# Automata-theoretic study on infinite games and fragments of modal μ-calculus

| | |
|---|---|
| 著者 | LI Wenjuan |
| 学位授与機関 | Tohoku University |
| 学位授与番号 | 11301甲第17839号 |
| URL | http://hdl.handle.net/10097/00123030 |

博士論文

# Automata-theoretic study on infinite games and fragments of modal $\mu$-calculus

(無限ゲームと様相 $\mu$ 計算の部分体系についての
オートマトン理論的研究)

by

Wenjuan Li

(李　文娟)

to

Mathematical Institute, Tohoku University

for the degree of Doctor of Science

平成 29 年

March 2018

# Contents of thesis

1 **General Introduction**

2 **Infinite Games, Automata and Logic**

2.1 Topology of infinite words

2.2 Automata on infinite words

2.2.1 Finite automata and $\omega$-regular languages

2.2.2 Pushdown automata and context free $\omega$-languages

2.2.3 Turing machines and Turing recognizable $\omega$-languages

2.3 Infinite games

2.3.1 Gale-Stewart games

2.3.2 Wadge games

2.3.3 Games on graph

2.4 Modal $\mu$-calculus

2.4.1 Syntax

2.4.2 Semantics

2.5 Second order arithmetic

2.5.1 Subsystems of second order arithmetic

3 **The determinacy strength of pushdown  -languages**

3.1 Motivation

3.2 Preliminaries

3.2.1 Decidability of infinite games recognized by various automata

3.2.2 Visibly pushdown automata

3.2.3 Reverse mathematics and the determinacy of infinite games

3.3 Determinacy strength of infinite games recognized by 2VPA

3.4 Determinacy strength of infinite games recognized by PDA

4 **Alternation hierarchy and fragments of modal $\mu$-calculus**

4.1 Motivation

4.2 Alternation hierarchy of modal $\mu$-calculus

4.3 The strictness of alternation hierarchy

4.3.1 The strictness of alternation hierarchy for arithmetic $\mu$-calculus

4.3.2 The strictness of alternation hierarchy for modal $\mu$-calculus

4.4 Variable hierarchy of modal $\mu$-calculus

4.5 A transfinite extension of modal $\mu$-calculus via arithmetic $\mu$-calculus

4.5.1 Game quantifier, difference hierarchy and arithmetic $\mu$-calculus

4.5.2 A transfinite extension of modal $\mu$-calculus

4.6 Tree automata and parity games for modal $\mu$-calculus

4.7 One variable fragment of modal $\mu$-calculus

4.7.1 Weak alternation hierarchy

4.7.2 Weak alternation hierarchy and one-variable fragment of modal $\mu$-calculus

5 **Appendix**

5.1 Equilibrium points of games trees

5.2 Conclusions and open problems

**Bibliography**

**Index**

# Summary

This thesis is a contribution to the automata-theoretic study on the infinite games and modal $\mu$-calculus. It mainly consists of two parts preceded by an introduction to the fundamental concepts and results on infinite games, automata and logic.

The first part is dedicated to determinacy of infinite games recognized by some variants of pushdown automata, characterizing the complexity of winning sets and winning strategies from an automata-theoretic view.

Particularly, 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 Reverse Mathematics. Notice that infinite games recognized by nondeterministic pushdown automata bear some resemblance to those recognized by deterministic 2-stack visibly pushdown automata with the same acceptance conditions. So, we start with the determinacy of games recognized by deterministic 2-stack visibly pushdown automata, together with those recognized 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 $\omega$-languages with a Büchi condition is known to be independent of $\mathsf{ZFC}$, we here show that for the co-Büchi condition, the determinacy is exactly captured by $\mathsf{ATR}_0$, 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 the first part indeed hold for 1-counter automata.

The second part is concerned with the alternation hierarchy and fragments of modal $\mu$-calculus. We introduce studies on the alternation hierarchy of modal $\mu$-calculus, including its strictness, the relation with arithmetic $\mu$-calculus, the relation with variable hierarchy of modal $\mu$-calculus and the transfinite extension.

Then we concentrate on the one-variable fragment of modal $\mu$-calculus. We introduce the alternation hierarchy of one-variable fragment of modal $\mu$-calculus and prove that, over a certain class of finitely branching transition systems, the simple alternation hierarchy within such a fragment of modal $\mu$-calculus is still strict, which intimately corresponds with the counterparts of weak alternating tree automata and weak games.

## Part I: Determinacy of pushdown $\omega$-languages

Two-player infinite games have been intensively studied in Descriptive Set Theory in the past several decades, mainly focusing on the determinacy of infinite games and the topological complexity of winning conditions. Recall the Gale-Stewart games $\mathbb{G}(X)$ over alphabet $A$, where $X \subseteq A^{\omega}$. In each round, player I and II alternatively choose a letter from the alphabet $A$ and after infinite steps, the two players have produced an infinite word $x$, which is called a play. We say player I wins with the play $x$ if and only if $x \in X$, otherwise player II wins. If player I (resp. player II) can always win no matter how the other player plays, player I (resp. player II) has a winning strategy. The game $\mathbb{G}(X)$ is said to be determined if one of the two players has a winning strategy. One of the most important results due to Martin states that "All Gale-Stewart games with Borel winning conditions are determined" [27].

