Variants of Infinite Games and Reverse Mathematics

Kazuyuki Tanaka

Mathematical Institute, Tohoku University

Nov. 18, 2014 (revised Jan. 8, 2015)
### The strength of determinacy

<table>
<thead>
<tr>
<th>Game</th>
<th>Space</th>
<th>Baire $(\omega^\omega)$</th>
<th>Cantor $(2^\omega)$</th>
<th>Turing invariant</th>
<th>REG$_\omega$</th>
<th>DCFL$_\omega$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Sigma^0_3$</td>
<td>$\Pi^1_3$-CA$_0$ ⊨ $\Pi^1_3$-CA$_0$ ⊨</td>
<td>$\text{ATR}_0$</td>
<td>×</td>
<td>$\text{RCA}_0$ ⊨</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$B(\Sigma^0_2)$</td>
<td>$\Pi^1_2$-CA$_0$ ⊨ $\Pi^1_2$-CA$_0$ ⊨</td>
<td>$\text{ACA}_0$ ⊨</td>
<td>$\text{RCA}_0$ ⊨</td>
<td>$\text{RCA}_0$ ⊨</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Sigma^0_2$</td>
<td>$\Sigma^1_1$-ID$_0$</td>
<td>$\text{ATR}_0$</td>
<td></td>
<td>$\text{RCA}_0$ ⊨</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Delta^0_2$</td>
<td>$\Pi^1_1$-TR$_0$</td>
<td>$\text{ATR}_0$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$B(\Sigma^0_1)$</td>
<td>$\Pi^1_1$-CA$_0$</td>
<td>$\text{ACA}_0$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Sigma^0_1$</td>
<td>$\text{ATR}_0$</td>
<td></td>
<td></td>
<td>$\text{WKL}_0$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Delta^0_1$</td>
<td>$\text{ATR}_0$</td>
<td></td>
<td></td>
<td>$\text{WKL}_0$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

By Steel, MedSalem, Nemoto, Welch, Montalbán, Shore, Büchi, Walukiewicz, etc.
Outline

1 Automata
   - Finite Automata
   - Pushdown Automata

2 Infinite Games
   - Gale-Stewart Games
   - Wadge Games
   - Determinacy and computable winning strategy
Outline

1 Automata
   - Finite Automata
   - Pushdown Automata

2 Infinite Games
   - Gale-Stewart Games
   - Wadge Games
   - Determinacy and computable winning strategy
Recall automata on finite words

A run on input $a_1...a_n$ is a finite sequence of states:

$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} q_n.$$ 

We say a finite word $a_1...a_n$ is accepted by this automaton if $q_n$ is in the given $F$ of final states. The language accepted by $M$ is denoted as $L(M)$. 
How about automata on infinite words

A run on input \(a_1...a_n...\) is an infinite sequence of states:

\[ q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} ... \xrightarrow{a_n} q_n... \]

We say an infinite word \(a_1...a_n...\) is accepted by this automaton with a Büchi condition if during this run a state in the given final set \(F\) appears infinitely many times. The language is denoted by \(L_\omega(\mathcal{M})\) or simply \(L(\mathcal{M})\).
REG and REG\(_\omega\)

**Definition**

A language \( L \subseteq A^{<\omega} \) is **regular** (REG) if there exists a finite automaton \( M \) such that \( L = L(M) \).

**Theorem**

\( \text{REG} = \text{deterministic REG} \) (i.e., each input produces a unique run).

**Definition**

A language \( L \subseteq A^\omega \) is an **\( \omega \)-regular language** (REG\(_\omega\)) if there exists a Büchi automaton \( M \) such that \( L = L_\omega(M) \).

**Theorem**

\( \text{REG}_\omega \supseteq \text{deterministic REG}_\omega \) with a Büchi condition.
Theorem (Büchi, 1960)

\[ \text{REG}_\omega = \omega-KC(\text{REG}), \]

i.e., \( L \in \text{REG}_\omega \) iff

\[ L = \bigcup_{i=1}^{n} U_i V_i^\omega, \]

where \( U_i, V_i \in \text{REG} \), for all \( 1 \leq i \leq n \) and \( n \in \mathbb{N} \).

