing Z_{i-1} }
  Add new transitions to $\mathscr{C}$ by the saturation procedure presented in [4]:
  repeat
    (Player 0) if *$p \in Q_0$, $p\gamma \hookrightarrow q\mu \in \Delta$ and $q^i \xrightarrow{\mu}_* S$ in the current automaton,* then *add a new transition $p^i \xrightarrow{\gamma} S$.*
    (Player 1) if *$p \in Q_1$, $\{p\gamma \hookrightarrow q_1\mu_1, \cdots, p\gamma \hookrightarrow q_n\mu_n\}$ are* all *the Δ-rules (game moves) starting from $p\gamma$ and $\forall k$, $q_k^i \xrightarrow{\mu_k}_* S_k$ in the current automaton,* then *add a new transition $p^i \xrightarrow{\gamma} \bigcup_k S_k$.*
  until *no new transition can be added*
  { the obtained automaton recognizes $Attr_0(Z_{i-1})$ }
  remove the ϵ-transitions.
  { obtain $\mathscr{B}_i'$ recognizing $Attr_0^+(Z_{i-1}) = Z_i'$ }
  replace each transition $q^i \xrightarrow{\gamma} S$ by $q^i \xrightarrow{\gamma} \pi^i(S)$.
  { obtain $\mathscr{B}_i''$ recognizing $Z_i'' \subseteq Z_i'$ }
  replace each transition $q^i \xrightarrow{\gamma} S$ by $q^i \xrightarrow{\gamma} S \cup \{f_0\}$
  { obtain $\mathscr{B}_i$ recognizing $Z_i'' \cap Q\Gamma^{\leqslant M} = Z_i$, we have $\bigcap_{i \geq 0} Y_i^M \subseteq Z_i$ }
  set $\mathscr{C} := \mathscr{B}_i$, finishing generation number i

until $i > 1$ *and* $\forall p, \gamma :\ p^i \xrightarrow{\gamma} S \iff p^{i-1} \xrightarrow{\gamma} \phi(S)$.

Note that we can erase the q^{i-1}'s and their transitions as soon as the generation i is done. To compare successive generations we have the following property.

Proposition 10 *In Algorithm 9, for all* $\nu \in \Gamma^*, q \in Q, i \geqslant 1$ *we have*

$$q^{i+1} \xrightarrow{\nu}_* S \Rightarrow q^i \xrightarrow{\nu}_* \phi(S)$$

The proofs of this proposition and of the following theorem are similar to the corresponding claims in [4]; for completeness they are given in the appendix. Note that because of the projection π, the transitions $q^i \xrightarrow{\nu}_* S$ verify $S \subseteq (Q \times \{i\}) \cup \{f_0, \cdots, f_M\}$. Note also that no new transition from the states $f_0, \cdots, f_M$ is added.

Theorem 11 *The automaton* $\mathscr{C}$ *constructed in Algorithm 9 recognizes* Y_∞^M.

It remains to eliminate the quantification on M implicit in $\bigcup_{M>0} \text{Büchi}_0(Q\Gamma^{\leqslant M})$, by choosing a sufficiently large bound for M. We introduce an ordering relation which permits to compare transitions.

Definition 12 *For any* $S, S' \subseteq Q^i \cup \{f_0, \cdots, f_M\}$,

$$S \sqsubseteq S' \ \Leftrightarrow \begin{cases} S \cap Q^i \ \subseteq \ S' \cap Q^i \ \textit{and} \\ \max(\{j \mid f_j \in S\} \cup \{-1\}) \leqslant \max(\{j \mid f_j \in S'\} \cup \{-1\}) \end{cases}$$

The idea is that in case $S \sqsubseteq S'$, one recognizes "more" after a transition $q^i \longrightarrow S$ than after a transition $q^i \longrightarrow S'$. To compare transitions $q^i \longrightarrow S$ and $q^j \longrightarrow S'$, with $i < j$, one considers $\pi_j(S)$ and S' with respect to $\sqsubseteq$. The index j of $f_j \in S$ measures the possibility for Player 1 to increase the length of the stack, and possibly win.