Büchi and Landweber [8] first paid attention to the computational aspect of winning sets and winning strategies. They studied the Gale-Stewart game $\mathbb{G}(X)$, where $X$ is an $\omega$-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 $\mathbb{G}(X)$ and a winning strategy can be constructed by a finite state transducer.

Walukiewicz [36, 37] 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. Subsequent to Thomas's suggestion for higher Borel games in [35], Cachat, Duparc and Thomas [10] defined a $\mathbf{\Sigma}_3^0$-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] investigated the infinite games with arbitrary finite Borel level by introducing a finite chain of real-time (namely, the $\varepsilon$-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 [26, 9, 11].

On the other hand, for $\omega$-languages accepted by nondeterministic pushdown automata, the situations are quite different. Context-free $\omega$-languages, accepted by nondeterministic Büchi (or Muller) pushdown automata, are beyond finite Borel hierarchy [17]. Finkel proved that the determinacy of context-free $\omega$-languages is equivalent to the determinacy of effective analytic games [16], which is not even provable in the set theory $\mathsf{ZFC}$. In [15], he furthermore showed that there exists an infinite game with an effective $\Delta^0_3$ 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 $\mathsf{RCA}_0$, $\mathsf{WKL}_0$, $\mathsf{ACA}_0$, $\mathsf{ATR}_0$, and $\Pi^1_1$-$\mathsf{CA}_0$, in order of increasing strength. Observe that even full second order arithmetic $\mathsf{Z}_2$ is a much weaker system than $\mathsf{ZFC}$. In particular, $\mathsf{ZFC}$ proves that every Borel game is determined, while $\mathsf{Z}_2$ does not even prove determinacy for general $\Delta^0_4$ games [28]. Note that weaker is *good* in this context, since subsystems of $\mathsf{Z}_2$ can distinguish different kinds of Borel games below $\Delta^0_4$ which are all characterized as determined by $\mathsf{ZFC}$. In fact, studies on determinacy of infinite games are closely connected with the origin and backbone of reverse mathematics (cf.[19, 33, 31, 34]).

In Part I, we downscale Finkel's results to lower levels of Borel hierarchy. We investigate the determinacy strength of infinite games whose winning sets are recognized by variants of 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. 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 $\omega$-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.

We recall the formal definition of pushdown automata is as follows.

**Definition 1.** *A (nondeterministic)* pushdown automaton *(PDA) is a tuple* $\mathcal{M} = (Q, X,$

$\Gamma, q_{in}, \delta, F)$, *where*

- $Q$ *is a finite set of states,*
- $X$ *is a finite input alphabet,*
- $\Gamma$ *is a finite stack alphabet, which includes a special bottom letter* $\perp$,
- $q_{in} \in Q$ *is the initial state,*
- $\delta : Q \times (X \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}(Q \times \Gamma^{\leq 2})$ *is called a transition relation, and*
- $F \subseteq Q$ *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 $\mathcal{M} = (Q, X, \Gamma, q_{\mathsf{in}}, \delta, F)$ is said to be *deterministic* if $\mid \delta(q, a, \gamma) \mid + \mid \delta(q, \varepsilon, \gamma) \mid \leq 1$ for any $q \in Q$, $a \in X$ and $\gamma \in \Gamma$. By $\mid S \mid$, we denote the number of elements in a finite set $S$.

**Definition 2.** *A* configuration *of a pushdown automaton* $\mathcal{M}$ *is a pair* $(q, \gamma)$, *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,* $\varepsilon$*-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 [32].

**Definition 3.** *Let* $\alpha = a_1 a_2 \cdots a_n \cdots$ *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* $\alpha$, *starting from the initial configuration* $(q_{in}, \perp)$, *if and only if*

*(1)* $(q_0, \gamma_0) = (q_{in}, \perp)$, *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* $a_1 a_2 \cdots a_n \cdots = b_1 b_2 \cdots b_n \cdots$ *or* $b_1 b_2 \cdots b_n \cdots$ *is a prefix of* $a_1 a_2 \cdots a_n \cdots$.

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

Remark that a *run* is defined in line with [32], which does not require the pushdown automata to read through the whole tape. Such a condition differs from the ones mentioned in [12, 14], 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 $\omega$-languages for pushdown automata [32].

The acceptance conditions we treat in this study are as follows, as well as the $\omega$-languages defined by such conditions.

- Safety (or $\Pi_1$) acceptance condition.

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

- Reachability (or $\Sigma_1$) acceptance condition.

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

- Co-Büchi (or $\Sigma_2$) acceptance condition.

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

We also treat the following $\omega$-languages with the combinations of the above conditions.

