

THE DETERMINACY STRENGTH OF PUSHDOWN ω -LANGUAGES *

WENJUAN LI¹ AND KAZUYUKI TANAKA¹

Abstract. We investigate the determinacy strength of infinite games whose winning sets are recognized by nondeterministic pushdown automata with various acceptance conditions, *e.g.*, safety, reachability and co-Büchi conditions. In terms of the foundational program “Reverse Mathematics”, the determinacy strength of such games is measured by the complexity of a winning strategy required by the determinacy. Infinite games recognized by nondeterministic pushdown automata have some resemblance to those by deterministic 2-stack visibly pushdown automata with the same acceptance conditions. So, we first investigate the determinacy of games recognized by deterministic 2-stack visibly pushdown automata, together with that by nondeterministic ones. Then, for instance, we prove that the determinacy of games recognized by pushdown automata with a reachability condition is equivalent to the weak König lemma, stating that every infinite binary tree has an infinite path. While the determinacy for pushdown ω -languages with a Büchi condition is known to be independent from ZFC, we here show that for the co-Büchi condition, the determinacy is exactly captured by ATR_0 , another popular system of reverse mathematics asserting the existence of a transfinite hierarchy produced by iterating arithmetical comprehension along a given well-order. Finally, we conclude that all results for pushdown automata in this paper indeed hold for 1-counter automata.

Mathematics Subject Classification. 03D05, 03B30, 68Q45.

1. INTRODUCTION

In this paper, we are mainly concerned with the determinacy strength of infinite games (due to Gale and Stewart) whose winning sets are recognized by pushdown automata with various acceptance conditions. From a standpoint congenial to the foundational program “reverse mathematics” [32], the determinacy strength is measured by the complexity of a winning strategy of the game, which yields to one of the two players by the determinacy.

Büchi and Landweber [3] first studied the Gale–Stewart game $\mathbb{G}(A)$, where A is an ω -regular language accepted by a finite Büchi automaton or equivalently a deterministic Muller automaton. They showed that one can effectively decide the winner of such $\mathbb{G}(A)$ and a winning strategy can be constructed by a finite state transducer.

Walukiewicz [39, 40] showed that the games with winning sets accepted by deterministic Muller pushdown automata are determined with computable winning strategies that can be carried out by a pushdown transducer.

Keywords and phrases. Gale–Stewart games, pushdown ω -languages, 2-stack visibly pushdown automata, reverse mathematics.

* This work was supported in part by the JSPS KAKENHI Grant Number JP26540001 and JP15H03634.

¹ Mathematical Institute, Tohoku University, 6-3, Aramaki Aza-Aoba, Aoba-ku, Sendai 980-8578, Japan.
wenjuan.li1701@gmail.com; tanaka.math@tohoku.ac.jp

Subsequent to Thomas's suggestion for higher Borel games in [38], Cachat, Duparc and Thomas [5] defined a Σ_3^0 -complete acceptance condition and showed the infinite games whose winning sets are accepted by deterministic pushdown automata with such a condition are determined with computable winning strategies. Serre [30, 31] investigated the infinite games with arbitrary finite Borel level by introducing a finite chain of real-time (namely, the ε -transition is not allowed) deterministic pushdown automata with restriction on the stack, and showed such games are also determined with computable winning strategies. More extensions to infinite games recognized by other types of machines, *e.g.*, Büchi visibly pushdown automata (equivalent to deterministic Stair Büchi pushdown automata) and deterministic higher-order pushdown automata, can be found in [4, 6, 21].

On the other hand, for ω -languages accepted by nondeterministic pushdown automata, the situations are quite different. Context-free ω -languages, accepted by nondeterministic Büchi (or Muller) pushdown automata, are beyond finite Borel hierarchy [16]. Finkel proved that the determinacy of context-free ω -languages is equivalent to the determinacy of effective analytic games [14], which is not even provable in the set theory ZFC. In [12], he furthermore showed that there exists an infinite game with an effective Δ_3^0 winning set accepted by a real-time Büchi 1-counter automaton (a special kind of pushdown automaton) such that none of the players has a hyperarithmetical winning strategy. This indicates that for infinite games recognized by nondeterministic pushdown automata even at low levels of Borel hierarchy, the winning strategies might be highly undecidable. Then the following question emerges: if the winning strategies in such games are undecidable, exactly how undecidable are they?

In order to calibrate the complexity of winning strategies, we follow the terminologies from reverse mathematics, a framework to measure the provability of mathematical statements. Reverse mathematics makes use of several subsystems of second order arithmetic, of which the five particular subsystems are RCA_0 , WKL_0 , ACA_0 , ATR_0 , and $\text{II}_1^1\text{-CA}_0$, in order of increasing strength. Observe that even full second order arithmetic Z_2 is a much weaker system than ZFC. In particular, ZFC proves that every Borel game is determined, while Z_2 does not even prove determinacy for general Δ_4^0 games [25]. Note that weaker is *good* in this context, since subsystems of Z_2 can distinguish different kinds of Borel games below Δ_4^0 which are all characterized as determined by ZFC. In fact, studies on determinacy of infinite games are closely connected with the origin and backbone of reverse mathematics (*cf.* [19, 32, 34, 35]).

Although our main purpose is to analyze the complexity of winning strategies for infinite games whose winning sets are ω -languages accepted by pushdown automata with various acceptance conditions, we also deal with the infinite games recognized by 2-stack visibly pushdown automata. The 2-stack visibly pushdown automata is a kind of input-driven pushdown automata with two stacks. The input alphabet is partitioned into push, pop alphabet for each stack separately, and internal alphabet, which decide its visible actions on the stacks. We first investigate the determinacy strength of the classes of ω -languages recognized by deterministic 2-stack visibly pushdown automata, and in particular those with acceptance conditions of low complexity, such as safety, reachability and co-Büchi. Then, we show that most of the results also hold for nondeterministic 2-stack visibly pushdown automata with the same acceptance conditions.

Based on the techniques and results for deterministic 2-stack visibly pushdown automata, we investigate complexity of winning strategies for infinite games whose winning sets are ω -languages recognized by nondeterministic pushdown automata with the same various acceptance conditions and analogous results are obtained, except that the safety case is determined with computable winning strategies. In fact, all the results for pushdown automata in this study also hold for 1-counter automata.

We also remark that all the logical equivalences in this study with respect to reverse mathematics are finally established by considering the boldface classes of ω -languages, that is, ones defined by some kind of automata with an oracle tape as parameters, which are developed in order to keep in harmony with the technical requirements of reverse mathematics.

The remainder of this paper is organized as follows. In Section 2, we recall some basic notions of Gale–Stewart games, automata theory, and also give an introduction to reverse mathematics, as well as some previous studies on determinacy in reverse mathematics, which will be useful in the following. Sections 3 and 4 are dedicated to

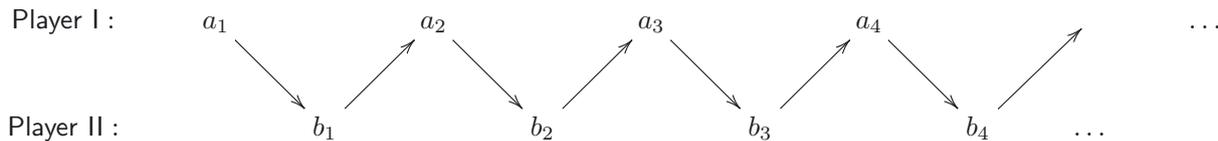


FIGURE 1. A Gale–Stewart game.

investigations on the complexity of winning strategies for infinite games recognized by 2-stack visibly pushdown automata and pushdown automata with various acceptance conditions. The conclusions are drawn in Section 5.

2. PRELIMINARIES

A *finite word*, over an alphabet X , is a finite sequence of letters $u = a_1a_2\dots a_n$, where $a_i \in X$ for all $0 < i \leq n$. For $0 < i \leq n$, $u(i)$ denotes the i th letter of u . For $i \leq n$, $u[i] = u(1)u(2)\dots u(i)$, which denotes the prefix of u of length i . ε is the *empty sequence*. $X^{<\omega}$ denotes the *set of finite words* over X . A *finite language* over X is a subset of $X^{<\omega}$. By $u.v$ or uv , we denote the concatenation of finite words u and v .

An *infinite word* or ω -*word* over X is an infinite sequence $\alpha = a_1a_2\dots a_n\dots$, where for all $i \geq 1$, $a_i \in X$. When α is an infinite word over X , we write $\alpha = \alpha(1)\alpha(2)\dots\alpha(n)\dots$ and $\alpha[n] = \alpha(1)\alpha(2)\dots\alpha(n)$, which is the prefix of α of length n . X^ω denotes the *set of infinite words* over X . An ω -*language* over X is a subset of X^ω . The concatenation of a finite word u and an infinite word v is also written as uv . For $V \subseteq X^{<\omega}$, $V^\omega = \{v_1\dots v_n\dots \in X^\omega : \forall i \geq 1, v_i \in V\}$ is called the ω -*power* of V .

Let X be a (countable) set. The topology on X^ω is defined by a metric $d(\alpha, \beta) = 2^{-\ell}$ where $\ell = \min\{i : \alpha(i) \neq \beta(i)\}$, where $\min \emptyset = +\infty$ and $2^{-\infty} = 0$. Then we say a subset of X^ω is open (*i.e.*, in Σ_1^0) if and only if it is in the form $YX^\omega = \{uv \in X^\omega : u \in Y, Y \subset X^{<\omega} \text{ and } v \in X^\omega\}$. A closed set (*i.e.*, in Π_1^0) is the complement of an open set. Δ_1^1 is the class of all Borel sets.

2.1. Gale–Stewart games

The Gale–Stewart game $\mathbb{G}(A)$ is a two-player game of perfect information, where $A \subseteq X^\omega$ is called a *winning set* (or winning condition) and X is an alphabet. Player I and player II select an element of X alternatively as shown in Figure 1.

Eventually, they produce an infinite sequence $x = a_1b_1a_2b_2\dots$ of X^ω , which is called a *play* in $\mathbb{G}(A)$. Player I wins with the play x if and only if $x \in A$. Otherwise, II wins.

A *strategy for player I* is a mapping $\sigma : (X^2)^{<\omega} \rightarrow X$. A *strategy for player II* is a mapping $\tau : (X^2)^{<\omega} \rightarrow X$. A strategy σ is a *winning strategy for player I* if any sequence that follows σ belongs to the winning set A , *i.e.*, if $a_1 = \sigma(\varepsilon)$ and for every $n > 1$ $a_n = \sigma(a_1b_1\dots a_{n-1}b_{n-1})$, we have $a_1b_1a_2b_2\dots \in A$. A strategy τ is a *winning strategy for player II* if it guarantees that $a_0b_0a_1b_1\dots \notin A$. A Gale–Stewart game is *determined* if one of the two players has a winning strategy.

We recall some classical determinacy results on Gale–Stewart games. Gale and Stewart in 1953 first showed the determinacy of Σ_1^0 games. In sequel, the determinacy of Π_2^0 , Σ_3^0 , Π_4^0 games are proved by Wolfe (1955), Davis (1964) and Paris (1972), respectively. Using level by level induction, Martin [22] showed that, in ZFC, all games with Borel winning conditions are determined, which is known as Borel determinacy. Note that (assuming Borel determinacy), any Borel game has a Δ_2^1 winning strategy, but Borel determinacy and Δ_2^1 comprehension (see Def. 1.11 in Sect. 2.3) are not comparable [24]. Beyond Borel hierarchy, the determinacy of Σ_1^1 (of analytic sets) games, as well as that of Σ_1^1 (of effective analytic sets) games, is not provable in ZFC, all of which need large cardinal assumptions [18, 23].

2.2. Pushdown automata

We introduce the basic notions on pushdown automata, 1-stack visibly pushdown automata and 2-stack visibly pushdown automata. We will also briefly recall some known results on infinite games whose winning sets are ω -languages accepted by these kinds of pushdown automata.

We start with the formal definition of pushdown automata as follows.

Definition 2.1. A (nondeterministic) *pushdown automaton* is a tuple $\mathcal{M} = (Q, X, \Gamma, q_{\text{in}}, \delta, F)$, where

- Q is a finite set of states,
- X is a finite input alphabet,
- Γ is a finite stack alphabet, which includes a special bottom letter \perp ,
- $q_{\text{in}} \in Q$ is the initial state,
- $\delta : Q \times (X \cup \{\varepsilon\}) \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma^{\leq 2})$ is the transition relation, and
- F is a set of final states.