Theorem

\( \text{REG}_\omega = \text{the deterministic } \omega\text{-regular languages with a “Muller” condition or a “Rabin” condition (i.e., a Boolean combination of Büchi conditions).} \)
Outline

1 Automata
   - Finite Automata
   - Pushdown Automata

2 Infinite Games
   - Gale-Stewart Games
   - Wadge Games
   - Determinacy and computable winning strategy
A **run** on input $a_1...a_n$ is a finite sequence of configurations such that:

$$
(q_{in}, \downarrow) \xrightarrow{a_1 \text{ or } \varepsilon} (q_1, \gamma_1) \cdots \xrightarrow{a_n \text{ or } \varepsilon} (q_s, \gamma_s) \text{ for } s \geq n.
$$

A finite word $a_1...a_n \in A^{<\omega}$ is **accepted** by an automaton $\mathcal{M}$ if there exists a run which ends with a state in $F$. 
What if the input is infinite

A run on $a_1...a_n...$ is an infinite sequence of configurations:

$$(q_{in}, \bot) \xrightarrow{a_1 \text{ or } \varepsilon} (q_1, \gamma_1)... \xrightarrow{a_n \text{ or } \varepsilon} (q_s, \gamma_s) \xrightarrow{a_{n+1} \text{ or } \varepsilon} ...$$

An infinite word $a_1...a_n... \in A^\omega$ is accepted by a Büchi pushdown automaton if there exists a run visiting a state in $F$ infinitely many times.
### Definition
A language $L \subseteq A^{<\omega}$ is **context-free** (CFL) if there exists a pushdown automaton $M$ such that $L = L(M)$.

### Theorem
**CFL** $\supseteq$ **DCFL** (deterministic CFL).

### Definition
A language $L \subseteq A^\omega$ is a **context-free $\omega$-language** (CFL$_\omega$) if there exists a Büchi pushdown automaton $M$ such that $L = L_\omega(M)$.

### Theorem
**CFL$_\omega$** $\supseteq$ **DCFL$_\omega$**.
Theorem (Linna, 1976; Cohen and Gold, 1977)

\[ \text{CFL}_\omega = \omega-KC(\text{CFL}), \]

i.e., \( L \in \text{CFL}_\omega \) iff

\[ L = \bigcup_{i=1}^{n} U_i V_i^\omega, \]

where \( U_i, V_i \in \text{CFL} \), for all \( 1 \leq i \leq n \) and \( n \in \mathbb{N} \).

Corollary

\[ \text{CFL}_\omega \supset \omega-KC(\text{DCFL}) \supset \text{DCFL}_\omega \]
Recall Borel hierarchy

**Definition**

For a non-null countable ordinal $\alpha$, the classes $\Sigma_\alpha^0$ and $\Pi_\alpha^0$ of the Borel hierarchy on the topological space $A^\omega$ are defined as follows:

- $\Sigma_1^0$ is the class of open sets of $A^\omega$.
- $\Pi_1^0$ is the class of closed sets of $A^\omega$.

And for any countable ordinal $\alpha \geq 2$:

- $\Sigma_\alpha^0$ is the class of countable unions of subsets of $A^\omega$ in $\bigcup_{\gamma < \alpha} \Pi_\gamma^0$.
- $\Pi_\alpha^0$ is the class of countable intersections of subsets of $A^\omega$ in $\bigcup_{\gamma < \alpha} \Sigma_\gamma^0$.

For $\alpha < \omega_1$, $\Delta_\alpha^0 = \Sigma_\alpha^0 \cap \Pi_\alpha^0$.

A set $L \subseteq A^\omega$ is Borel iff it is in the union $\bigcup_{\alpha < \omega_1} \Pi_\alpha^0 = \bigcup_{\alpha < \omega_1} \Sigma_\alpha^0$, where $\omega_1$ is the first uncountable ordinal.
Recall Borel hierarchy
Recall Wadge reducibility

**Definition (Wadge, 1983)**

For $X \subseteq A^\omega$ and $Y \subseteq B^\omega$, $X$ is said to be **Wadge reducible** to $Y$, denoted by $X \leq_W Y$, if and only if there exists a continuous function $f : A^\omega \rightarrow B^\omega$, such that $X = f^{-1}(Y)$.