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

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

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

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

- $\Delta_2$ acceptance condition. There exist $F_b$, $F_c \subseteq Q$,

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

Notice that infinite games recognized by nondeterministic pushdown automata bear some resemblance to those recognized by deterministic 2-stack visibly pushdown automata with the same acceptance conditions. 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.

**Definition 4.** *A 2-*stack visibly pushdown automaton *(2VPA) is a tuple* $\mathcal{M} = (Q, X, \Gamma, q_{in}, \delta, F)$, *where*

- $Q$ *is a finite set of states,*
- $X = \boldsymbol{Push}_1 \cup \boldsymbol{Pop}_1 \cup \boldsymbol{Push}_2 \cup \boldsymbol{Pop}_2 \cup \boldsymbol{Int}$ *is a finite input alphabet,*
- $\Gamma$ *is a finite stack alphabet, which contains a special bottom letter* $\perp$,
- $q_{in} \in Q$ *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*
  - ⋆ $\delta_{\text{Push}_1} \subseteq Q \times \boldsymbol{Push}_1 \times Q \times (\Gamma \setminus \{\perp\})$,
  - ⋆ $\delta_{\text{Pop}_1} \subseteq Q \times \boldsymbol{Pop}_1 \times \Gamma \times Q$,
  - ⋆ $\delta_{\text{Push}_2} \subseteq Q \times \boldsymbol{Push}_2 \times Q \times (\Gamma \setminus \{\perp\})$,
  - ⋆ $\delta_{\text{Pop}_2} \subseteq Q \times \boldsymbol{Pop}_2 \times \Gamma \times Q$,
  - ⋆ $\delta_{\text{Int}} \subseteq Q \times \boldsymbol{Int} \times Q$,
- $F \subseteq Q$ *is a set of final states.*

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

**Definition 5.** *Let* $\alpha = a_1 a_2 \cdots a_n \cdots$ *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* $\alpha$, *starting from the initial configuration* $(q_{in}, \perp, \perp)$, *if and only if*

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

We start with the determinacy of games recognized by deterministic 2-stack visibly pushdown automata, together with those recognized by nondeterministic ones. The classes of $\omega$-languages accepted by deterministic 2-stack visibly pushdown automata (2DVPA) with safety, reachabiliy, co-Büchi and Büchi conditions are denoted as follows in Table 1.

Table 1: The classes of $\omega$-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 $\omega$-languages, we would like first recall some results on deterministic and nondeterministic Turing machines. We follows the definition of Turing machines from [32], in which the machines are not required to finish reading the whole tape. By $\mathbf{TM}(\mathsf{C})$ (respectively, $\mathbf{DTM}(\mathsf{C})$), we denote the class of $\omega$-languages recognized by nondeterministic (respectively, deterministic) Turing machines with acceptance condition $\mathsf{C}$.

**Theorem 1** (cf. [32]).

$$\begin{array}{lcl} \mathbf{DTM}_\omega(\Pi_1) & = & \mathbf{TM}_\omega(\Pi_1) \;=\; \Pi^0_1 \\ \mathbf{DTM}_\omega(\Sigma_1) & = & \mathbf{TM}_\omega(\Sigma_1) \;=\; \Sigma^0_1 \\ \mathbf{DTM}_\omega(\Sigma_2) & = & \mathbf{TM}_\omega(\Sigma_2) \;=\; \Sigma^0_2 \\ \mathbf{DTM}_\omega(\Pi_2) & = & \Pi^0_2 \\ \mathbf{TM}_\omega(\Pi_2) & = & \Sigma^1_1 \end{array}$$

Note that the equalities of Theorem 1 also hold for the boldface versions. We here remark that boldface/lightface $\mathbf{2(D)VPL}_\omega$ and $\mathbf{PDL}_\omega$ (namely, the class of $\omega$-languages recognized by pushdown automata that we will treat later) are included in the corresponding boldface/lightface $\mathbf{(D)TM}_\omega$, 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 $\Pi^0_1$ (respectively, $\Sigma^0_1$, $\Sigma^0_2$, $\Pi^0_2$) 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 2.** *There exists an infinite game in* $\mathbf{2DVPL}_\omega(\Sigma_1 \wedge \Pi_1)$ *with only* $\Sigma^0_1$*-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 $\Sigma^0_1$-hard.

Similarly, we can prove

**Corollary 1.** *For any* $n$*, there exists an infinite game in* $\mathbf{2DVPL}_\omega(\mathcal{B}(\Sigma_1))$ *with only* $\Sigma^0_n$*-hard winning strategies.*

Theorem 2 (and Corollary 1) 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 $\omega$-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 $f : \omega \to \omega$ in the form of $1^{f(0)}01^{f(1)}01^{f(2)}\cdots$. Such an oracle is similar with that used in [21]. In the following, by $\mathbf{2DVPL}_\omega(\mathbf{C})$ for a boldface acceptance condition $\mathbf{C}$, we denote the boldface class of $\omega$-languages accepted by the corresponding deterministic 2-stack visibly pushdown automata with an oracle tape. For instance, by $\mathbf{2DVPL}_\omega(\boldsymbol{\Sigma}_1 \wedge \boldsymbol{\Pi}_1)$, we denote the boldface class of $\omega$-languages accepted by deterministic 2-stack visibly pushdown automata with a $\Sigma_1 \wedge \Pi_1$ acceptance condition and an oracle tape.