The content of a stack is denoted by $\gamma \in (\Gamma \setminus \{\perp\})^{<\omega} \{\perp\}$. The leftmost letter will be assumed to be on the top of stack, also the bottom letter \perp can never be deleted and the rightmost letter is always \perp .

A pushdown automaton $\mathcal{M} = (Q, X, \Gamma, q_{\text{in}}, \delta, F)$ is said to be *deterministic* if for any $q \in Q$, $a \in X$ and $\gamma \in \Gamma$, $|\delta(q, a, \gamma)| + |\delta(q, \varepsilon, \gamma)| \leq 1$. By $|S|$, we denote the number of elements in a finite set S .

Definition 2.2. A *configuration* of a pushdown automaton \mathcal{M} is a pair (q, γ) , where $q \in Q$ and $\gamma \in (\Gamma \setminus \{\perp\})^{<\omega} \{\perp\}$. For $a \in X \cup \{\varepsilon\}$, $\gamma \in (\Gamma \setminus \{\perp\})^{<\omega} \{\perp\}$, $p, q \in Q$, $v \in \Gamma$ and $\beta \in \Gamma^{\leq 2}$, if $(q, \beta) \in \delta(p, a, v)$, then we denote $a : (p, v\gamma) \mapsto_{\mathcal{M}} (q, \beta\gamma)$. $\mapsto_{\mathcal{M}}^{<\omega}$ is the transitive and reflexive closure of $\mapsto_{\mathcal{M}}$. Notice that this transition is not a real-time one, namely, ε -transitions are not allowed.

Note that, in this study, we assume that for all $a \in X \cup \{\varepsilon\}$, $p \in Q$, $v \in \Gamma$, $|\delta(p, a, v)| > 0$ following the convention from [33].

Definition 2.3. Let $\alpha = a_1 a_2 \dots a_n \dots$ be an infinite word over X . An infinite sequence of configurations $r = (q_i, \gamma_i)_{i \geq 0}$ is called a *run* of \mathcal{M} on α , starting from the initial configuration (q_{in}, \perp) , if and only if

- (1) $(q_0, \gamma_0) = (q_{\text{in}}, \perp)$, and
- (2) for each $i \geq 1$, there exists $b_i \in X \cup \{\varepsilon\}$ such that $b_i : (q_{i-1}, \gamma_{i-1}) \mapsto_{\mathcal{M}} (q_i, \gamma_i)$ and such that $a_1 a_2 \dots a_n \dots = b_1 b_2 \dots b_n \dots$ or $b_1 b_2 \dots b_n \dots$ is a prefix of $a_1 a_2 \dots a_n \dots$.

For every run r , $\text{Inf}(r)$ is the set of states that are visited infinitely many times during the run r .

Remark that a “run” is defined in line with [33], which does not require the pushdown automata to read through the whole tape. Such a condition differs from the ones mentioned in [10, 11], which force the pushdown automata to eventually finish reading the whole tape. However, for Büchi and Muller acceptance conditions, the former and latter conditions define the same classes of ω -languages for pushdown automata [33].

Definition 2.4. A *Büchi pushdown automaton* is a tuple $\mathcal{M} = (Q, X, \Gamma, \delta, q_{\text{in}}, F)$. The ω -language accepted by \mathcal{M} is

$$L(\mathcal{M}) = \{\alpha \in X^\omega : \text{there is a run } r \text{ of } \mathcal{M} \text{ on } \alpha \text{ such that } \text{Inf}(r) \cap F \neq \emptyset\}.$$

Definition 2.5. A *Muller pushdown automaton* is a tuple $\mathcal{M} = (Q, X, \Gamma, \delta, q_{\text{in}}, \mathcal{F})$, where $\mathcal{F} \subseteq \mathcal{P}(Q)$ is a collection of accepting sets of states. The ω -language accepted by \mathcal{M} is

$$L(\mathcal{M}) = \{\alpha \in X^\omega : \text{there is a run } r \text{ of } \mathcal{M} \text{ on } \alpha \text{ such that } \text{Inf}(r) \in \mathcal{F}\}.$$

An ω -language is a *context-free ω -language* if it is accepted by a Büchi pushdown automaton (or Muller pushdown automaton). Then we denote

$$\mathbf{CFL}_\omega = \{L \subseteq X^\omega : \text{there exists a Büchi pushdown automaton } \mathcal{M} \text{ s.t. } L(\mathcal{M}) = L\}.$$

By \mathbf{DCFL}_ω , we denote the class of deterministic context-free ω -languages that are accepted by deterministic Muller pushdown automata. \mathbf{DCFL}_ω is a subclass of $\mathcal{B}(\Sigma_2^0)$. The class of ω -languages accepted by deterministic Büchi pushdown automata is strictly included in \mathbf{DCFL}_ω .

Finkel [17] proved that \mathbf{CFL}_ω exhausts the finite ranks of the Borel hierarchy, *i.e.*, for each $n \geq 1$, there exists a Σ_n^0 -complete context-free ω -language and a Π_n^0 -complete context-free ω -language. He [16] also showed that there exist some context-free ω -languages which are Borel sets of infinite rank. Moreover, Finkel [15] proved that the Wadge degrees, hence also Borel ranks, of \mathbf{CFL}_ω coincides with those of ω -languages of nondeterministic Turing machines. A survey dedicated to recent studies on context-free ω -languages can be found in [13].

Now we briefly recall the results for Gale–Stewart games in \mathbf{REG}_ω , \mathbf{DCFL}_ω , and \mathbf{CFL}_ω . For games in \mathbf{REG}_ω , Büchi and Landweber [3] showed that one of the two players has a winning strategy, which can be effectively constructed, which is known as the Büchi–Landweber theorem.

Theorem 2.6 [3]. *\mathbf{REG}_ω games are effectively determined with a computable winning strategy.*

This result is extended by Walukiewicz [39, 40] to deterministic Muller pushdown automata.

Theorem 2.7 [39, 40]. *\mathbf{DCFL}_ω games are effectively determined with a computable winning strategy.*

Finkel [17] proved that it is undecidable to determine the winner in the Gale–Stewart games recognized by Büchi pushdown automata. Subsequently, he showed that

Theorem 2.8 [14]. *The determinacy of \mathbf{CFL}_ω games is equivalent to the determinacy of Σ_1^1 games, where Σ_1^1 is the class of effective analytic sets.*

Recently, Finkel [12] extended such an equivalence to the class of ω -languages accepted by 2-tape Büchi automata. Note that determinacy of Σ_1^1 games is not provable in ZFC and requires a large cardinal assumption.

Finkel [12] also proved that there exists an infinite game with an effective Δ_3^0 winning set which is accepted by a real-time Büchi 1-counter automaton (a special kind of pushdown automaton) such that none of the players has a hyperarithmetical winning strategy.

To downscale the winning sets to lower complexity, apart from the Büchi and Muller conditions, we also consider the ω -languages defined by the following acceptance conditions. For simplicity we here write a run r as the sequence of states appearing in r .

- Safety (or Π_1) acceptance condition.

$$L(\mathcal{M}) = \{\alpha \in X^\omega : \text{there is a run } r = (q_i)_{i \geq 0} \text{ of } \mathcal{M} \text{ on } \alpha \text{ such that } \forall i, q_i \in F\}.$$

- Reachability (or Σ_1) acceptance condition.

$$L(\mathcal{M}) = \{\alpha \in X^\omega : \text{there is a run } r = (q_i)_{i \geq 0} \text{ of } \mathcal{M} \text{ on } \alpha \text{ such that } \exists i, q_i \in F\}.$$

- Co-Büchi (or Σ_2) acceptance condition.

$$L(\mathcal{M}) = \{\alpha \in X^\omega : \text{there is a run } r = (q_i)_{i \geq 0} \text{ of } \mathcal{M} \text{ on } \alpha \text{ such that } \text{Inf}(r) \subseteq F\}.$$

We also treat the following ω -languages with the combinations of the above conditions.

- $(\Sigma_1 \wedge \Pi_1)$ acceptance condition. There exist $F_r, F_s \subseteq Q$,

$$L(\mathcal{M}) = \{\alpha \in X^\omega : \text{there is a run } r = (q_i)_{i \geq 0} \text{ of } \mathcal{M} \text{ on } \alpha \text{ such that } \exists i, q_i \in F_r \wedge \forall i, q_i \in F_s\}.$$

- $(\Sigma_1 \vee \Pi_1)$ acceptance condition. There exist $F_r, F_s \subseteq Q$,

$$L(\mathcal{M}) = \{\alpha \in X^\omega : \text{there is a run } r = (q_i)_{i \geq 0} \text{ of } \mathcal{M} \text{ on } \alpha \text{ such that } \exists i, q_i \in F_r \vee \forall i, q_i \in F_s\}.$$

- Δ_2 acceptance condition. There exist $F_b, F_c \subseteq Q$,

$$\begin{aligned} L(\mathcal{M}) &= \{\alpha \in X^\omega : \text{there is a run } r \text{ of } \mathcal{M} \text{ on } \alpha \text{ such that } \text{Inf}(r) \cap F_b \neq \emptyset\} \\ &= \{\alpha \in X^\omega : \text{there is a run } r \text{ of } \mathcal{M} \text{ on } \alpha \text{ such that } \text{Inf}(r) \subseteq F_c\}. \end{aligned}$$

Visibly pushdown automata

Visibly pushdown automata, initially introduced by Alur and Madhusudan in [1], is a special kind of pushdown automata with restriction on the input alphabet. The alphabet is partitioned into **Push**, **Pop**, **Int**. The *transitions* are input-driven in the sense that while it reads a letter a on the input tape, the operations on the stack depend on the affiliation of a . That is, if $a \in \mathbf{Push}$, the visibly pushdown automaton pushes one letter to the stack; if $a \in \mathbf{Pop}$, it pops the top one letter from the stack; if $a \in \mathbf{Int}$, it does not touch the stack. See [1, 21] for more introduction on visibly pushdown automata.

Similarly, we can define the classes of ω -languages accepted by Büchi 1-stack visibly pushdown automata and deterministic Muller 1-stack visibly pushdown automata, and denote them as \mathbf{VPL}_ω and \mathbf{DVPL}_ω .

Lödning, Madhusudan, Serre [21] showed that \mathbf{VPL}_ω is in $\mathcal{B}(\Sigma_3^0)$. They also proved that the visibly pushdown games with visibly pushdown winning conditions are determined, where visibly pushdown games are played on graphs generated by visibly pushdown processes. Since \mathbf{DVPL}_ω is a subclass of \mathbf{DCFL}_ω , \mathbf{DVPL}_ω games are also effectively determined with a computable winning strategy.

For the 2-stack visibly pushdown automata, the alphabet is partitioned into \mathbf{Push}_1 , \mathbf{Pop}_1 , \mathbf{Push}_2 , \mathbf{Pop}_2 and **Int**, where each subscript indicates the stack that associated with this subalphabet [8]. For finite words, it can express properties beyond the context-free languages [7, 8, 36, 37]. On infinite words, the expressive power of Büchi 2-stack visibly pushdown automata is equivalent to the existential monadic second-order logic [2].

The formal definition of 2-stack visibly pushdown automata is given as follows.

Definition 2.9. A 2-stack visibly pushdown automaton is a tuple $\mathcal{M} = (Q, X, \Gamma, q_{\text{in}}, \delta, F)$, where

- Q is a finite set of states,
- $X = \mathbf{Push}_1 \cup \mathbf{Pop}_1 \cup \mathbf{Push}_2 \cup \mathbf{Pop}_2 \cup \mathbf{Int}$ is a finite input alphabet,
- Γ is a finite stack alphabet, which contains a special bottom letter \perp ,
- $q_{\text{in}} \in Q$ is the initial state,
- $\delta = \delta_{\text{Push}_1} \cup \delta_{\text{Pop}_1} \cup \delta_{\text{Push}_2} \cup \delta_{\text{Pop}_2} \cup \delta_{\text{Int}}$ is a transition relation, where:
 - ★ $\delta_{\text{Push}_1} \subseteq Q \times \mathbf{Push}_1 \times Q \times (\Gamma \setminus \{\perp\})$,
 - ★ $\delta_{\text{Pop}_1} \subseteq Q \times \mathbf{Pop}_1 \times \Gamma \times Q$,
 - ★ $\delta_{\text{Push}_2} \subseteq Q \times \mathbf{Push}_2 \times Q \times (\Gamma \setminus \{\perp\})$,
 - ★ $\delta_{\text{Pop}_2} \subseteq Q \times \mathbf{Pop}_2 \times \Gamma \times Q$,
 - ★ $\delta_{\text{Int}} \subseteq Q \times \mathbf{Int} \times Q$,
- $F \subseteq Q$ is a set of final states.