Proposition 13 *In the automaton* $\mathscr{C}$ *constructed by Algorithm 9, assume that for a transition* $q^i \xrightarrow{\gamma} S$ *we have* $S \sqsubseteq S'$ *for each transition* $q^i \xrightarrow{\gamma} S'$ *from the same state. If* $\ell = \max\{j \mid f_j \in S\} \geqslant 0$*, then from the configuration* $q\gamma$*, Player 1 has a strategy to reach a configuration where the length of the stack is at least* ℓ and *such that this* ℓ *letters will never be "popped".*

Proof of proposition 13: Induction on the number of transitions constructed by the algorithm. Note that the projection π^i does not change the value of ℓ. If $\ell = 0$, the property is trivially true.

In the generation number 1, there is an ϵ-transition from each q^i to f_0. If for some q, γ and ℓ we have the hypotheses of the proposition, then it follows that:
- either $q \in Q_0$ and every transition $q\gamma \hookrightarrow q'\mu \in \Delta$ add at least ℓ letters to the stack, moreover Player 0 has no possibility to decrease the stack length from $q'\mu$ (otherwise we would have a path $q^i \xrightarrow{\mu}_* S$, and another transition added),
- or $q \in Q_1$ and there is a transition $q\gamma \hookrightarrow q'\mu \in \Delta$ that adds at least ℓ letters to the stack, moreover Player 0 has no possibility to decrease the stack length from $q'\mu$ (otherwise we would have a path $q'^i \xrightarrow{\mu}_* S$, and another transition added).

For the next generations we have the same argument, relying on the preceding generations. If $q \in Q_1$ then there is a transition $q\gamma \hookrightarrow q'\gamma'\nu \in \Delta$ such that from $q'\gamma'$ Player 1 can reach a configuration with ℓ_1 letters (applying the proposition to the previous generation) and $\ell = \ell_1 + |\nu|$. And similarly for $q \in Q_0$. $\dashv$

We consider now $N = 1 + |\Gamma||Q| \max\{|\mu| - 1 \mid \exists\, p\gamma \hookrightarrow q\mu \in \Delta\}$. The rightmost factor is the maximal number of letters that can be added to the stack in one move.

Proposition 14 *In the automaton $\mathscr{C}$ constructed by Algorithm 9, assume again that for a transition $q^i \xrightarrow{\gamma} S$ we have $S \sqsubseteq S'$ for each transition $q^i \xrightarrow{\gamma} S'$ from the same state. If $\ell = \max\{j \mid f_j \in S\} \geqslant N$, then from configuration $q\gamma$, Player 1 can win the game by increasing the stack indefinitely.*

Proof of proposition 14: According to the previous proposition, Player 1 can ensure the stack increases by at least ℓ letters, that will *never* be poped. Using an argument similar to the classical pumping argument (see e.g. [8]), there exists $(q, \alpha) \in Q \times \Gamma$ such that, during this process, two different configurations $q\alpha\nu$ and $q\alpha\xi\nu$ are met ($\nu \in \Gamma^*, \xi \in \Gamma^+$), and the letters of ν and ξ are not scanned (nor changed) any more in the stack after these configurations. This proves that continuing from $q\alpha\xi\nu$, Player 1 can force the stack to increase indefinitely. This shows that a configuration in $q\gamma\Gamma^*$ cannot be in the winning region W_0 of Player 0. $\dashv$

It follows from the proposition that in $\mathscr{C}$ we can eliminate transitions $q^i \longrightarrow S$, such that $f_j \in S, j > N$. So intuitively the computation of Y_∞^N is sufficient to determine Y_∞^M for all $M \geqslant N$. We have clearly $Y_\infty^N \subseteq W_0$, and also $Y_\infty^N \cdot \Gamma^* \subseteq W_0$ because in the computation of Y_∞^N we have assumed that if the stack is empty, Player 0 loses. Hence $\mathit{Attr}_0(Y_\infty^N \Gamma^*) \subseteq W_0$, but the equality does not hold in general.