**Corollary 2.** *The determinacy of games in* $\mathbf{2DVPL}_\omega(\boldsymbol{\Sigma}_1 \wedge \boldsymbol{\Pi}_1)$ *implies* $\mathsf{ACA}_0$*. In fact, they are equivalent to each other over* $\mathsf{RCA}_0$*.*

In sequel, we show that the determinacy of infinite games whose winning sets are in $\mathbf{2DVPL}_\omega(\boldsymbol{\Sigma}_1)$ (respectively, $\mathbf{2DVPL}_\omega(\boldsymbol{\Delta}_2)$ $\mathbf{2DVPL}_\omega(\boldsymbol{\Sigma}_2)$) is equivalent to $\mathsf{WKL}_0$ (respectively, $\boldsymbol{\Delta}_1$-$\mathsf{Det}$ in $\omega^\omega$, $\mathsf{ATR}_0$).

Moreover we show that most of the above results for 2DVPA 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 $\omega$-languages accepted by the corresponding nondeterministic 2-stack visibly pushdown automata with an oracle tape.

**Theorem 3.** *For an acceptance condition* $\mathbf{C} \in \{\boldsymbol{\Sigma}_1, \boldsymbol{\Pi}_1, \boldsymbol{\Sigma}_1 \wedge \boldsymbol{\Pi}_1, \boldsymbol{\Delta}_2, \boldsymbol{\Sigma}_2\}$, $\mathsf{RCA}_0$ *proves*

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

Before we move on to treat other pushdown $\omega$-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.

Then we obtain the following analogous results for real-time (namley, no $\varepsilon$-transition is allowed) pushdown $\omega$-languages ($r$-$\mathbf{PDL}_\omega$) as stated in Theorem 4. It 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

**Theorem 4.** *The following diagram holds over* $\mathsf{RCA}_0$.

$$\begin{tikzcd}
r\text{-}\mathbf{PDL}_\omega(\boldsymbol{\Sigma}_2)\text{-}\mathsf{Det} \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] & \mathsf{ATR}_0 \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] & \mathbf{2DVPL}_\omega(\boldsymbol{\Sigma}_2)\text{-}\mathsf{Det} \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] & \mathbf{2DVPL}_\omega(\boldsymbol{\Pi}_2)\text{-}\mathsf{Det} \\
r\text{-}\mathbf{PDL}_\omega(\boldsymbol{\Delta}_2)\text{-}\mathsf{Det} \arrow[r, leftrightarrow] & \boldsymbol{\Delta}^0_1\text{-}\mathsf{Det} \arrow[r, leftrightarrow] \arrow[d] & \mathbf{2DVPL}_\omega(\boldsymbol{\Delta}_2)\text{-}\mathsf{Det} & \\
r\text{-}\mathbf{PDL}_\omega(\boldsymbol{\Sigma}_1 \wedge \boldsymbol{\Pi}_1)\text{-}\mathsf{Det} \arrow[r, leftrightarrow] & \mathsf{ACA}_0 \arrow[r, leftrightarrow] \arrow[d] & \mathbf{2DVPL}_\omega(\boldsymbol{\Sigma}_1 \wedge \boldsymbol{\Pi}_1)\text{-}\mathsf{Det} & \\
r\text{-}\mathbf{PDL}_\omega(\boldsymbol{\Sigma}_1)\text{-}\mathsf{Det} \arrow[r, leftrightarrow] & \mathsf{WKL}_0 \arrow[r, leftrightarrow] & \mathbf{2DVPL}_\omega(\boldsymbol{\Sigma}_1)\text{-}\mathsf{Det} \arrow[r, leftrightarrow] & \mathbf{2DVPL}_\omega(\boldsymbol{\Pi}_1)\text{-}\mathsf{Det}
\end{tikzcd}$$

Recall that Finkel [16] proved that the determinacy of $\mathbf{PDL}_\omega(\Pi_2)$ games is equivalent to the determinacy of effective analytic games.

In contract with the deterministic 2-stack visibly pushdown case, we have

**Theorem 5.** $\mathsf{RCA}_0 \vdash \mathbf{PDL}_\omega(\Pi_1)\text{-}\mathsf{Det}$.

We also show that all the real-times results in Theorem 4 also hold for the corresponding non-real-time pushdown $\omega$-langauges.