A configuration of a 2-stack visibly pushdown automaton is in the form (q, γ^1, γ^2) , where $q \in Q$ and $\gamma^1, \gamma^2 \in (\Gamma \setminus \{\perp\})^{<\omega}$ represent the contents of the two stacks.

Definition 2.10. Let $\alpha = a_1 a_2 \dots a_n \dots$ be an infinite word over X . An infinite sequence of configurations $r = (q_i, \gamma_i^1, \gamma_i^2)_{i \geq 0}$ is called a *run* of a 2-stack visibly pushdown automaton on α , starting from the initial configuration $(q_{\text{in}}, \perp, \perp)$, if and only if:

- (1) $(q_0, \gamma_0^1, \gamma_0^2) = (q_{\text{in}}, \perp, \perp)$, and
- (2) for each $i > 1$,
 - $(q_{i-1}, a_i, q_i, v) \in \delta_{\text{Push}_1}$, $\gamma_i^1 = v\gamma_{i-1}^1$, and $\gamma_i^2 = \gamma_{i-1}^2$, or
 - $(q_{i-1}, a_i, v, q_i) \in \delta_{\text{Pop}_1}$ and either $(v \in \Gamma \setminus \{\perp\}, \gamma_{i-1}^1 = v\gamma_i^1, \gamma_i^2 = \gamma_{i-1}^2)$ or $(v = \perp = \gamma_{i-1}^1 = \gamma_i^1, \gamma_i^2 = \gamma_{i-1}^2)$, or
 - $(q_{i-1}, a_i, q_i, v) \in \delta_{\text{Push}_2}$, $\gamma_i^1 = \gamma_{i-1}^1$ and $\gamma_i^2 = v\gamma_{i-1}^2$, or
 - $(q_{i-1}, a_i, v, q_i) \in \delta_{\text{Pop}_2}$ and either $(v \in \Gamma \setminus \{\perp\}, \gamma_{i-1}^1 = \gamma_i^1, \gamma_{i-1}^2 = v\gamma_i^2)$ or $(\gamma_{i-1}^1 = \gamma_i^1, v = \perp = \gamma_{i-1}^2 = \gamma_i^2)$, or
 - $(q_{i-1}, a_i, q_i) \in \delta_{\text{Int}}$, $\gamma_i^1 = \gamma_{i-1}^1$, and $\gamma_i^2 = \gamma_{i-1}^2$.

The nondeterministic 2-stack visibly pushdown automata (on finite words) is not closed under determination, which is shown by Torre, Madhusudan, and Parlato *via* the following example. Given $X = \{a\} \cup \{c, d\} \cup \{b\} \cup \{x, y\}$, where $\mathbf{Int} = \emptyset$, the language $\{(ab)^n c^i d^{n-i} x^i y^{n-i} : n, i \in \omega \text{ and } i \leq n\}$ is recognized by a nondeterministic 2-stack visibly pushdown automaton, but not any deterministic one. Moreover, the languages accepted by (2-stack) visibly pushdown automata and pushdown automata are not comparable, since $\{a^n b a^n : n \in \omega\}$ can be easily accepted by a deterministic pushdown automata but not any 1-stack or 2-stack visibly pushdown automaton. The above properties easily hold for (2-stack) visibly pushdown automata on infinite words.

By $\mathbf{2VPL}_\omega$ and $\mathbf{2DVPL}_\omega$, we denote the classes of ω -languages accepted by Büchi 2-stack visibly pushdown automata and deterministic Muller 2-stack visibly pushdown automata. The expressive power of pushdown automata with two stacks is very strong, even for the restricted visibly case. The Gale–Stewart games in $\mathbf{2VPL}_\omega$ and $\mathbf{2DVPL}_\omega$ are undecidable.

Figure 2 shows the inclusion relations on some classes of ω -languages, as well as the decidability of infinite games in these classes, where \mathbf{BTM}_ω and $\mathbf{2BCL}_\omega$ denote the classes of ω -languages that are recognized by Büchi Turing machines and Büchi 2-counter automata, respectively. By “decidable”, we specify the games that are effectively determined (namely, which player has a winning strategy is effectively determined) with computable winning strategies.

All the classes of ω -languages in Figure 2 are accepted by some kind of machines with a Büchi and/or a Muller condition. In the following, we consider several other acceptance conditions of lower complexity, and particularly the infinite games whose winning sets are recognized by deterministic (respectively, nondeterministic) 2-stack

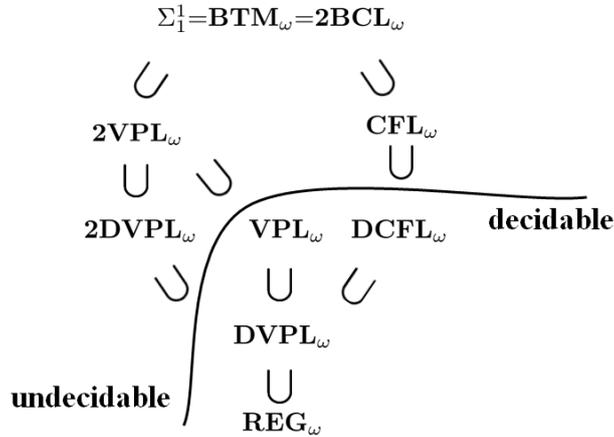


FIGURE 2. The classes of ω -languages and the decidability of their games.

visibly pushdown automata and nondeterministic pushdown automata with such conditions, which turns out to provide us more classes that are closer to the boundary in Figure 2.

2.3. Second order arithmetic

Second order arithmetic Z_2 is formally developed in the two-sorted first-order language with number variables x, y, z, \dots , and unary function variables f, g, h, \dots , constant symbols $0, 1$, binary function symbol $+$, \cdot , and binary relation symbols $=, <$. Set variables X, Y, Z, \dots , are also used to range over the $\{0, 1\}$ -functions. The classifications of formulas are given as follows, where $i \in \{1, 2\}$, $n \in \omega$.

- (1) φ is Π_0^0 (bounded) if φ is constructed by atomic formulas with propositional connectives and bounded number quantifiers $\forall x < t$ and $\exists x < t$, where t does not contain x .
- (2) φ is Π_0^1 (arithmetical) if it contains no function quantifiers.
- (3) $\neg\varphi$ is Σ_1^i if φ is Π_1^i .
- (4) $\forall x_1 \dots \forall x_k \varphi$ is Π_{n+1}^0 if $\varphi \in \Sigma_n^0$.
- (5) $\forall f_1 \dots \forall f_k \varphi$ is Π_{n+1}^1 if $\varphi \in \Sigma_n^1$.

Based on such classification of formulas, we consider the following schemes of the set existence axioms.

Definition 2.11. Let $n \in \omega$ and $i \in \{1, 2\}$.

- (1) Δ_n^i -CA, which stands for Δ_n^i comprehension, consists of all the axioms of the form

$$\forall x(\varphi(x) \leftrightarrow \psi(x)) \rightarrow \exists X \forall x(x \in X \leftrightarrow \varphi(x)),$$

where $\varphi(x)$ is Σ_n^i , $\psi(x)$ is Π_n^i and X does not occur freely in φ .

- (2) Π_n^i -CA (respectively, Σ_n^i -CA), which stands for Π_n^i comprehension (respectively, Σ_n^i comprehension), consists of all the axioms of the form

$$\exists X \forall x(x \in X \leftrightarrow \varphi(x)),$$

where $\varphi(x)$ is Π_n^i (respectively, Σ_n^i) and X does not occur freely in φ .

- (3) Σ_n^i -SP, which stands for Σ_n^i separation, consists of all the axioms of the form

$$\neg \exists x(\varphi_0(x) \wedge \varphi_1(x)) \rightarrow \exists X \forall x((\varphi_0(x) \rightarrow x \in X) \wedge (\varphi_1(x) \rightarrow x \notin X)),$$

where $\varphi_0(x)$ and $\varphi_1(x)$ are Σ_n^i .

It is worth remarking that the above schemes should be in boldface versions, in the sense that they are defined with the class of formulas with real parameters. However the boldface notion is customarily used only for sets of reals. The above axioms assert the existence of sets of natural numbers, and their lightface counterparts are often denoted as Δ_n^i -CA⁻, etc.

A standard base theory for developing reverse mathematics is the system RCA_0 , which stands for recursive comprehension axiom. RCA_0 consists of algebraic axioms for $(\omega, +, \cdot, 0, 1, <)$ plus induction axioms for Σ_1^0 formulas and Δ_1^0 -CA. In general, adding axiom T to RCA_0 , we obtain a theory T_0 . The following are well-known in the study of reverse mathematics.

- (1) Σ_1^0 -SP₀ is equivalent (over RCA_0) to weak König's lemma, which states that every infinite binary tree has an infinite path. So, we also write WKL_0 for Σ_1^0 -SP₀.
- (2) Σ_1^0 -CA₀ is equivalent (over RCA_0) to arithmetical comprehension, asserting the existence of Σ_n^0 sets for all $n \in \omega$. So we also write ACA_0 for Π_0^1 -CA₀.
- (3) Σ_1^1 -SP₀ is equivalent (over RCA_0) to arithmetical transfinite recursion, which asserts the existence of a transfinite hierarchy produced by iterating arithmetical comprehension along a given well-order. So we write ATR_0 for Σ_1^1 -SP₀.

- (4) To measure the determinacy strength of Δ_3^0 games in the Baire space, the second author of this paper also defined $\mathcal{C}\text{-TR}_0$ (and $\mathcal{C}\text{-ID}_0$) as the system RCA_0 plus transfinite recursion for formulas on \mathcal{C} (and in the definition for operations in \mathcal{C}) (cf. [32, 35]). Since we do not treat a statement stronger than ATR_0 , we have omitted their precise definitions.
- (5) The so-called *Big Five* systems are

$$\begin{aligned} \text{RCA}_0 &\equiv \Delta_1^0\text{-CA}_0 \equiv \Pi_1^0\text{-SP}_0 \\ \text{WKL}_0 &\equiv \Sigma_1^0\text{-SP}_0 \\ \text{ACA}_0 &\equiv \Sigma_1^0\text{-CA}_0 \equiv \Pi_0^1\text{-CA}_0 \equiv \Pi_0^1\text{-SP}_0 \\ \text{ATR}_0 &\equiv \Sigma_1^1\text{-SP}_0 \\ \Pi_1^1\text{-CA}_0 &\equiv \Sigma_1^1\text{-CA}_0 \end{aligned}$$

Here we are concerned with the logical strength of determinacy of infinite games. *Determinacy of class \mathcal{C}* , denoted by $\mathcal{C}\text{-Det}$, asserts that all infinite games with winning sets in \mathcal{C} are determined. When we say the determinacy of a class \mathcal{C} is equivalent to (the set existence axioms of) a subsystem \mathcal{S} of \mathbf{Z}_2 , we mean that not only determinacy of \mathcal{C} is provable in \mathcal{S} but also $\text{RCA}_0 + \mathcal{C}\text{-Det}$ proves all the axioms of \mathcal{S} . Thus, by reverse mathematical investigations, we can characterize the complexity of winning strategies.

Infinite games and strategies in the Baire space (ω^ω) and the Cantor space (2^ω) are easily formalized in RCA_0 . However, even $\Delta_1^0\text{-Det}$ (in ω^ω) is not provable in RCA_0 . This is because a winning strategy of a Δ_1^0 game is not recursive but hyperarithmetical. In fact, boldface $\Delta_1^0\text{-Det}$ turns out to be equivalent to ATR_0 . Table 1 summarizes the previous results on the determinacy in the Baire space and the Cantor space, e.g., [28, 32], where $\text{Sep}(\mathcal{C}, \mathcal{D}) \equiv \{(C \cap D_1^c) \cup (C^c \cap D_2) : C \in \mathcal{C}, D_1, D_2 \in \mathcal{D}\}$, and the items in the same line are pairwise equivalent over RCA_0 . For example, the third line is read as

$$\text{ATR}_0 \leftrightarrow \Delta_1^0\text{-Det}(\text{in } \omega^\omega) \leftrightarrow \Sigma_1^0\text{-Det}(\text{in } \omega^\omega) \leftrightarrow \Delta_2^0\text{-Det}(\text{in } 2^\omega) \leftrightarrow \Sigma_2^0\text{-Det}(\text{in } 2^\omega).$$

It is worth noting that the lightface versions of determinacy are not so easily classified. For instance, Nemoto, MedSalem and Tanaka [28] proved the equivalence of boldface $\Pi_1^1\text{-CA}_0$ and $\Sigma_1^0 \wedge \Pi_1^0\text{-Det}$, and also the non-equivalence of lightface counterparts over RCA_0 . Moreover, we also remark that the clopen determinacy and open determinacy over the reals are known to be different in higher-order reverse mathematics [29].