In the automaton obtained from Algorithm 9 for a given $M > 0$ the next step is to merge the states f_k to a unique final state f, and to add a transition $f \xrightarrow{\gamma} f$ for all $\gamma \in \Gamma$, to obtain an automaton $\mathscr{C}'$. We will note $Y_\infty^M \bullet \Gamma^*$ the set of configurations recognized by $\mathscr{C}'$. Nevertheless the set $Y_\infty^M \bullet \Gamma^*$ is defined only by the algorithm, and there is no language theoretical definition of $Y_\infty^M \bullet \Gamma^*$ from Y_∞^M. We have $Y_\infty^M \cdot \Gamma^* \subseteq Y_\infty^M \bullet \Gamma^*$ but the equality does not hold in general, because the automaton is alternating. Now we can prove the following.

Corollary 15 *For all $M \geqslant N$, $\quad Y_\infty^N \bullet \Gamma^* = Y_\infty^M \bullet \Gamma^*$.*

Proof of corollary 15: The inclusion from left to right is clear. For the other inclusion, the automaton recognizing $Y_\infty^M \bullet \Gamma^*$ "contains" that of $Y_\infty^N \bullet \Gamma^*$. It has possibly some other transitions $q^i \longrightarrow S$, with $f_j \in S, j > N$, which verify the hypotheses of Proposition 14. Those transitions do not permit to accept a configuration in W_0, *i.e.*, no winning play from such a configuration is possible. But clearly $Y_\infty^M \bullet \Gamma^* \subseteq \bigcup_{M>0} \text{Büchi}_0(Q\Gamma^{\leqslant M}) \subseteq W_0$ (a play from Y_∞^M is also possible from $Y_\infty^M \bullet \Gamma^*$. See Proposition 6). $\dashv$

Theorem 16 *Given a pushdown game system, one can compute a finite automaton recognizing the winning region*

$$W_0 = Attr_0(Y_\infty^N \bullet \Gamma^*)$$

of Player 0 w.r.t. the $\boldsymbol{\Sigma}_3$*-winning condition (1).*

Proof of theorem 16: Clearly $Attr_0(Y_\infty^N \bullet \Gamma^*) \subseteq W_0$. Proposition 6 states that

$$W_0 = \bigcup_{M>0} \text{Büchi}_0(Q\Gamma^{\leqslant M}) \ ,$$

which is, by the preceding proposition,

$$\bigcup_{M>0} Attr_0(Y_\infty^M) \subseteq \bigcup_{M>0} Attr_0(Y_\infty^M \bullet \Gamma^*) \subseteq Attr_0(Y_\infty^N \bullet \Gamma^*) \ .$$

⊣

The construction of an automaton recognizing $W_0 = Attr_0(Y_\infty^N \bullet \Gamma^*)$ works as follows: one uses Algorithm 9 with $M = N$. The resulting automaton $\mathscr{C}$ recognizes Y_∞^N. Now one merges the states f_k to a unique final state f, and one adds a transition $f \xrightarrow{\gamma} f$ for all $\gamma \in \Gamma$, in order to obtain an automaton $\mathscr{C}'$ which recognizes $Y_\infty^N \bullet \Gamma^*$. To recognize $Attr_0(Y_\infty^N \bullet \Gamma^*)$ we just need another application of the saturation procedure as it appears in Algorithm 9, which finally results in an (alternating) automaton $\mathscr{C}''$ which recognizes W_0.

In the definition of $Attr_0$, the "usual" convention about deadlocks (with empty stack) is implicitly assumed. So finally at the end of the computation of $Attr_0(Y_\infty^N \bullet \Gamma^*)$, the states p^{i+1} with $p \in Q_1$ has to be marked as final. They correspond to the possibility for Player 0 to win by reaching a configuration with empty stack when Player 1 is on. We did not assumed this convention in the definition of $Attr_0^+$ because it would have compromised the computation of $\text{Büchi}_0(Q\Gamma^{\leqslant N}) \bullet \Gamma^*$. It is also possible to consider a bottom stack symbol ($\perp$), but it has to be defined explicitly in Γ and Δ and treated as a stack letter.