**Theorem 6.** *For an acceptance condition* $\mathbf{C} \in \{\mathbf{\Sigma}_1, \mathbf{\Sigma}_1 \wedge \mathbf{\Pi}_1, \mathbf{\Delta}_2, \mathbf{\Sigma}_2\}$, $\mathsf{RCA}_0$ *proves*

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

Finally, we conclude that all the arguments about pushdown automata in Part I 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 $\omega$-languages recognized by 1-counter automata (resp., real-time 1-counter automata) with boldface acceptance condition $\mathbf{C}$ is denoted as $\mathbf{CL}_\omega(\mathbf{C})$ (resp., $r$-$\mathbf{CL}_\omega(\mathbf{C})$). We show that

**Theorem 7.** *For an acceptance condition* $\mathbf{C} \in \{\mathbf{\Sigma}_1, \mathbf{\Sigma}_1 \wedge \mathbf{\Pi}_1, \mathbf{\Delta}_2, \mathbf{\Sigma}_2\}$, $\mathsf{RCA}_0$ *proves*

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

The results of Part I have been published in [24] and [25].

### Part II: Alternation hierarchy and fragments of modal $\mu$-calculus

Modal $\mu$-calculus, introduced by Kozen [22], is an extension of modal logic by adding greatest and least fixpoint operators. Such a logic is capable of capturing the greatest and least solutions of the equation $X = \Gamma(X)$, where $\Gamma$ is a monotone function with $X$ a set variable. Recall that modal logic is just the propositional logic with modalities $\Box$ (universal modality, which is interpreted as necessity) and $\Diamond$ (existential modality, which is interpreted as possibility).

From an automata-theoretic view, modal $\mu$-calculus is closely related with (alternating) tree automata. The equivalence between modal $\mu$-calculus and (alternating) parity tree automata over binary trees is established by Emerson and Jutla [13]. Study along this line is motivated by Rabin's investigations on the decidability of monadic second order logic with two successors [29], and highly concerned with the positional determinacy of parity games [13, 18].

A fundamental issue on modal $\mu$-calculus is the strictness of alternation hierarchy of modal $\mu$-calculus. The alternation hierarchy classifies the formulas by their alternation depth, that is, the number of alternating blocks of least and greatest fixpoint operators. Note that alternation depth, in a game-theoretic view, is related with the number of priorities in parity games, and from an automata-theoretic perspective, it concerns with the Rabin index of Rabin tree automata. The strictness of alternation hierarchy of modal

$\mu$-calculus was first established by Bradfield [5, 6, 7], and at the same time by Lenzi [23]. In sequel, Arnold [2] and Bradfield [4] showed that the alternation hierarchy of modal $\mu$-calculus is strict over infinite binary trees. Alberucci and Facchini [1] further proved that the alternation hierarchy is strict over reflexive transition systems.

In Part II, we first compare the three kinds of alternation hierarchy for modal $\mu$-calculus, namely, Niwiński, Emerson-Lei, and simple alternation hierarchy.

We start with the so-called simple (or syntactic) alternation hierarchy by counting simply syntactic alternation of $\mu$ and $\nu$ as follows, where the superscript $S$ means simple or syntactic.

**Definition 6.** *The simple alternation hierarchy of modal $\mu$-calculus is defined as follows.*

- $\Sigma_0^{S\mu}, \Pi_0^{S\mu}$*: the class of formulas with no fixpoint operators*
- $\Sigma_{n+1}^{S\mu}$ *: containing* $\Sigma_n^{S\mu} \cup \Pi_n^{S\mu}$ *and closed under the following operations*

  *(i) if* $\varphi_1, \varphi_2 \in \Sigma_{n+1}^{S\mu}$*, then* $\varphi_1 \vee \varphi_2$*,* $\varphi_1 \wedge \varphi_2$*,* $\Box_R \varphi_1$*,* $\Diamond_R \varphi_1 \in \Sigma_{n+1}^{S\mu}$*,*

  *(ii) if* $\varphi \in \Sigma_{n+1}^{S\mu}$*, then* $\mu X.\varphi \in \Sigma_{n+1}^{S\mu}$

- *dually for* $\Pi_{n+1}^{S\mu}$
- $\Delta_n^{S\mu} := \Sigma_n^{S\mu} \bigcap \Pi_n^{S\mu}$

*A formula is strict* $\Sigma_n^{S\mu}$ *if it is in* $\Sigma_n^{S\mu} - \Pi_n^{S\mu}$*.*

Notice that the above notion of simple alternation does not capture the complexity of dependence of fixpoints. A stronger notion is introduced as Emerson-Lei alternation hierarchy.

**Definition 7.** *The Emerson-Lei alternation hierarchy of modal $\mu$-calculus is defined as follows.*

- $\Sigma_0^{EL\mu}, \Pi_0^{EL\mu}$ *: the class of formulas with no fixpoint operators*
- $\Sigma_{n+1}^{EL\mu}$ *: containing* $\Sigma_n^{EL\mu} \cup \Pi_n^{EL\mu}$ *and closed under the following operations*

  *(i) if* $\varphi_1$*,* $\varphi_2 \in \Sigma_{n+1}^{EL\mu}$*, then* $\varphi_1 \vee \varphi_2$*,* $\varphi_1 \wedge \varphi_2$*,* $\Box_R \varphi_1$*,* $\Diamond_R \varphi_1 \in \Sigma_{n+1}^{EL\mu}$*,*