TABLE 1. Reverse mathematics and infinite games in ω^ω and 2^ω .

	Determinacy in ω^ω	Determinacy in 2^ω
WKL_0		Δ_1^0, Σ_1^0
ACA_0		$\mathcal{B}(\Sigma_1^0)$
ATR_0	Δ_1^0, Σ_1^0	Δ_2^0, Σ_2^0
$\Pi_1^1\text{-CA}_0$	$\mathcal{B}(\Sigma_1^0)$	$\text{Sep}(\Sigma_1^0, \Sigma_2^0)$
$\Pi_1^1\text{-TR}_0$	Δ_2^0	$\text{Sep}(\Delta_2^0, \Sigma_2^0)$
$\Sigma_1^1\text{-ID}_0$	Σ_2^0	$\Sigma_2^0 \wedge \Pi_2^0$
$\Sigma_\infty^0(\Sigma_1^1)\text{-ID}_0 (\approx \Pi_2^1\text{-CA}_0)$	$\mathcal{B}(\Sigma_2^0)$	$\mathcal{B}(\Sigma_2^0)$

3. DETERMINACY STRENGTH OF INFINITE GAMES RECOGNIZED BY 2-STACK VISIBLY PUSHDOWN AUTOMATA

In this section, we investigate the determinacy strength of infinite games whose winning sets are accepted by 2-stack visibly pushdown automata. We start with analyzing the complexity of winning strategies in the infinite games recognized by deterministic 2-stack visibly pushdown automata. Then we show that most of the corresponding results also hold for nondeterministic ones.

The classes of ω -languages accepted by deterministic 2-stack visibly pushdown automata (2DVPA) with safety, reachability, co-Büchi and Büchi conditions are denoted as shown in Table 2.

TABLE 2. The classes of ω -languages accepted by 2DVPA.

Acceptance conditions	Subclass of $\mathbf{2DVPL}_\omega$
Reachability	$\mathbf{2DVPL}_\omega(\Sigma_1)$
Safety	$\mathbf{2DVPL}_\omega(\Pi_1)$
Co-Büchi	$\mathbf{2DVPL}_\omega(\Sigma_2)$
Büchi	$\mathbf{2DVPL}_\omega(\Pi_2)$

To characterize the complexity of the above classes of ω -languages, we would like first recall some results on deterministic and nondeterministic Turing machines. We follow the definition of Turing machines from [33], in which the machines are not required to finish reading the whole tape. By $\mathbf{TM}(C)$ (respectively, $\mathbf{DTM}(C)$), we denote the class of ω -languages recognized by nondeterministic (respectively, deterministic) Turing machines with acceptance condition C .

Theorem 3.1 (*cf.* [33]).

$$\mathbf{DTM}_\omega(\Pi_1) = \mathbf{TM}_\omega(\Pi_1) = \Pi_1^0$$

$$\mathbf{DTM}_\omega(\Sigma_1) = \mathbf{TM}_\omega(\Sigma_1) = \Sigma_1^0$$

$$\mathbf{DTM}_\omega(\Sigma_2) = \mathbf{TM}_\omega(\Sigma_2) = \Sigma_2^0$$

$$\mathbf{DTM}_\omega(\Pi_2) = \Pi_2^0$$

$$\mathbf{TM}_\omega(\Pi_2) = \Sigma_1^1$$

Note that the equalities of Theorem 3.1 also hold for the boldface versions. We here remark that boldface/lightface $\mathbf{2(D)VPL}_\omega$ and \mathbf{PDL}_ω (namely, the class of ω -languages recognized by pushdown automata that we will treat in the next section) are included in the corresponding boldface/lightface $(\mathbf{D})\mathbf{TM}_\omega$, and hence also by the corresponding formulas with/without parameters. In particular, the lightface $\mathbf{2DVPL}_\omega(\Pi_1)$ (respectively, $\mathbf{2DVPL}_\omega(\Sigma_1)$, $\mathbf{2DVPL}_\omega(\Sigma_2)$, $\mathbf{2DVPL}_\omega(\Pi_2)$) is a subclass of effective Π_1^0 (respectively, Σ_1^0 , Σ_2^0 , Π_2^0) class.

We now begin with considering the infinite games whose winning sets for player I are recognized by deterministic 2-stack visibly pushdown automata with a $(\Sigma_1 \wedge \Pi_1)$ acceptance condition. We prove that:

Theorem 3.2. *There exists an infinite game in $\mathbf{2DVPL}_\omega(\Sigma_1 \wedge \Pi_1)$ with only Σ_1^0 -hard winning strategies.*

Our goal is to show that there exists a deterministic 2-stack visibly pushdown automaton \mathcal{M} with a $(\Sigma_1 \wedge \Pi_1)$ acceptance condition such that in the game $\mathbb{G}(L(\mathcal{M}))$, player II has a winning strategy and all winning strategies are Σ_1^0 -hard.

To prove this theorem, we first recall the notion of (universal nondeterministic) 2-counter automata. A 2-counter automaton can be seen as a restricted 2-stack pushdown automaton with just one symbol for each stack, in the sense that the number of the symbols in a stack is expressed as a nonnegative integer in a counter. The input of a 2-counter automaton is a natural number m which is initially stored in one of the counters. Then by the current state and the tests results on whether each counter is zero or not, the automaton goes to the next state and do operations on the two counters by increasing the counter(s) by 1, or decreasing the counter(s) by 1 if the counter is not zero.

A 2-counter automaton is a tuple $\mathcal{R} = (Q, \delta, q_{\text{in}}, F)$, where Q is a finite set of states, $\delta \subseteq Q \times \{0, 1\}^2 \times Q \times \{-1, 0, 1\}^2$ a transition relation, q_{in} an initial state and F a set of halting states. A configuration of 2-counter automata is (q, m, n) , where $q \in Q$, and m, n are nonnegative integers in the two counters. For any input natural number m of \mathcal{R} , the initial configuration is $(q_{\text{in}}, m, 0)$.

We code a configuration (q, m, n) of a 2-counter automaton as qa^mb^n . A run for a natural number m on a 2-counter automaton \mathcal{R} is a sequence of configurations such that $q_{\text{in}}a^{m_0}b^{n_0} \mapsto_{\mathcal{R}} q_1a^{m_1}b^{n_1} \mapsto_{\mathcal{R}} q_2a^{m_2}b^{n_2} \mapsto_{\mathcal{R}} \dots$, where q_{in} is the initial state, $m_0 = m$, $n_0 = 0$ and $\mapsto_{\mathcal{R}}$ is defined by the transition relation δ of \mathcal{R} . A run is halting if it reaches a halting configuration.

We define a number $m \in L(\mathcal{R})$ if and only if there exists a run on m such that $q_{\text{in}}a^mb^{n_0} \mapsto_{\mathcal{R}} q_1a^{m_1}b^{n_1} \mapsto_{\mathcal{R}} \dots \mapsto_{\mathcal{R}} q_sa^{m_s}b^{n_s}$, where $n_0 = 0$ and $q_s \in F$. It is known that a 2-counter automaton, even a deterministic one, is equivalent to a Turing machine [26, 27]. Thus the halting problem for a certain (universal deterministic) 2-counter automaton is Σ_1^0 -complete. In the following, by 2-counter automata, we mean deterministic 2-counter automata unless stated otherwise.

Proof of Theorem 3.2. Let \mathcal{R} be a universal 2-counter automaton. We will construct a game associated with \mathcal{R} , denoted by $\mathbb{G}_{\mathcal{R}}$, such that the halting problem of \mathcal{R} is computable in any winning strategy of player II, while player I has no winning strategy, and moreover the winning set for player II is recognized by a deterministic 2-stack visibly pushdown automaton with a $(\Sigma_1 \vee \Pi_1)$ acceptance condition. Note that in this case, the winning set for player I is recognized by a deterministic 2-stack visibly pushdown automaton \mathcal{M} with a $(\Sigma_1 \wedge \Pi_1)$ acceptance condition.

We consider the following two-stage infinite game $\mathbb{G}_{\mathcal{R}}$.

In the first stage, player I produces a code of a number m in the form $\overbrace{111 \dots 1}^m 0$ in the first $m + 1$ rounds. During these rounds, player II can choose anything (denoted as \star), but makes no sense to the game. By choosing 0, player I announces an end of the code for m and asks whether m is accepted by \mathcal{R} or not. In this stage, if player I breaks the above rule of the game, *e.g.*, player I just produces infinite many 1's, which makes the game sink in the first stage and never enter the second stage, then player I loses.

If the game enters the second stage, it is player II's turn to answer yes (denoted as 1) or no (denoted as 0), leading the game to the following possible cases.

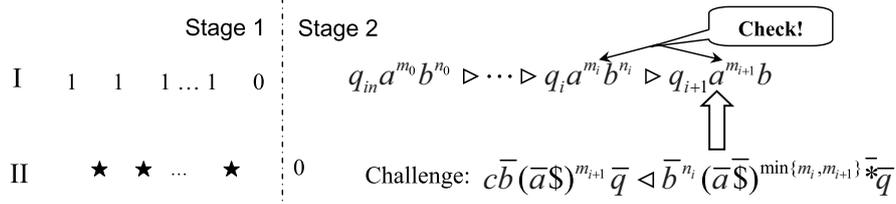
- If player II chooses yes,
 - player II should provide a sequence of configurations on m of \mathcal{R} to support her argument, and
 - while player II is making the sequence of configurations, player I may choose to challenge at the point he believes player II has cheated.

Player II wins if player I never challenge, or fails the challenge (namely, no error is found), otherwise I wins.

- If player II chooses no, then
 - player I defends by providing a sequence of configurations on m of \mathcal{R} that he claims correct, and
 - while player I is making the sequence of configurations, player II may challenge at the point she believes player I has cheated.

Player II wins if she manages to find an error by the challenge, otherwise I wins.

For simplicity, here we just explain the case where player II challenges when she answers no. Player II needs to check whether or not the sequence of configurations provided by player I is a halting run of \mathcal{R} with m . That is, player II makes sure that player I has obeyed the following rules.

FIGURE 3. A game $\mathbb{G}_{\mathcal{R}}$ with a challenge by player II.

- (1) the sequence of configurations provided by player I is a sequence in the form of $qa^m b^n$ and connected by \triangleright (for simplicity \triangleright is the code of $\mapsto_{\mathcal{R}}$),
- (2) it starts with the initial configuration,
- (3) any two consecutive configurations constitute a valid transition of \mathcal{R} , and
- (4) the sequence of configurations is ended with a halting configuration.

The rules (1), (2), and (4) are easy to check with Σ_1 conditions (namely, player I loses with Π_1). In the following, we treat the case that “player II answers no and player II challenges to confirm rule (3)” as a subgame of $\mathbb{G}_{\mathcal{R}}$, denoted as $\mathbb{G}_{\mathcal{R}}^*$.

Now in $\mathbb{G}_{\mathcal{R}}^*$, player II challenges by providing a modified reverse form of the last two configurations including a witness for error. It is in the form of $c\bar{b}(\bar{a}\$)^{m_{i+1}}\bar{q} \triangleleft \bar{b}^{n_i}(\bar{a}\$)^{\min\{m_i, m_{i+1}\}}\bar{*}\bar{q}$ as shown in Figure 3, where c indicates a start of challenge, $\$$ and $\bar{\$}$ are for comparing m_{i+1} with m_i , and $\bar{*}$ is a witness for error as we will explain below.

The winning set for player II in $\mathbb{G}_{\mathcal{R}}^*$ can be accepted by a deterministic 2-stack visibly pushdown automaton with the input alphabet partitioned into

- $\mathbf{Push}_1 = \{1, 0, a, b, \triangleright, q_{in}, q_1, \dots, q_j\}$, $\mathbf{Pop}_1 = \{\bar{a}, \bar{b}, \triangleleft, \bar{q}\}$,
- $\mathbf{Push}_2 = \{\$, \bar{\$}\}$, $\mathbf{Pop}_2 = \{\bar{\$}\}$, and $\mathbf{Int} = \{c, \star\}$.