Following the constructions of [4], it is easy to extract a (positional) winning strategy for player 0 on the set W_0. The choice of an appropriate transition from a game graph vertex $qw \in W_0$ is done by analyzing an accepting run of the automaton $\mathscr{C}'$ on the input qw. For the details we have to refer to [4].

5 Discussion and Concluding Remarks

The $\boldsymbol{\Sigma}_3$-acceptance condition considered above was introduced as an example, illustrating the possibility to reach higher levels of the Borel hierarchy than $B(\boldsymbol{\Sigma}_2)$. For applications in ω-language theory a more general form is appropriate, referring to a set $F \subseteq Q\Gamma^*$: Call a DPDA-run ρ accepting if

$$\exists w \in F \ \forall i \ \exists j \geqslant i \ \rho(j) = w \ . \tag{2}$$

If F is finite, then this condition is equivalent to

$$\forall i \ \exists j \geqslant i \ \exists w \in F \ \rho(j) = w \ , \tag{3}$$

i.e., to the usual Büchi acceptance condition. In order to define an interesting class of ω-languages including true $\boldsymbol{\Sigma}_3$-sets, it is necessary to combine the acceptance conditions (2) and (3). Note that condition (2) alone does not allow to simulate condition (3): For example, the ω-language over $\{0,1,\$\}$ which contains:
– all ω-words over $\{0,1\}$, and
– the ω-words $u\$u^{\mathrm{R}}x$ with $u \in \{0,1\}^*$ and arbitrary $x \in \{0,1,\$\}^*$
is recognizable by a DPDA with the Büchi acceptance condition (3) but not definable with acceptance condition (2).

How can one reach even higher levels of the Borel hierarchy than just $\boldsymbol{\Sigma}_3$? A natural idea is to require infinitely many configurations, each of them being visited infinitely often, as accepting condition:

$$\forall j \ \exists q\mu \in Q\Gamma^j\Gamma^* \ \forall n \ \exists \ m > n \quad x \upharpoonright m : \ (q_i, \bot) \underset{\mathcal{P}}{\overset{*}{\vdash\!\!\!-\!\!\!-}} (q, \mu). \tag{4}$$

Remarkably, this condition comes down to a $\boldsymbol{\Sigma}_3$ condition: it is logically equivalent to the conjunction of our $\boldsymbol{\Sigma}_3$ condition (one configuration is visited infinitely often) and the condition that the stack growth is unbounded:

$$\exists q\mu \in Q\Gamma^* \ \forall n \ \exists \ r, s, t > n \ \exists q'\mu' \in Q\Gamma^s$$
$$\left(x \upharpoonright r : \ (q_i, \bot) \underset{\mathcal{P}}{\overset{*}{\vdash\!\!\!-\!\!\!-}} (q, \mu) \quad \wedge \quad x \upharpoonright t : \ (q_i, \bot) \underset{\mathcal{P}}{\overset{*}{\vdash\!\!\!-\!\!\!-}} (q', \mu') \right) \ .$$

Let us modify (4) by moving slightly the occurrence of control state in the formula:

$$\forall j \ \forall q \in Q \ \exists \mu \in \Gamma^j\Gamma^* \ \forall n \ \exists \ m > n \quad x \upharpoonright m : \ (q_i, \bot) \underset{\mathcal{P}}{\overset{*}{\vdash\!\!\!-\!\!\!-}} (q, \mu). \tag{5}$$

In other words, if we call q-configuration the words of the form $q\mu \in \{q\}\Gamma^*$, we deal with the condition:

for all state q there exists infinitely many
q-configurations that are visited infinitely often,

This can be shown to be a $\boldsymbol{\Pi}_4$-acceptance condition which does not collapse to $\boldsymbol{\Sigma}_3$: it leads to true Π_4 sets. The same holds for the closely related condition where q is fixed

there exists infinitely many q-configurations
that are visited infinitely often.

Acknowledgment

We thank Olivier Serre and the referees for useful remarks.