  *(ii) if* $\varphi \in \Sigma_{n+1}^{EL\mu}$*, then* $\mu Z.\varphi \in \Sigma_{n+1}^{EL\mu}$*, and*

  *(iii) if* $\varphi(X)$*,* $\psi \in \Sigma_{n+1}^{EL\mu}$ *and* $\psi$ *a closed formula (namely, no free variables), then* $\varphi(X \backslash \psi) \in \Sigma_{n+1}^{EL\mu}$*.*

- *dually for* $\Pi^{EL\mu}_{n+1}$
- $\Delta^{EL\mu}_{n} := \Sigma^{EL\mu}_{n} \bigcap \Pi^{EL\mu}_{n}$

Condition (iii) means that one can substitute a free variable $X$ of $\varphi \in \Sigma^{EL\mu}_{n+1}$ by a closed formula $\psi \in \Sigma^{EL\mu}_{n+1}$ such that the resulted formula $\varphi(X \backslash \psi)$ is still $\Sigma^{EL\mu}_{n+1}$.

Another stronger notion of alternation hierarchy is introduced by Niwiński.

**Definition 8.** *The Niwiński alternation hierarchy of modal $\mu$-calculus is defined as follows.*

- $\Sigma^{N\mu}_{0}, \Pi^{N\mu}_{0}$ : *the class of formulas with no fixpoint operators*
- $\Sigma^{N\mu}_{n+1}$ : *containing* $\Sigma^{N\mu}_{n} \cup \Pi^{N\mu}_{n}$ *and closed under the following operations*
  - *(i) if* $\varphi_1$, $\varphi_2 \in \Sigma^{N\mu}_{n+1}$, *then* $\varphi_1 \vee \varphi_2$, $\varphi_1 \wedge \varphi_2$, $\Box_R \varphi_1$, $\Diamond_R \varphi_1 \in \Sigma^{N\mu}_{n+1}$,
  - *(ii) if* $\varphi \in \Sigma^{N\mu}_{n+1}$, *then* $\mu Z.\varphi \in \Sigma^{N\mu}_{n+1}$, *and*
  - *(iii) if* $\varphi(X)$, $\psi \in \Sigma^{N\mu}_{n+1}$ *and no free variable of* $\psi$ *is captured by* $\varphi$, *then* $\varphi(X \backslash \psi) \in \Sigma^{N\mu}_{n+1}$.
- *dually for* $\Pi^{N\mu}_{n+1}$
- $\Delta^{N\mu}_{n} := \Sigma^{N\mu}_{n} \bigcap \Pi^{N\mu}_{n}$

*The Niwiński alternation depth of a formula* $\phi$ *is the least* $n$ *such that* $\phi \in \Delta^{N\mu}_{n+1}$.

**Fact 1.** $\Sigma^{S\mu}_{n} \subsetneq \Sigma^{EL\mu}_{n} \subsetneq \Sigma^{N\mu}_{n}$.

We also review the descriptive-set-theoretic and automata-theoretic arguments on strictness of such hierarchies. Moreover, the relations with arithmetic $\mu$-calculus, variable hierarchy, (alternating) parity tree automata and parity games are explained. We also introduce the transfinite extension of modal $\mu$-calculus.

Now we concentrate on the one-variable fragment of modal $\mu$-calculus. Apart from the alternation depths, the number of variables contained in a formula also serves as an important measure of complexity for formulas [20]. Thus in Part II, we define the alternation hierarchy of one-variable fragment of modal mu-calculus and weak alternation hierarchy. We prove that simple alternation hierarchy of one-variable fragment of modal $\mu$-calculus is strict, which is obtained by analyzing the correspondence with the counterparts of weak alternating tree automata and weak parity games.

In the following, we relax the condition that "all the fixpoint variables should be distinct". In such a relaxed context, a set variable can be bounded by $\mu$ and/or $\nu$ more than once. Then one-variable fragment of modal $\mu$-calculus, denoted by $L_\mu[1]$, consists of formulas each of which only contains one fixpoint variable, for instance, $\nu.X\Diamond_a(X \wedge \mu X.p \vee \Diamond_a X)$.

We can define the simple alternation hierarchy of $L_\mu[1]$ by modifying the definition of simple alternation hierarchy for $L_\mu$, via level-by-level restricting the formulas with only one fixpoint variable in Definition 6, for instance, $\Sigma_n^{S\mu}[1] = \Sigma_n^{S\mu} \bigcap L_\mu[1]$. By applying such restriction, we can get Emerson-Lei and Niwiński alternation hierarchy of $L_\mu[n]$, where $n$ denotes the number of fixpoint variables.

We first note that one-variable fragment of modal $\mu$-calculus is contained in the whole weak alternation hierarchy. By definition, it is obvious that the relation

$$\underbrace{\bigcup_{n<\omega} \Sigma_n^{S\mu}[1]}_{\text{Simple altern. hierar. of } L_\mu[1]} \subseteq \Delta_2^{N\mu}[2] \subseteq \Delta_2^{N\mu} = comp(\Sigma_1^{N\mu}, \Pi_1^{N\mu}) = \underbrace{\bigcup_{n<\omega} \Sigma_n^{W\mu}}_{\text{Weak altern. hierar.}}$$

holds over finitely branching transition systems.