We consider such a play in $\mathbb{G}_{\mathcal{R}}^*$ as an input of a deterministic 2-stack visibly pushdown automaton, then the operations on the two stacks while reading the play from left to right are done according to the partition of the alphabet as shown in Figure 4.

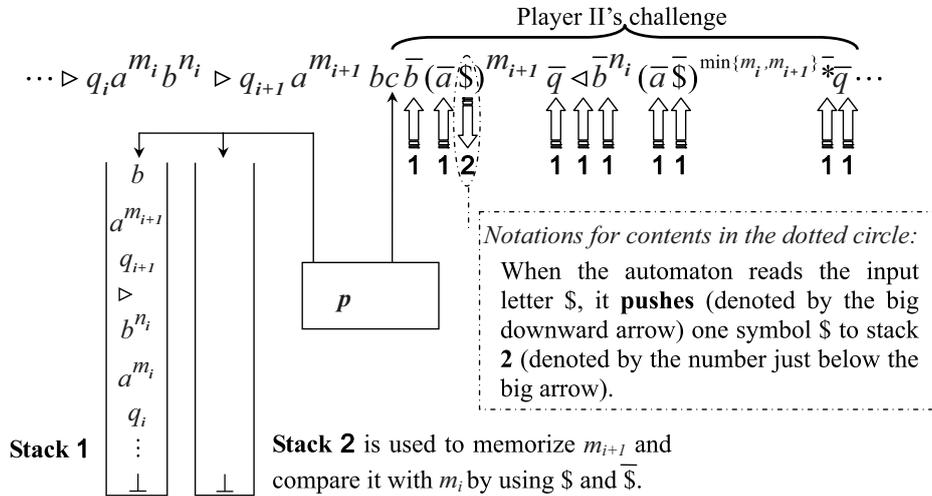


FIGURE 4. Regarding a play as an input of a deterministic 2-stack visibly pushdown automaton.

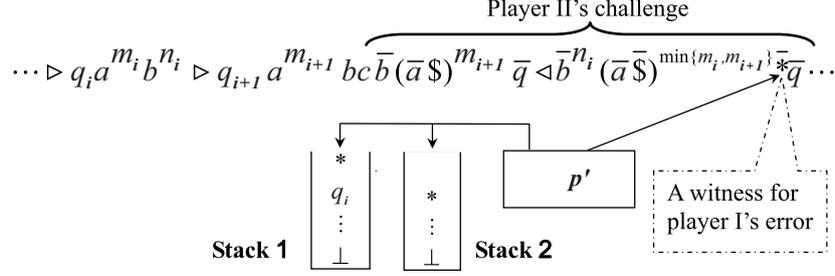


FIGURE 5. Regarding a play as an input of a deterministic 2-stack visibly pushdown automaton.

We look at more details on how the witness for error works in this case. After reading $(\bar{a}\$)^{\min\{m_i, m_{i+1}\}}$, it pops $\min\{m_i, m_{i+1}\}$ many a 's from the top of stack 1 and also $\min\{m_i, m_{i+1}\}$ many $\$$ from the top of stack 2. Then, the current condition of the top of two stacks is in one of the following forms,

- $[a, \perp]$ if $m_{i+1} < m_i$,
- $[q_i, \$]$ if $m_{i+1} > m_i$, or
- $[q_i, \perp]$ if $m_{i+1} = m_i$.

Finally, it meets the letter \bar{q} , a witness for error as in Figure 5. There are three conditions for the top of the two stacks as described above: $[*, \perp]$, $[q_i, *]$ or $[q_i, \perp]$, where $*$ can be either a (on the top of stack 1) or $\$$ (on the top of stack 2) that is determined by the successive transition from q_i and will be popped by reading \bar{q} .

If all the following are true, player II succeeds her challenge and wins this game.

- $(q_i, a^{m_i}, b^{n_i}) \mapsto_{\mathcal{R}} (q_{i+1}, a^{m_i+1}, b^{n_i}) \rightarrow \bar{q} \neq \bar{\$}$, and
- $(q_i, a^{m_i}, b^{n_i}) \mapsto_{\mathcal{R}} (q_{i+1}, a^{m_i-1}, b^{n_i}) \rightarrow \bar{q} \neq \bar{a}$, and
- neither $(q_i, a^{m_i}, b^{n_i}) \mapsto_{\mathcal{R}} (q_{i+1}, a^{m_i+1}, b^{n_i})$ nor $(q_i, a^{m_i}, b^{n_i}) \mapsto_{\mathcal{R}} (q_{i+1}, a^{m_i-1}, b^{n_i}) \rightarrow \bar{q} \neq \emptyset$.

It should be noted that the last letters \bar{q} and \bar{q} in player II's challenge can be used to detect other possible errors in the sequence provided by player I. For instance, if $m_{i+1} = m_i + 2$ and $(q_i, a^{m_i}, b^{n_i}) \mapsto_{\mathcal{R}} (q_{i+1}, a^{m_i+1}, b^{n_i})$, then player II can not only confirm $m_{i+1} > m_i$ since $\bar{q} = \bar{\$}$, but also detect the error that $m_{i+1} \neq m_i + 1$.

The winning plays of player II in this subgame $\mathbb{G}_{\mathcal{R}}^*$ can be recognized by a deterministic 2-stack visibly pushdown automaton with a Σ_1 acceptance condition.

If we consider the game $\mathbb{G}_{\mathcal{R}}$, player II's winning set can be recognized by an enlarged deterministic 2-stack visibly pushdown automaton with a $(\Sigma_1 \vee \Pi_1)$ acceptance condition. Note that in this game, player I has no winning strategies. Assume that player II has a winning strategy τ , then the halting set for \mathcal{R} is

$$L(\mathcal{R}) = \{m : \text{player II follows her winning strategy } \tau \text{ and answers "yes" with } m \text{ many 1's in a } \mathbf{2DVPL}_{\omega}(\Sigma_1 \wedge \Pi_1) \text{ game } \mathbb{G}_{\mathcal{R}}\}.$$

Since the halting problem of \mathcal{R} is Σ_1^0 -complete, any winning strategy for player II is Σ_1^0 -hard. \square

In the same way as the proof of Theorem 3.2, we can also show that:

Corollary 3.3. *For any n , there exists an infinite game in $\mathbf{2DVPL}_{\omega}(\mathcal{B}(\Sigma_1))$ with only Σ_n^0 -hard winning strategies.*

In other words, the corollary means that, for any n , there exists a deterministic 2-stack visibly pushdown automaton \mathcal{M} with a $\mathcal{B}(\Sigma_1)$ acceptance condition such that in the game $\mathbb{G}(L(\mathcal{M}))$, player II has a winning strategy and all the winning strategies are Σ_n^0 -hard.

An idea of the proof of Corollary 3.3 is as follows. Take the case $n = 3$ as an example. Let A be any Σ_3^0 set. Then there is a 2-counter automaton \mathcal{R} such that $m_0 \in A$ if and only if $\exists m_1 \forall m_2 \mathcal{R}$ halts on $m = 2^{m_0} 3^{m_1} 5^{m_2}$. Now we consider the following game.

- Player I starts the game by asking if $m_0 \in A$.
- Player II answers yes or no.
 - If player II answers yes, she also needs to choose m_1 , and then player I chooses m_2 . After that, player II constructs a sequence of configurations of \mathcal{R} on $m = 2^{m_0} 3^{m_1} 5^{m_2}$ in the same way as in the proof for Theorem 3.2.
 - If player II answers no, then the game continues similarly as the roles of the players switched.

Theorem 3.2 (and Cor. 3.3) and their proofs can be easily formalized in second order arithmetic. However, to get a statement nicely fit for the classification due to reverse mathematics, we shall consider deterministic 2-stack visibly pushdown automata with an oracle tape and obtain the corresponding boldface classes of ω -languages.

An oracle tape is a read-only, non-real-time infinite tape and distinct from the input tape. It serves as an oracle function $f : \omega \rightarrow \omega$ in the form of $1^{f(0)} 0 1^{f(1)} 0 1^{f(2)} \dots$. Such an oracle is similar with that used in [20]. In the following, by $\mathbf{2DVPL}_\omega(\mathbf{C})$ for a boldface acceptance condition \mathbf{C} , we denote the boldface class of ω -languages accepted by the corresponding deterministic 2-stack visibly pushdown automata with an oracle tape.

Corollary 3.4. *The determinacy of games in $\mathbf{2DVPL}_\omega(\Sigma_1 \wedge \Pi_1)$ implies ACA_0 . In fact, they are equivalent to each other over RCA_0 .*

Proof. Since ACA_0 is equivalent to $\Sigma_1^0\text{-CA}_0$, it suffices to show that the determinacy implies the existence of Σ_1^0 sets with any oracle.

We consider a 3-counter automaton $\mathcal{R}^f = (Q \cup \{q^\diamond\}, \delta \cup \delta^\diamond, q_{\text{in}}, F)$ with a function $f : \omega \rightarrow \omega$. q^\diamond is a query state associated with the content of the third counter and the function f . Intuitively, the transition relation $\delta \cup \delta^\diamond$ satisfies that if \mathcal{R}^f is in a state belonging to Q , it works as a usual 3-counter automaton; if it reaches the query state q^\diamond , it will query the value of f with current number in the third counter, update the third counter with the returned value of f and go to next state. We refer the latter case as the *query transition*.

A configuration (q, m, n, k) of \mathcal{R}^f , where $q \in (Q \cup \{q^\diamond\})$ and m, n, k denote the contents of the three counters, is coded as $qa^m b^n e^k$. We define $L(\mathcal{R}^f)$ by $m \in L(\mathcal{R}^f)$ if and only if there exists a run on m such that

$$q_{\text{in}} a^m b^{n_0} e^{k_0} \mapsto_{\mathcal{R}^f} q_1 a^{m_1} b^{n_1} e^{k_1} \mapsto_{\mathcal{R}^f} \dots \mapsto_{\mathcal{R}^f} q_s a^{m_s} b^{n_s} e^{k_s},$$

where $n_0 = k_0 = 0$ and q_s is a halting state.

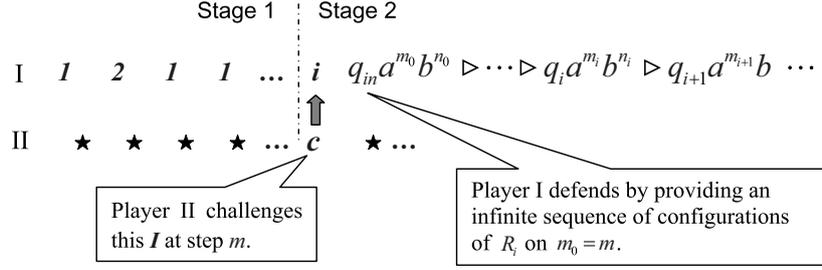
The game $\mathbb{G}_{\mathcal{R}^f}$ proceeds in the same way as $\mathbb{G}_{\mathcal{R}}$ in Theorem 3.2, except that it also needs to make sure that for each appearance of the query state q^\diamond in the computation sequence of \mathcal{R}^f , the content in the third counter is updated according to the query transition of \mathcal{R}^f .

In this game, player II has a winning strategy and her winning set can be accepted by a deterministic 2-stack visibly pushdown automaton with a $(\Sigma_1 \vee \Pi_1)$ condition and an oracle tape in the form of $1^{f(0)} 0 1^{f(1)} 0 1^{f(2)} \dots$, in which the oracle tape is used to check the query transition is correct or not in the computation sequence of \mathcal{R}^f . Such automata in fact defines the corresponding boldface class $\mathbf{2DVPL}_\omega(\Sigma_1 \vee \Pi_1)$.

Assume that player II has a winning strategy τ . Then we have

$$L(\mathcal{R}^f) = \{m : \text{player II follows her winning strategy } \tau \text{ and answers "yes" with } m \\ \text{many 1's in a } \mathbf{2DVPL}_\omega(\Sigma_1 \wedge \Pi_1) \text{ game } \mathbb{G}_{\mathcal{R}^f}\}.$$

To show the equivalence, it is enough to see that this game is a $\Sigma_1 \wedge \Pi_1$ game in the Cantor space, where the determinacy follows from ACA_0 [28]. \square

FIGURE 6. A game $\mathbb{G}_{\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset}$ with challenges.