References

1. A. Bouajjani, J. Esparza, and O. Maler, Reachability analysis of pushdown automata: Application to model-checking, CONCUR '97, LNCS 1243, pp 135-150, 1997.
2. J.R. Büchi, Landweber L.H., *Solving sequential conditions by finite-state strategy.* Transactions of the American Mathematical Society vol. 138 (1969) 295–311
3. J.R. Büchi, State-strategies for games in $F_{\sigma\delta} \cap G_{\delta\sigma}$ J. Symbolic Logic 48 (1983), no. 4, 1171–1198.
4. T. Cachat, Symbolic strategy synthesis for games on pushdown graphs, in: ICALP'02, Springer LNCS (to appear). `http://www-i7.informatik.rwth-aachen.de/~cachat/`
5. E. A. Emerson and C. S. Jutla, *Tree automata, mu-calculus and determinacy*, FoCS '91, IEEE Computer Society Press (1991), pp. 368–377.
6. O. Finkel, Topological properties of omega context-free languages, *Theoret. Comput. Sci.* 262 (2001), no. 1-2, 669–697.
7. O. Finkel, Wadge hierarchy of omega context-free languages, *Theoret. Comput. Sci.* 269 (2001), no. 1-2, 283–315.
8. J. E. Hopcroft and J. D. Ullman, *Formal Languages and their relation to automata*, Addison-Wesley, 1969.
9. A. S. Kechris, *Classical descriptive set theory*, Graduate texts in mathematics, vol 156, Springer Verlag (1994)
10. O. Kupferman and M. Y. Vardi, An Automata-Theoretic Approach to Reasoning about Infinite-State Systems, CAV 2000, LNCS 1855, 2000.
11. S. Seibert, *Effektive Strategiekonstruktionen für Gale-Stewart-Spiele auf Transitionsgraphen*, Technical Report 9611, Institut für Informatik und Praktische Mathematik, Christian-Albrechts-Universität zu Kiel, Germany, July 1996.
12. C. Stirling, *Modal and Temporal Properties of Processes*, Springer (Texts in Computer Science), 2001.
13. W. Thomas, On the synthesis of strategies in infinite games, STACS '95, LNCS 900, pp. 1–13, 1995.
14. W. Thomas, Languages, automata, and logic, in *Handbook of Formal Language Theory* (G. Rozenberg, A. Salomaa, Eds.), Vol 3, Springer-Verlag, Berlin 1997, pp. 389–455.
15. W.W. Wadge *Reducibility and Determinateness on the Baire Space* Ph.D. Thesis, University of California, Berkeley, 1984.
16. I. Walukiewicz, Pushdown processes: games and model checking, CAV '96, LNCS 1102, pp 62-74, 1996. Full version in Information and Computation 164, pp. 234-263, 2001.

Appendix

Proof: Proposition 10
We proceed by induction on i. The proposition is a direct consequence of the same property over the transitions (by induction on the length of the word).