Berwanger [3] showed that when we consider variable hierarchy, the formulas expressing the winning region of parity games, which exhaust the finite levels of alternation hierarchy of modal $\mu$-calculus, in fact, can be reduced to two-variable fragment of modal $\mu$-calculus. Similarly, our goal is to show the one variable is enough to express the winning region of weak games. Note that the formulas expressing the winning regions in weak games $G_n$ for $n < \omega$ already witness the strictness of the *weak alternation hierarchy.*

A *weak game* is given as a rooted structure $\mathcal{G}$, $v_0$ with $\mathcal{G} = (V, V_\Diamond, V_\Box, E, \Omega, n)$ and a priority function $\Omega : V_\Diamond \bigcup V_\Box \to \{1, 2, \ldots, n\}$, namely, $\Omega_i (= \{v : \Omega(v) = i\})$'s are a partition of $V_\Diamond \bigcup V_\Box$. Player I wins with a play $x$ if the priority sequence of $x$ is nonincreasing.

We follow the notion from [3]. Given $n$, we can construct the following formulas for $i = 1, \ldots, n$,

$$\begin{cases} \varphi_i := \nu X.\Big(\varphi_{i-1} \vee (\Omega_i \wedge \rhd X)\Big), & \text{if } i \text{ is odd} \\ \varphi_i := \mu X.\Big(\big(\varphi_{i-1} \vee \nu X.(\Omega_i \wedge \rhd X)\big) \vee (\Omega_i \wedge \rhd X)\Big), & \text{if } i \text{ is even} \end{cases}$$

where

$$\rhd X := (V_\Diamond \wedge \Diamond X) \vee (V_\Box \wedge \Box X)$$

express that player $\Diamond$ can ensure that from the current position, the set $X$ will be visited in one move.

The formula $\varphi_n$ describes that player $\Diamond$ has a winning strategy in a weak game with priority $n$.

**Example 1.** *For n=2,*

$$\varphi_1 = \nu X.(\Omega_1 \wedge \triangleright X)$$

$$\varphi_2 = \mu X.\left(\left(\nu X.(\Omega_1 \wedge \triangleright X) \vee \nu X.(\Omega_2 \wedge \triangleright X)\right) \vee (\Omega_2 \wedge \triangleright X)\right)$$

*Note that* $\varphi_2 \in \Sigma_2^{S\mu}[1]$.

We can see that the formula $\varphi_2$ expresses that the priority sequence for a play $\rho$ satisfies one of the following condition

$$\underbrace{22\ldots 22\ldots,}_{\text{always 2}} \qquad \underbrace{11\ldots 11\ldots,}_{\text{always 1}} \qquad \text{or} \qquad \underbrace{22\ldots 211\ldots 11\ldots}_{\text{evetually 1}}$$

which implies such $\rho$ is a winning play for player $\Diamond$ in a weak game with priority ranging over $\{1,2\}$.

Inductively, we can show that

**Theorem 8.** *The winning regions in weak games can be expressed by formulas of one-variable fragment of modal $\mu$-calculus.*

Recall that the formulas expressing the winning regions in weak games $G_n$ for $n < \omega$ witness the strictness of the weak alternation hierarchy. Then we have

**Theorem 9.** *The simple alternation hierarchy of* $L_\mu[1]$ *is strict over finitely branching transition systems. Moreover, the simple alternation hierarchy of* $L_\mu[1]$ *exhausts the weak alternation hierarchy.*

That is, $L_\mu[1]$ formulas are enough to express properties at any level of the weak alternation hierarchy.

# References

[1] L. Alberucci and A. Facchini, The modal $\mu$-calculus hierarchy over restricted classes of transition systems. *J. Symb. Log.* **74** (2009) 1367-1400.

[2] A. Arnold, The $\mu$-calculus alternation-depth hierarchy is strict on binary trees. *RAIRO-Theor. Inf. Appl.* **33** (1999), 329-339.

[3] D. Berwanger, Game logic is strong enough for parity games. *Studia Logica* **75** (2003), 205-219.

[4] J.C. Bradfield, Fixpoint alternation: arithmetic, transition systems and the binary tree. *RAIRO-Theor. Inf. Appl.* **33** (1999), 341-356.

[5] J.C. Bradfield, Simplifying the modal mu-calculus alternation hierarchy, in *Proceedings of 15th Annual Symposium on Theoretical Aspects of Computer Science, STACS 1998. Lecture Notes in Comput. Sci.* **1373** (1998), 39-49.

[6] J.C. Bradfield, The modal $\mu$-calculus hierarchy is strict. *Theoret. Comput. Sci.* **195** (1998), 133-153.

[7] J.C. Bradfield, The modal $\mu$-calculus hierarchy is strict, in *Proceedings of the 7th International Conference on Concurrency Theory, CONCUR 1996. Lecture Notes in Comput. Sci.* **1119** (1996), 233-246.