We move on to treat infinite games whose winning sets are ω -languages recognized by deterministic 2-stack visibly pushdown automata with a Π_1 acceptance condition.

Theorem 3.5. *The determinacy of games in $\mathbf{2DVPL}_\omega(\Pi_1)$ implies the separation principle for Σ_1^0 set without parameters, namely, $\Sigma_1^0\text{-SP}^-$.*

Proof. Let \mathcal{R}_1 and \mathcal{R}_2 be two 2-counter automata such that $L(\mathcal{R}_1) \cap L(\mathcal{R}_2) = \emptyset$. We are looking for a set to separate $L(\mathcal{R}_1)$ and $L(\mathcal{R}_2)$. Or equivalently, we want a function $g : \omega \rightarrow \{1, 2\}$ such that for each $m \in \omega$, $m \notin L(\mathcal{R}_{g(m)})$. We obtain such a g from a winning strategy of the following two-stage game $\mathbb{G}_{\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset}$.

In the first stage, player I produces a sequence g of 1's and 2's such that at each step m , $\mathcal{R}_{g(m)}$ should not halt with m . Player II may challenge player I's choice i at any m if she thinks the 2-counter automaton \mathcal{R}_i halts with this m , or choose to pass if she agrees with player I.

If player II challenges at a certain m , then the game enters the second stage and player I defends by producing an infinite sequence of configurations of \mathcal{R}_i on m , $q_{i_0} a^{m_0} b^{n_0} \triangleright q_1 a^{m_1} b^{n_1} \dots$, where $m_0 = m$ and $n_0 = 0$, as illustrated in Figure 6.

While player I is making such an infinite sequence of configurations of \mathcal{R}_i , if player II perceives that player I has cheated, she may challenge again. Then the game continues in the same way as the challenge part in Theorem 3.2.

The winning set for player I can be described as

- (1) player II never challenge, namely, player I eventually produces an infinite sequence from $\{1, 2\}$,
- (2) player II proposes a challenge at a certain m in the first stage but no challenge in the second stage, or
- (3) player II challenges in both stages, but fails in the challenge in the second stage.

We can see that player I's winning sets can be recognized by deterministic 2-stack visibly pushdown automata with a Π_1 acceptance condition. Assume that player I has a winning strategy σ in the game $\mathbb{G}_{\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset}$, then the desired separating set is

$$S = \{m \in \omega : \text{Following strategy } \sigma, \text{ player I picks 2 for } m \text{ in the first stage of } \mathbb{G}_{\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset}\}. \quad \square$$

Considering the corresponding 2-stack visibly pushdown automata with an oracle tape, we can show that

Corollary 3.6. *The determinacy of games in $\mathbf{2DVPL}_\omega(\Pi_1)$ is equivalent to WKL_0 over RCA_0 .*

Proof. Similarly, we can construct an infinite game such that the winning set for player I is recognized by deterministic 2-stack visibly pushdown automaton with a Π_1 acceptance condition and an oracle tape. By $\text{WKL}_0 \equiv \Sigma_1^0\text{-SP}_0$ and Theorem 3.5, the determinacy of such games implies WKL_0 . It follows from [28] that $\text{WKL}_0 \rightarrow \Sigma_1^0\text{-Det}(\text{in } 2^\omega) \rightarrow \mathbf{2DVPL}_\omega(\Sigma_1)\text{-Det} \leftrightarrow \mathbf{2DVPL}_\omega(\Pi_1)\text{-Det}$. \square

To treat the determinacy of $\mathbf{2DVPL}_\omega(\Delta_2)$, we first recall a coding technique to convert sequences from ω^ω to 2^ω [28]. For a finite sequence $u \in 2^{<\omega}$, u is called *regular*, if it satisfies:

- (r1) $u(|u| - 1) = 1 - u(|u| - 2)$,
- (r2) $u(2m) = 0$ implies $u(2m + 1) = 0$ for all $2m + 1 < |u|$, and
- (r3) $u(2m + 1) = 1$ implies $u(2m + 2) = 1$ for all $2m + 2 < |u|$.

A regular $u \in 2^{<\omega}$ determines a unique sequence in $\tilde{u} \in \omega^{<\omega}$, such that

$$u = \underbrace{0 \dots 0}_{2\tilde{u}(0)} 1 \underbrace{1 \dots 1}_{2\tilde{u}(1)} 0 \dots i_n, \underbrace{i_n \dots i_n}_{2\tilde{u}(|\tilde{u}|-1)} (1 - i_n),$$

where $i_n = 0$ for $n = 2m$ and $i_n = 1$ for $n = 2m + 1$. Note that \tilde{u} is defined recursively from u . The statement “ s is regular” is a Π_0^0 statement.

For an infinite sequence $\alpha \in 2^\omega$, α is called *totally regular* if it satisfies

$$\underbrace{\forall n \exists m > n \underbrace{\alpha[m] \text{ is regular.}}_{\Pi_0^0}}_{\Pi_2^0}$$

A totally regular $\alpha \in 2^\omega$ determines a unique sequence in $\tilde{\alpha} \in \omega^\omega$, such that

$$\alpha = \underbrace{0 \dots 0}_{2\tilde{\alpha}(0)} 1 \underbrace{1 \dots 1}_{2\tilde{\alpha}(1)} 0 \underbrace{0 \dots 0}_{2\tilde{\alpha}(2)} 1 \dots$$

We also adopt the “translation rules” for players from [28], which is used to convert games from ω^ω to 2^ω . For an infinite play $\alpha \in 2^\omega$, we say that “player I (respectively, player II) obeys the translation rules” if either of the following cases holds.

- (1) It is player II (respectively, player I) who first plays against r2 or r3.
- (2) Neither player I nor player II plays against r2 or r3, and for all n , $\alpha(n) = 0$ (respectively, $\alpha(n) = 1$) implies there exists $m > n$, $\alpha(m) = 1$ (respectively, $\alpha(m) = 0$).

We can see that “player I (or player II) obeys the translation rules” is a Π_2^0 statement.

Then we show that:

Theorem 3.7. *The determinacy of $\mathbf{2DVPL}_\omega(\Delta_2)$ implies the determinacy of Δ_1^0 games in ω^ω .*

Proof. By the coding technique in [28] as described above, we write $\tilde{\alpha} \in \omega^\omega$ for the unique sequence coded by $\alpha \in 2^\omega$. Note that not all sequences in 2^ω code a sequence in ω^ω .

Then, a play $\tilde{\alpha}$ in a Δ_1^0 game in ω^ω can be translated into a play α in 2^ω , and α is winning for player 1 (respectively, player 2) if and only if

- (a) $\tilde{\alpha}$ is a winning play (respectively, $\tilde{\alpha}$ is not a winning play) in the Δ_1^0 game in ω^ω while both players obey the translation rules to produce a play α , or
- (b) while they are producing α , player 2 (respectively, player 1) breaks the rules,

which constitutes a Σ_2^0 winning set for player 1 (respectively, player 2). Thus the game is Δ_2^0 in 2^ω . Note that the increase in complexity of winning condition is mainly due to the complexity of the coding rules that we follow.

Now we convert this Δ_1^0 game in ω^ω to a $\mathbf{2DVPL}_\omega(\Delta_2)$ game relying on the translation rules used in the above Δ_2^0 game in 2^ω . The translation rules do not need any modification for $\mathbf{2DVPL}_\omega(\Delta_2)$. So, for simplicity, the two players are assumed to obey this translation rule and we just treat the above case (a).

Given a Δ_1^0 game, there exist two 2-counter automata \mathcal{R}_1 and \mathcal{R}_2 such that

$$x \text{ is a winning play} \leftrightarrow \exists n \#x[n] \in L(\mathcal{R}_1) \leftrightarrow \neg \exists n \#x[n] \in L(\mathcal{R}_2),$$

where $\#x[n]$ denotes a code of the initial n -segment of s .

We construct a game $\mathbb{G}_{\mathcal{R}_1, \mathcal{R}_2}$ as follows.

- When a player produces a finite sequence $\alpha[n]$ in the Δ_2^0 game in 2^ω such that $\#\tilde{\alpha}[n] \in L(\mathcal{R}_i) (i \in \{1, 2\})$, player i in the game $\mathbb{G}_{\mathcal{R}_1, \mathcal{R}_2}$ starts providing a sequence of configurations of \mathcal{R}_i on $\#\tilde{\alpha}[n]$, which player i claims to halt in finite steps.
- While player i is making such a sequence of configurations of \mathcal{R}_i on $\#\tilde{\alpha}[n]$, the player $3 - i$ may challenge at any point.

We can see that the winning set for player 1 in the constructed game $\mathbb{G}_{\mathcal{R}_1, \mathcal{R}_2}$ is in $\mathbf{2DVPL}_\omega(\Delta_2)$. Moreover, if player i has a winning strategy in $\mathbf{2DVPL}_\omega(\Delta_2)$, then player i in the original Δ_1^0 game in ω^ω also has a winning strategy. \square

Corollary 3.8. *The determinacy of games in $\mathbf{2DVPL}_\omega(\Delta_2)$, or $\mathbf{2DVPL}_\omega(\Sigma_2)$, is equivalent to ATR_0 over RCA_0 .*

Proof. Since $\text{ATR}_0 \leftrightarrow \Delta_1^0\text{-Det}$, by Theorem 3.7, the determinacy of $\mathbf{2DVPL}_\omega(\Delta_2)$ implies ATR_0 . For the other direction, it follows from [28] that

$$\text{ATR}_0 \rightarrow \Sigma_2^0\text{-Det(in } 2^\omega) \rightarrow \text{DTM}_\omega(\Sigma_2)\text{-Det} \rightarrow \mathbf{2DVPL}_\omega(\Sigma_2)\text{-Det} \rightarrow \mathbf{2DVPL}_\omega(\Delta_2)\text{-Det.} \quad \square$$

Next, we will show that most of the above results for deterministic 2-stack visibly pushdown automata also hold for nondeterministic ones. Similarly, by $\mathbf{2VPL}_\omega(\mathbf{C})$ with a boldface acceptance condition \mathbf{C} , we denote the boldface class of ω -languages accepted by the corresponding nondeterministic 2-stack visibly pushdown automata with an oracle tape.

Theorem 3.9. *For an acceptance condition $\mathbf{C} \in \{\Sigma_1, \Pi_1, \Sigma_1 \wedge \Pi_1, \Delta_2, \Sigma_2\}$, RCA_0 proves*

$$\mathbf{2DVPL}_\omega(\mathbf{C})\text{-Det} \leftrightarrow \mathbf{2VPL}_\omega(\mathbf{C})\text{-Det} \leftrightarrow \text{TM}_\omega(\mathbf{C})\text{-Det.}$$

Proof. Given an acceptance condition $\mathbf{C} \in \{\Sigma_1, \Pi_1, \Sigma_1 \wedge \Pi_1, \Delta_2, \Sigma_2\}$, we know that

$$\text{TM}_\omega(\mathbf{C}) \supseteq \mathbf{2VPL}_\omega(\mathbf{C}) \supseteq \mathbf{2DVPL}_\omega(\mathbf{C}).$$

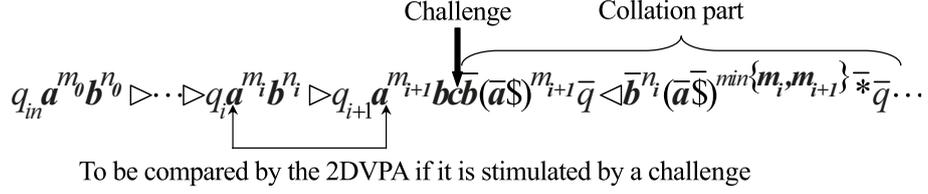
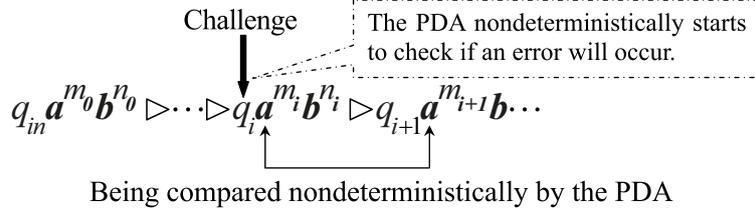
So, $\text{TM}_\omega(\mathbf{C})\text{-Det} \rightarrow \mathbf{2DVPL}_\omega(\mathbf{C})\text{-Det}$.