- For $i = 1$, we use another induction on the number of transitions starting from p^2 added by the saturation procedure: *during* the saturation procedure, for each new transition $p^2 \xrightarrow{\gamma} S$, we want to check that $p^1 \xrightarrow{\gamma} \phi(\pi^2(S) \cup f_0)$ (because at the end of generation 2, we will have $p^2 \xrightarrow{\gamma} \pi^2(S) \cup f_0$).
  As a preliminary remark, we observe that once the first generation is done, each path $q^1 \xrightarrow{\mu}_* S$ from a state q^1 is such that $S \subseteq Q \times \{1\} \cup \{f_0, \cdots, f_M\}$, hence $\pi^2(S) \subseteq Q \times \{2\} \cup \{f_0, \cdots, f_M\}$, $\phi(\pi^2(S)) \subseteq Q \times \{1\} \cup \{f_0, \cdots, f_M\}$, and $\phi(\pi^2(S)) = S$. We also know that $f_0 \in S$.
  - At the beginning of the second iteration, one has no other transitions from the states q^2 than $q^2 \xrightarrow{\epsilon} \{q^2\}$ and $q^2 \xrightarrow{\epsilon} \{q^1\}$, which are temporary.
  - *during* the saturation procedure, as a new transition $p^2 \xrightarrow{\gamma} S$ is added, it is through an existing path $q^2 \xrightarrow{\mu}_* S$ (see algorithm).
  If this path uses just ϵ-transitions ($\mu = \epsilon$, $S \subseteq Q \times [1, 2]$), then a similar path form q^1 existed during the first iteration, generating a transition $p^1 \xrightarrow{\gamma} \phi(\pi^2(S))$ (where we choose to stay in $Q \times \{1\}$). Hence at the end of first generation, one gets $p^1 \xrightarrow{\gamma} \pi^1(\phi(\pi^2(S))) \cup f_0$, and $\pi^1(\phi(\pi^2(S))) \cup f_0 = \phi(\pi^2(S)) \cup f_0 = \phi(\pi^2(S) \cup f_0)$. Whereas at the end of the second generation, the transition generated is actually $p^2 \xrightarrow{\gamma} \pi^2(S) \cup f_0$.
  If the first segment of this path $q^2 \xrightarrow{\mu}_* S$ is a "real" transition $q^2 \xrightarrow{\alpha} T$, $\alpha \in \Gamma$, then by induction hypothesis one has also $q^1 \xrightarrow{\mu}_* \phi(\pi^2(S) \cup f_0)$, and the corresponding transition from p^1 was added.
  If the first segment is $q^2 \xrightarrow{\epsilon} q^1 \xrightarrow{\alpha} T$, then similarly, because of $q^1 \xrightarrow{\alpha} T$ already existing, the corresponding transition from p^1 was added.
- induction hypothesis: $\forall S,\ p^i \xrightarrow{\alpha} S \Rightarrow p^{i-1} \xrightarrow{\alpha} \phi(S)$
- the proof for $i + 1$ is similar to the case $i = 1$

■

Proof: Theorem 11
Proof of termination
Thanks to the projections π^i's, there is only bounded number of possible transitions from each row of p^i's. And thanks to Proposition 10, there is less and less transitions until the algorithm reaches a fixed point.

Proof of correctness
We note Z_i the language recognized by $\mathscr{C}$ from the initial states p^i's. We denote $n+1$ the last generation of the algorithm, which is such that $Z_n = Z_{n+1}$, by convention we still consider $Z_i = Z_n$ for all $i \geqslant n$. One has to show that $Z_n = Y_\infty^M$. We consider the intermediate results (stages) of Algorithm 9: near the end of the i-th generation, just after one removes the ϵ-transitions, one gets the automaton $\mathscr{B}'_i$, recognizing Z'_i. Then, just after the projection, one gets $\mathscr{B}''_i$, recognizing

Z_i''. Finally one gets $\mathscr{B}_i$, recognizing Z_i, by replacing each transition $p^i \xrightarrow{\gamma} S$ by $p^i \xrightarrow{\gamma} S \cup \{f_0\}$.

First part: $Z_n \subseteq Y_\infty^M$.
We prove by induction on i that for all i, $Z_i \subseteq Y_i$.

- Remembering that for all $p \in Q$, p^0 is set to be f_0, $Z_0 = Y_0 = Q\Gamma^{\leqslant M}$.
- Induction hypothesis: $Z_i \subseteq Y_i$.
- The algorithm first determines the Attractor$^+$ of the language (see [4]). By monotonicity,

$$Z'_{i+1} = Attr_0^+(Z_i) \subseteq Attr_0^+(Y_i) = Y'_{i+1} \ .$$

  After the projection of the transitions, the obtained language Z''_{i+1} is contained in Z'_{i+1}: Proposition 10 shows that an accepting path from a state p^{i+1} was possible before the projection (through the states p^i). So $Z''_{i+1} \subseteq Z'_{i+1} \subseteq Y'_{i+1}$ (for $i = 0$ the projection does not change any transition). The next operation of the algorithm "computes" the intersection with $Q\Gamma^{\leqslant M}$. Here we just need an inclusion. In the resulting automaton $\mathscr{B}_{i+1}$, each transition from a p^{i+1} has at least a "branch" that goes to $\{f_0, \cdots, f_M\}$, so it is clear that $Z_i \subseteq Q\Gamma^{\leqslant M}$. Also $Z_i \subseteq Z''_{i+1}$, and

$$Z_{i+1} \ \subseteq \ Z''_{i+1} \cap Q\Gamma^{\leqslant M} \ \subseteq \ Y'_{i+1} \cap Q\Gamma^{\leqslant M} \ = \ Y_{i+1} \ .$$

We conclude that $\forall i, Z_n = Z_{n+1} \subseteq Y_i$, and so $Z_n \subseteq Y_\infty^M$.