[8] J.R. Büchi and L.H. Landweber, Solving sequential conditions by finite-state strategies. *Trans. Amer. Math. Soc.* **138** (1969) 295-311.

[9] T. Cachat, Higher order pushdown automata, the Caucal hierarchy of graphs and parity games, in *Proc. ICALP' 03. International Colloquium on Automata, Languages, and Programming* Springer Berlin Heidelberg (2003) 556-569.

[10] T. Cachat, J. Duparc and W. Thomas, Solving pushdown games with a $\Sigma_3^0$ winning condition, in *Proc. of International Workshop on Computer Science Logic* Springer Berlin Heidelberg (2002) 322-336.

[11] A. Carayol, M. Hague, A. Meyer, C. H. Ong and O. Serre, Winning regions of higher-order pushdown games, in *Logic in Computer Science*, 2008. *Proc. LICS'08. 23rd Annual IEEE Symposium on. IEEE* (2008) 193-204.

[12] R.S. Cohen and A.Y. Gold, Theory of $\omega$-languages. Part I: Characterization of $\omega$-context-free languages. *J. Comput. System Sci.* **15** (1977) 169-184.

[13] E.A. Emerson and C.S. Jutla, Tree automata, mu-calculus and determinacy, in *Proc. FOCS' 91. 32nd Annual Symposium on. IEEE* IEEE Computer Society Press (1991) 368-377.

[14] J. Engelfriet and H.J. Hoogeboom, X-automata on $\omega$-words. *Theoret. Comput. Sci.* **110** (1993) 1-51.

[15] O. Finkel, Infinite games specified by 2-tape automata. *Ann. Pure Appl. Logic* **167** (2016) 1184-1212.

[16] O. Finkel, The determinacy of context-free games. *J. Symb. Log.* **78** (2013) 1115-1134.

[17] O. Finkel, On omega context free languages which are Borel sets of infinite rank. *Theoret. Comput. Sci.* **299** (2003) 327-346.

[18] Y. Gurevich and L. Harrington, Trees, automata, and games. in *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, STOC'82* (1982) 60-65.

[19] D.R. Hirschfeldt, *Slicing the truth: On the computable and reverse mathematics of combinatorial principles.* World Scientific, Lecture Notes Series, vol. 28, edited by Institute for Mathematical Sciences, National University of Singapore (2014).

[20] N. Immerman, J. F. Buss and D. A. M. Barrington, Number of variables is equivalent to space. *J. Symb. Log.* **66** (2001) 1217-1230.

[21] E. Jeandel, On immortal configurations in Turing machines, in *Conference on Computability in Europe* Springer Berlin Heidelberg, 2012, 334-343.

[22] D. Kozen, Results on the propositional $\mu$-calculus. *Theoret. Comput. Sci.* **27** (1983) 333-354.

[23] G. Lenzi, A hierarchy theorem for the $\mu$-calculus, in *Proceedings of the 25th International Colloquium on Automata, Languages and Programming, ICALP 1996.* **1099** *Lecture Notes in Comput. Sci.* (1996), 87-97.

[24] W. Li and K. Tanaka, The determinacy of pushdown $\omega$-languages. *RAIRO-Theor. Inf. Appl.* **51** (2017) 29-50.

[25] W. Li, S. Okisaka and K. Tanaka, Infinite games recognized by 2-stack visibly pushdown automata. *RIMS Kokyuroku* **1950** (2015) 121-137.

[26] 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.

[27] D.A. Martin, Borel determinacy. *Ann. of Math.* **102** (1975) 363-371.

[28] A. Montalbán and R.A. Shore, The limits of determinacy in second order arithmetic, in *Proc. of the London Mathematical Society* **104** (2012) 223-252.

[29] M.O. Rabin, Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. Soc.* **141** (1969) 1-35.

[30] O. Serre, Games with winning conditions of high Borel complexity, in *International Colloquium on Automata, Languages, and Programming* Springer Berlin Heidelberg (2004) 1150-1162. *Theoret. Comput. Sci.* **350** (2006) 345-372.

[31] S.G. Simpson, *Subsystems of second order arithmetic.* Perspectives in Logic. Cambridge University Press, Cambridge; Association for Symbolic Logic, Poughkeepsie, NY, second edition (2009).

[32] L. Staiger, $\omega$-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.

[33] J. R. Steel, *Determinateness and subsystems of analysis.* Ph.D. thesis. University of California, Berkeley (1977).

[34] K. Tanaka, *Descriptive set theory and subsystems of analysis.* Ph.D. thesis. University of California, Berkeley (1986).

[35] W. Thomas, Infinite games and verification (extended abstract of a tutorial). *Lecture Notes in Comput. Sci.* (2002) 58-64.

[36] I. Walukiewicz, Pushdown processes: games and model-checking. in *International Conference on Computer Aided Verification* Springer Berlin Heidelberg (1996) 62-74.

[37] I. Walukiewicz, Pushdown processes: games and model-checking. *Inform. and Comput.* **164** (2001) 234-263.