The other directions follow from the above results on $\mathbf{2DVPL}_\omega(\mathbf{C})$, together with the reverse mathematical investigations of determinacy [28]. \square

We also conjecture that $\mathbf{2DVPL}_\omega(\mathcal{B}(\Sigma_2))\text{-Det}$ is equivalent to $\mathcal{B}(\Sigma_2^0)\text{-Det}$, which is proof theoretically equivalent to $\Pi_2^1\text{-CA}_0$ (cf. [24]), and that $\mathbf{2VPL}_\omega(\Pi_2)\text{-Det}$ is equivalent to $\Sigma_1^1\text{-Det}$, which is known to be equivalent to the determinacy of pushdown ω -languages with a Π_2 acceptance condition [14]. Note that for deterministic case, $\mathbf{2DVPL}_\omega(\Pi_2)\text{-Det}$ is equivalent to $\mathbf{2DVPL}_\omega(\Sigma_2)\text{-Det}$.

4. DETERMINACY STRENGTH OF INFINITE GAMES RECOGNIZED BY NONDETERMINISTIC PUSHDOWN AUTOMATA

In this section we use the studies on deterministic 2-stack visibly pushdown automata in Section 3 to analyze the determinacy strength of infinite games whose winning sets are ω -languages recognized by pushdown automata with various acceptance conditions. Following the notations in Section 3, we define the ω -languages

FIGURE 7. A play in a 2DVPL_ω game.FIGURE 8. A play in a $r\text{-PDL}_\omega$ game.

$\mathbf{PDL}_\omega(\Sigma_1)$, $\mathbf{PDL}_\omega(\Pi_1)$, $\mathbf{PDL}_\omega(\Sigma_2)$, $\mathbf{PDL}_\omega(\Pi_2)$, $\mathbf{PDL}_\omega(\Sigma_1 \wedge \Pi_1)$ and $\mathbf{PDL}_\omega(\Delta_2)$. Moreover, we put r in front of the pushdown ω -languages, for instance $r\text{-PDL}_\omega(\Sigma_1)$, to denote the ω -languages accepted by the corresponding real-time pushdown automata.

By the compactness arguments, $\mathbf{PDL}_\omega(\Pi_1)$ (respectively, $\mathbf{PDL}_\omega(\Sigma_1)$, $\mathbf{PDL}_\omega(\Sigma_2)$, $\mathbf{PDL}_\omega(\Pi_2)$) is a subclass of the effective Π_1^0 (respectively, Σ_1^0 , Σ_2^0 , Σ_1^1) class (*cf.* [33]), and so does the corresponding real-time class. Similarly, the boldface pushdown ω -languages are included in their counterparts of Borel and analytic classes.

In Section 3, we use deterministic 2-stack visibly pushdown automata to check if an error has occurred or not in the code of configuration sequence of a 2-counter automaton, which is done in the collation part as shown in Figure 7.

Such a check can be carried out more easily by a nondeterministic pushdown automaton. Instead of players' collation parts, a pushdown automaton can nondeterministically check whether an error occurs or not as illustrated in Figure 8.

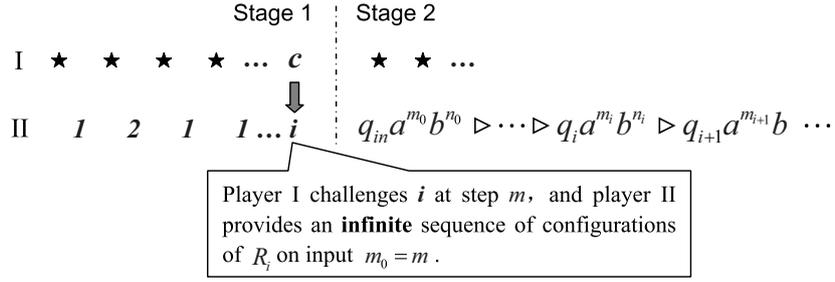
We begin with analyzing the complexity of winning strategies in the infinite games recognized by real-time pushdown automata with Σ_1 acceptance condition.

Theorem 4.1. *The determinacy of games in $r\text{-PDL}_\omega(\Sigma_1)$ implies $\Sigma_1^0\text{-SP}^-$.*

Proof. This is a straightforward adaptation of the proof of Theorem 3.5. We construct a two-stage game $\mathbb{G}_{\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset}$, where \mathcal{R}_1 and \mathcal{R}_2 are two 2-counter automata such that $L(\mathcal{R}_1) \cap L(\mathcal{R}_2) = \emptyset$. The game constructed here is no difference from that we used in the proof of Theorem 3.5 except that the two players interchange their roles and player I only challenges in the first stage.

In the first stage, it is player II who produces a sequence g . For each m , player II chooses $i \in \{1, 2\}$ such that \mathcal{R}_i does not halt with m . Player I attempts to detect player II's lie and challenge player II's choice i at any m if she thinks the 2-counter automaton \mathcal{R}_i halts with this m , or choose to pass if he agrees with player II. If player I never find the opportunity to challenge, he loses.

If player I challenges at a step m , the game enters the second stage. Then player II defends by producing an infinite sequence of configurations of \mathcal{R}_i on m , $q_{in} a^{m_0} b^{n_0} \triangleright q_1 a^{m_1} b^{n_1} \dots$, where $m_0 = m$ and $n_0 = 0$, as illustrated in Figure 9.

FIGURE 9. A game $\mathbb{G}_{\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset}$.

Player I wins if there exist some computation errors in the infinite sequence of configurations of the 2-counter automaton, which is provided by player II. We can see that the winning plays for player I in the game $\mathbb{G}_{\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset}$ can be recognized by a real-time nondeterministic pushdown automaton with a Σ_1 acceptance condition.

Notice that in this game, player II always wins if she plays rationally while player I has no winning strategy. Assume that player II has a winning strategy τ in the game $\mathbb{G}_{\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset}$, then the desired separating set is

$$S = \{m \in \omega : \text{Following strategy } \tau, \text{ player II picks 2 at } m \text{ in the first stage of the game } \mathbb{G}_{\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset}\}.$$

□

Similarly, when we turn to the reverse mathematical statement of the above result, we need to consider the corresponding boldface classes of ω -languages which are accepted by such pushdown automata with an oracle tape. The oracle tape is defined in the same way as Section 3. We can show that:

Corollary 4.2. *The determinacy of the games in $r\text{-PDL}_\omega(\Sigma_1)$ is equivalent to WKL_0 over RCA_0 .*

Note that, for nondeterministic pushdown ω -languages, a safety condition is not the complement of a reachability case. In contrast with the reachability case, we show that

Theorem 4.3. $\text{RCA}_0 \vdash \text{PDL}_\omega(\Pi_1)\text{-Det}$.

Proof. We shall prove that any game in $\text{PDL}_\omega(\Pi_1)$ is effectively determined with computable winning strategies by constructing an equivalent pushdown game with a Π_1 winning condition, which it known to be effectively determined with computable winning strategies.

Pushdown processes can be regarded as pushdown automata without input alphabet and labels on the transitions. Given a pushdown automaton $\mathcal{M} = (Q, X, \Gamma, q_{\text{in}}, \delta, F)$ with a Π_1 condition, we can formulate a pushdown process $\mathcal{P}_{\mathcal{M}} = (Q, \Gamma, V_1, V_2, \widehat{\delta})$, where

- $V_i = \{i\}Q\Gamma^{<\omega}$ for $i \in \{1, 2\}$, and
- $\widehat{\delta} = \left\{ \langle (i, p, \nu\gamma), (3-i, q, w\gamma) \rangle : \exists a \in X \cup \{\varepsilon\} \underbrace{\left((q, w) \in \delta(p, a, \nu) \wedge q \in F \right)}_{(a)} \right\}$

$\underbrace{\left((q, w) \in \delta(p, a, \nu) \wedge \forall q' \in F(q', w) \notin \delta(p, a, \nu) \right)}_{(b)}, i \in \{1, 2\}, \nu \in \Gamma, w \in \Gamma^{\leq 2}, \gamma \in (\Gamma \setminus \{\perp\})^{<\omega} \{\perp\}$ is the transition relation.

Intuitively, the transition of $\mathcal{P}_{\mathcal{M}}$ consists of, all the safe transitions of \mathcal{M} as shown in (a) and transitions to keep the continuity property as shown in (b).

A pushdown game graph $\mathcal{G}_{\mathcal{P}}$ can be generated by such a pushdown process, in which the set of vertices V is a subset of $V_1 \cup V_2$ and the set of edges $E \subseteq V \times V$ is defined *via* $\hat{\delta}$. We fix the configuration $(1, q_{\text{in}}, \perp)$ as the initial vertex.

In the game $\mathcal{G}_{\mathcal{P}}$, at a vertex $w \in V_i$, player i chooses a vertex $v \in V_{3-i}$ such that $(w, v) \in E$. Thus the two players choose alternatively with player 1's first position. Finally they produce an infinite sequence $\rho = v_1 v_2 \dots$ of V^ω . The deadlock case is avoided due to the construction of $\mathcal{P}_{\mathcal{M}}$.

A strategy for player i ($i \in \{1, 2\}$) is a mapping $f_i: V^*V_i \rightarrow V$. The winning set for player 1 is $T = \{\rho \in (\{0, 1\}F(F \setminus \{\perp\})^{<\omega}\{\perp\})^\omega : \rho \text{ defines an accepting run of } \mathcal{M}\}$. Player 1 wins with the play ρ if and only if $\rho \in T$, otherwise player 2 wins.

A pushdown game $\mathcal{G}_{\mathcal{P}}$ can be seen as the pushdown automaton \mathcal{M} in the sense that for all $\alpha \in L(\mathcal{M})$ (which is called an external winning condition) α induces a winning play belonging to T (which is called an internal winning condition) in $\mathcal{G}_{\mathcal{P}}$ and *vice versa*.

We now construct the following infinite game associated with \mathcal{M} , denoted by $\mathbb{G}(L(\mathcal{M}))$, where the two players produce a play x that simulates the play ρ in $\mathcal{G}_{\mathcal{P}}$: for all $j \geq 1$, player i in $\mathbb{G}(L(\mathcal{M}))$ chooses $a_j = \alpha(j)$ if player i in the game $\mathcal{G}_{\mathcal{P}}$ chooses v_j such that there exists $\alpha(j) : v_{j-1} \mapsto_{\mathcal{M}} v_j$, $v_{j-1} \in V_i$ and $v_j \in V_{3-i}$, where $i \in \{1, 2\}$ and $\alpha(j) \in X \cup \{\varepsilon\}$.

From the construction, we can see that player i has a winning strategy in $\mathcal{G}_{\mathcal{P}}$ iff play i has a winning strategy in $\mathbb{G}(L(\mathcal{M}))$, if it exists. Moreover, if there exists a computable winning strategy f_i for player i in $\mathcal{G}_{\mathcal{P}}$, then there exists a winning strategy f'_i for player i in $\mathbb{G}(L(\mathcal{M}))$ which is computable from f_i , and *vice versa*.

By Walukiewicz [39, 40], there is a computable winning strategy in $\mathcal{G}_{\mathcal{P}}$. Thus in the game $\mathbb{G}(L(\mathcal{M}))$ we can construct a computable winning strategy from that in $\mathcal{G}_{\mathcal{P}}$. \square

Before we move on to treat other pushdown ω -languages, we remark again the resemblance between infinite games recognized by nondeterministic pushdown automata and those by deterministic 2-stack visibly pushdown automata with the same acceptance conditions. Intuitively, deterministic 2-stack visibly pushdown automaton can check an error has occurred or not "in the history", while a pushdown automaton can nondeterministically predict an occurrence of an error "in the future" and execute a subsequent check.

A careful examination of this argument and the proofs for $\mathbf{2DVPL}_\omega$ games in Section 3, together with the proof that we presented in Theorem 4.1 as an example, reveals the following analogous results for pushdown ω -languages as stated in Theorem 4.4. It is worth noting that all the following equivalences are established based on infinite games defined by (real-time) pushdown automata with an oracle tape, which are developed in order to keep in harmony with the classification of reverse mathematics, and also consistent with the results in Section 3.