Second part: $Y_\infty^M \subseteq Z_n$.
We prove by induction on i that for all i, $Y_\infty^M \subseteq Z_i$.

- By construction, $Y_\infty^M \subseteq Y_0 = Z_0$.
- Induction hypothesis: $Y_\infty^M \subseteq Z_i$.
- Before the projection we have (see [4]):

$$\tilde{Y}^\infty = Attr_0^+(Y_\infty^M) \subseteq Attr_0^+(Z_i) = Z'_{i+1} \ .$$

  To go to Z''_{i+1}, we proceed by induction on the number of transitions that are changed by the projection, *i.e.*, we consider the successive automata $\mathscr{A}_0, \cdots, \mathscr{A}_m$, where $L(\mathscr{A}_0) = Z'_{i+1}, L(\mathscr{A}_m) = Z''_{i+1}$, and $\mathscr{A}_{j+1}$ is obtained from $\mathscr{A}_j$ by "projecting" one transition. We have to prove by induction on m that $\tilde{Y}^\infty \subseteq L(\mathscr{A}_m)$.
  - If $m = 0$, $\tilde{Y}^\infty \subseteq L(\mathscr{A}_0) = Z'_{i+1} = Z''_{i+1}$.
  - Induction hypothesis: $\tilde{Y}^\infty \subseteq L(\mathscr{A}_m)$.
  - We suppose by absurd that there is a configuration $p\mu \in L(\mathscr{A}_m)\backslash L(\mathscr{A}_{m+1})$ such that $p\mu \in \tilde{Y}^\infty$. We choose that of minimal length $|p\mu|$. For each accepting path labelled by $p\mu$ in $\mathscr{A}_m$, there is a decomposition $p^{i+1} \xrightarrow[m]{\nu}{}_* S \xrightarrow[m]{\xi}{}_* F \subseteq \{f_0, \cdots, f_M\}$ such that $\mu = \nu\xi$, with $q^i \in S$, and in $\mathscr{A}_{m+1}$: $\neg\exists(q^{i+1} \xrightarrow[m+1]{\xi}{}_* F' \subseteq \{f_0, \cdots, f_M\})$ (the transition that is projected was "leading" to q^i in $\mathscr{A}_m$

and is now "leading" to q^{i+1}).
This means that $q\xi \in Z_i$ and $q\xi \notin L(\mathscr{A}_{m+1})$.
If $q\xi \notin L(\mathscr{A}_m)$, then $q\xi \notin \tilde{Y}^\infty$ (Ind. hyp.), and $p\mu$ should not stay in $\tilde{Y}^\infty$ (see [4]), hence the contradiction.
If $q\xi \in L(\mathscr{A}_m)\backslash L(\mathscr{A}_{m+1})$, then $q\xi$ cannot be in $\tilde{Y}^\infty$ by hypotheses: $\nu \neq \epsilon$ and $|q\xi| < |p\mu|$. If $q\xi$ is not in $\tilde{Y}^\infty$, then $p\mu$ should not be in $\tilde{Y}^\infty$ (see [4]), hence the contradiction.
We conclude that $\tilde{Y}^\infty \subseteq L(\mathscr{A}_{m+1})$.
We have now $Attr_0^+(Y_\infty^M) \subseteq Z''_{i+1}$, hence

$$Y_\infty^M \;=\; Attr_0^+(Y_\infty^M) \cap Q\Gamma^{\leqslant M} \;\subseteq\; Z''_{i+1} \cap Q\Gamma^{\leqslant M} \;.$$

It is now clear that $Z''_{i+1} \cap Q\Gamma^{\leqslant M} \;\subseteq\; Z_{i+1}$. (note that the inclusion from right to left was proved above.)

■