Theorem 4.4. *The following diagram holds over RCA_0 .*

$$\begin{array}{ccccc}
r\text{-PDL}_\omega(\Sigma_2)\text{-Det} & \leftrightarrow \text{ATR}_0 \leftrightarrow & \mathbf{2DVPL}_\omega(\Sigma_2)\text{-Det} & \leftrightarrow & \mathbf{2DVPL}_\omega(\Pi_2)\text{-Det} \\
\updownarrow & & \updownarrow & & \updownarrow \\
r\text{-PDL}_\omega(\Delta_2)\text{-Det} & \leftrightarrow \Delta_1^0\text{-Det} \leftrightarrow & \mathbf{2DVPL}_\omega(\Delta_2)\text{-Det} & & \\
& \downarrow & & & \\
r\text{-PDL}_\omega(\Sigma_1 \wedge \Pi_1)\text{-Det} & \leftrightarrow \text{ACA}_0 \leftrightarrow & \mathbf{2DVPL}_\omega(\Sigma_1 \wedge \Pi_1)\text{-Det} & & \\
& \downarrow & & & \\
r\text{-PDL}_\omega(\Sigma_1)\text{-Det} & \leftrightarrow \text{WKL}_0 \leftrightarrow & \mathbf{2DVPL}_\omega(\Sigma_1)\text{-Det} & \leftrightarrow & \mathbf{2DVPL}_\omega(\Pi_1)\text{-Det}
\end{array}$$

Recall that the determinacy of infinite games recognized by pushdown automata with a Π_2 (Büchi) acceptance condition (without an oracle tape) is equivalent to the determinacy of effective analytic games, which is not provable in ZFC [14].

Moreover, the above results for real-time pushdown automata also hold for non-real-time ones.

Theorem 4.5. *For an acceptance condition $\mathbf{C} \in \{\Sigma_1, \Sigma_1 \wedge \Pi_1, \Delta_2, \Sigma_2\}$, RCA_0 proves*

$$r\text{-PDL}_\omega(\mathbf{C})\text{-Det} \leftrightarrow \text{PDL}_\omega(\mathbf{C})\text{-Det} \leftrightarrow \text{TM}_\omega(\mathbf{C})\text{-Det}.$$

Proof. Given an acceptance condition $\mathbf{C} \in \{\Sigma_1, \Sigma_1 \wedge \Pi_1, \Delta_2, \Sigma_2\}$, we know that

$$\mathbf{TM}_\omega(\mathbf{C}) \supseteq \mathbf{PDF}_\omega(\mathbf{C}) \supseteq r\text{-PDL}_\omega(\mathbf{C}).$$

Then we have $\mathbf{TM}_\omega(\mathbf{C})\text{-Det} \rightarrow \mathbf{PDL}_\omega(\mathbf{C})\text{-Det}$.

The other directions follow from the above results on $r\text{-PDL}_\omega(\mathbf{C})$ in Theorem 4.4, together with the reverse mathematical investigations of determinacy in [28]. \square

Finally, we can easily observe that all the arguments about pushdown automata in this section are, in fact, replaced by (nondeterministic) 1-counter automata, namely pushdown automata that can check whether the counter is zero or not with only one stack symbol. The ω -languages recognized by 1-counter automata (respectively, real-time 1-counter automata) with boldface acceptance condition \mathbf{C} is denoted as $\mathbf{CL}_\omega(\mathbf{C})$ (respectively, $r\text{-CL}_\omega(\mathbf{C})$). We show that

Theorem 4.6. *For an acceptance condition $\mathbf{C} \in \{\Sigma_1, \Sigma_1 \wedge \Pi_1, \Delta_2, \Sigma_2\}$, RCA_0 proves*

$$r\text{-CL}_\omega(\mathbf{C})\text{-Det} \leftrightarrow \mathbf{CL}_\omega(\mathbf{C})\text{-Det} \leftrightarrow \mathbf{TM}_\omega(\mathbf{C})\text{-Det}.$$

5. CONCLUSIONS

In this study, we proved that infinite games in several pushdown ω -languages with lower Borel complexity are still highly undecidable. It also remains to investigate the determinacy strength of ω -languages recognized by other types of machines and/or other acceptance conditions, which approaches to the boundary line between decidable and undecidable infinite games in Figure 2. For the classes of ω -languages accepted by 2-stack visibly pushdown automata, we studied some of their determinacy strength with acceptance conditions below Δ_2 and conjectured for $\mathbf{2DVPL}_\omega(\mathcal{B}(\Sigma_2))\text{-Det}$, as well as $\mathbf{2VPL}_\omega(\mathcal{B}(\Pi_2))\text{-Det}$.

An interesting problem is to extend studies to infinite games recognized by probabilistic automata. The probabilistic automata is a generalization of finite state automata, of which the nondeterministic transitions are replaced by transitions with (rational) probabilistic distributions. The acceptance of a word is assessed by not only the normal conditions, like Büchi, but also the acceptance probability of this word. Let $L(\text{PBA}^{>0})$ (respectively, $L(\text{PBA}^{=1})$) denote the class of ω -languages accepted by probabilistic Büchi automata with acceptance probabilities larger than 0 (respectively, equal to 1). It is known that $L(\text{PBA}^{>0}) \subseteq \mathcal{B}(\Pi_2^0)$ and $L(\text{PBA}^{=1}) \subseteq \Pi_2^0$ [9]. One possible direction is to investigate the determinacy strength of games in $L(\text{PBA}^{>0})$ (respectively, $L(\text{PBA}^{=1})$), and particularly, whether the determinacy of such games is equivalent to those in $\mathcal{B}(\Pi_2^0)$ (respectively, Π_2^0) or not. Furthermore, the investigations on their relations with the determinacy of stochastic games (*e.g.*, Blackwell determinacy) will appear in the future literature.

Acknowledgements. We would like to thank the anonymous referees for their careful review and valuable comments.

REFERENCES

- [1] R. Alur and P. Madhusudan, Visibly pushdown languages, in *Proc. of the 36th Annual ACM Symposium on Theory of Computing, STOC'04* (2004) 202–211.
- [2] B. Bollig, On the expressive power of 2-stack visibly pushdown automata. *Log. Methods Comput. Sci.* **4** (2008) 1–35.
- [3] J.R. Büchi and L.H. Landweber, Solving sequential conditions by finite-state strategies. *Trans. Amer. Math. Soc.* **138** (1969) 295–311.
- [4] T. Cachat, Higher order pushdown automata, the Caucal hierarchy of graphs and parity games, in *International Colloquium on Automata, Languages, and Programming*. Springer Berlin Heidelberg (2003) 556–569.

- [5] T. Cachat, J. Duparc and W. Thomas, Solving pushdown games with a Σ_3^0 winning condition, in *International Workshop on Computer Science Logic*. Springer Berlin Heidelberg (2002) 322–336.
- [6] A. Carayol, M. Hague, A. Meyer, C.H. Ong and O. Serre, Winning regions of higher-order pushdown games, in *LICS '08 Proc. of the 2008 23rd Annual IEEE Symposium on Logic in Computer Science* (2008) 193–204.
- [7] D. Carotenuto, A. Murano and A. Peron, Ordered multi-stack visibly pushdown automata, *Theoret. Comput. Sci.* **656** (2016) 1–26.
- [8] D. Carotenuto, A. Murano and A. Peron, 2-visibly pushdown automata, in *International Conference Developments in Language Theory*. Springer Berlin Heidelberg (2007) 132–144.
- [9] R. Chadha, A.P. Sistla and M. Viswanathan, Power of randomization in automata on infinite strings. *Log. Methods Comput. Sci.* **3** (2011) 1–31.
- [10] R.S. Cohen and A.Y. Gold, Theory of ω -languages. Part I: Characterization of ω -context-free languages. *J. Comput. Syst. Sci.* **15** (1977) 169–184.
- [11] J. Engelfriet and H.J. Hoogeboom, X -automata on ω -words. *Theoret. Comput. Sci.* **110** (1993) 1–51.
- [12] O. Finkel, Infinite games specified by 2-tape automata. *Ann. Pure Appl. Logic* **167** (2016) 1184–1212.
- [13] O. Finkel, Topological complexity of context free ω -languages: A survey, in *Language, Culture, Computation: Studies in Honor of Yaacov Choueka*, vol. 8001 of *Lect. Notes Comput. Sci.* Springer (2014) 50–77.
- [14] O. Finkel, The determinacy of context-free games. *J. Symb. Log.* **78** (2013) 1115–1134.
- [15] O. Finkel, Borel ranks and Wadge degrees of omega context free languages. *Math. Struct. Comput. Sci.* **16** (2006) 813–840.
- [16] O. Finkel, On omega context free languages which are Borel sets of infinite rank. *Theoret. Comput. Sci.* **299** (2003) 327–346.
- [17] O. Finkel, Topological properties of omega context-free languages. *Theoret. Comput. Sci.* **262** (2001) 669–697.
- [18] L. Harrington, Analytic determinacy and 0^\sharp . *J. Symb. Log.* **43** (1978) 685–693.
- [19] D.R. Hirschfeldt, Slicing the truth: On the computable and reverse mathematics of combinatorial principles. In Vol. 28 of *Lect. Notes Ser.*, edited by Institute for Math. Sci., National University of Singapore. World Scientific (2014).
- [20] E. Jeandel, On immortal configurations in Turing machines, in *Conference on Computability in Europe*. Vol. 7318 of *Lect. Notes Comput. Sci.* Springer Berlin Heidelberg (2012) 334–343
- [21] C. Löding, P. Madhusudan and O. Serre, Visibly pushdown games, in *Proc. of Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2004*. Springer Berlin Heidelberg (2005) 408–420.
- [22] D.A. Martin, Borel determinacy. *Ann. Math.* **102** (1975) 363–371.
- [23] D.A. Martin, Measurable cardinals and analytic games. *Fund. Math.* **66** (1969/1970) 287–291.
- [24] M.Y.O. MedSalem and K. Tanaka, Δ_3^0 -determinacy, comprehension and induction. *J. Symb. Log.* **72** (2007) 452–462.
- [25] A. Montalbán and R.A. Shore, The limits of determinacy in second order arithmetic, in *Proc. London Math. Soc.* **104** (2012) 223–252.
- [26] M.L. Minsky, Recursive unsolvability of Post’s problem of tag and other topics in theory of Turing machines. *Ann. Math.* **74** (1961) 437–455.
- [27] M.L. Minsky, *Computation: finite and infinite machines*. Prentice-Hall, Inc. (1967).
- [28] T. Nemoto, M.Y. Ould MedSalem and K. Tanaka, Infinite games in the Cantor space and subsystems of second order arithmetic. *MLQ Math. Log. Q.* **53** (2007) 226–236.
- [29] N. Schweber, Transfinite recursion in higher reverse mathematics. *J. Symb. Log.* **80** (2015) 940–969.
- [30] O. Serre, Games with winning conditions of high Borel complexity, in *International Colloquium on Automata, Languages, and Programming*. Vol. 3142 of *Lect. Notes Math.* Springer Berlin Heidelberg (2004) 1150–1162.
- [31] O. Serre, Games with winning conditions of high Borel complexity. *Theoret. Comput. Sci.* **350** (2006) 345–372.
- [32] S.G. Simpson, *Subsystems of second order arithmetic*. Perspectives in Logic. Association for Symbolic Logic, Poughkeepsie, NY, 2nd edition. Cambridge University Press, Cambridge (2009).
- [33] L. Staiger, ω -languages, in Chapter 6 of the *Handbook of Formal Languages*. Vol. 3, edited by G. Rozenberg and A. Salomaa. Springer Verlag, Berlin (1997) 339–387.
- [34] J.R. Steel, *Determinateness and subsystems of analysis*. Ph.D. thesis, University of California, Berkeley (1977).
- [35] K. Tanaka, *Descriptive set theory and subsystems of analysis*. Ph.D. thesis, University of California, Berkeley (1986).
- [36] S.L. Torre, M. Napoli and G. Parlato, A unifying approach for multistack pushdown automata. *Math. Foundations Comput. Sci.* Springer Berlin Heidelberg (2014) 377–389.
- [37] S.L. Torre, P. Madhusudan and G. Parlato, A robust class of context-sensitive languages, in *22nd Annual IEEE Symposium on Logic in Comput. Sci.* IEEE Ph.D. thesis, University (2007) 161–170.
- [38] W. Thomas, Infinite games and verification (extended abstract of a tutorial). *Lect. Notes Comput. Sci.* (2002) 58–64.
- [39] I. Walukiewicz, Pushdown processes: Games and model-checking, in *International Conference on Computer Aided Verification*. Springer Berlin Heidelberg (1996) 62–74.
- [40] I. Walukiewicz, Pushdown processes: Games and model-checking, *Inf. Comput.* **164** (2001) 234–263.

Communicated by Olivier Serre.

Received July 24, 2016. Accepted June 12, 2017.