**Definition**

A $\Pi^0_n$-subset $X \subseteq A^\omega$ is said to be a **$\Pi^0_n$-complete set**, if and only if

$$\text{for all } \Pi^0_n\text{-subset } Y \subseteq \Gamma^\omega, Y \leq_W X.$$  

**Example**

$(0^{<\omega}1)^\omega$ is a $\Pi^0_2$-complete set.
Outline

1. Automata
   - Finite Automata
   - Pushdown Automata

2. Infinite Games
   - Gale-Stewart Games
   - Wadge Games
   - Determinacy and computable winning strategy
Outline

1 Automata
   - Finite Automata
   - Pushdown Automata

2 Infinite Games
   - Gale-Stewart Games
   - Wadge Games
   - Determinacy and computable winning strategy
Gale-Stewart Games

Given $X \subseteq A^\omega$. The Gale-Stewart game $G(X)$:

<table>
<thead>
<tr>
<th>Player I</th>
<th>Player II</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1 \in A$</td>
<td>$b_1 \in A$</td>
</tr>
<tr>
<td>$a_2 \in A$</td>
<td>$b_2 \in A$</td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
</tr>
<tr>
<td>$x = a_1 b_1 a_2 b_2 \ldots$</td>
<td>$\ldots$</td>
</tr>
</tbody>
</table>

Player I wins iff $x \in X$. 
Definition

- **Player I** follows a strategy $f_I : (A^2)^{<\omega}$ in a play $a_1b_1a_2b_2...a_nb_n...$ if for each integer $n \geq 1$, $a_n = f_I(a_1b_1a_2b_2...a_{n-1}b_{n-1})$.
- If Player I wins every play in which he has followed the strategy $f_I$, then the strategy $f_I$ is called a **winning strategy**.

The winning strategy for Player II is defined in the same manner.
Definition

- The game $G(X)$ is said to be **determined** if one of the two players has a winning strategy.
- We shall denote $\text{Det}(C)$, where $C$ is a class of $\omega$-languages, to assert “Every Gale-Stewart game $G(X)$ is determined, where $X \subseteq A^\omega$ is an $\omega$-language in the class $C$”.
Outline

1 Automata
   - Finite Automata
   - Pushdown Automata

2 Infinite Games
   - Gale-Stewart Games
   - Wadge Games
   - Determinacy and computable winning strategy
Wadge games

Given $L \subseteq A^\omega$ and $L' \subseteq B^\omega$. Player I chooses elements from $A$. Player II chooses elements from $B \cup \{\varepsilon\}$. Note that $\varepsilon$ is an empty symbol and Player II selecting $\varepsilon$ means he skips in this round. The Wadge game $W(L, L')$:

Player I

\begin{align*}
a_1 & \in A \\
a_2 & \in A \\
\vdots & \\
\bar{a} & = a_1 a_2 \ldots \in A^\omega
\end{align*}

Player II

\begin{align*}
b_1 & \in B \cup \{\varepsilon\} \\
b_2 & \in B \cup \{\varepsilon\} \\
\vdots & \\
\bar{b} & = b_1 b_2 \ldots \in B^\omega
\end{align*}

Player II wins iff $\bar{a} \in L \iff \bar{b} \in L'$.

Note: Although Player II is allowed to skip, eventually Player II should produce an infinite word in $B^\omega$. 
Recalling the definition of Wadge reducibility, followings are two important theorems for Wadge games.

**Theorem**

Let $X \subseteq A^\omega$ and $Y \subseteq B^\omega$ be Borel sets. Then either $X \leq_W Y$ or $Y \leq_W X^c$.

**Theorem**

Player II has a winning strategy in $W(X, Y)$ iff $X \leq_W Y$.

**Theorem (Louveau and Saint-Raymond, 1988)**

Borel Wadge determinacy follows from open Gale-Stewart determinacy.
Outline

1. Automata
   - Finite Automata
   - Pushdown Automata

2. Infinite Games
   - Gale-Stewart Games
   - Wadge Games
   - Determinacy and computable winning strategy
Previous results on determinacy of Gale-Stewart games

Previous researches showed that Gale-Stewart games are determined over the following classes $C$’s.

<table>
<thead>
<tr>
<th>$C$</th>
<th>( \Sigma^0_1 )</th>
<th>( \Pi^0_2 )</th>
<th>( \Sigma^0_3 )</th>
<th>( \Pi^0_4 )</th>
<th>( \Delta^1_1 ) (Borel)</th>
<th>( \Sigma^1_1 )</th>
</tr>
</thead>
</table>
We are interested in the determinacy over the winning sets which are recognized by some machines.

<table>
<thead>
<tr>
<th>Languages</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{REG}_\omega$</td>
<td>$\subset B(\Sigma^0_2)$</td>
</tr>
<tr>
<td>$\text{DCFL}_\omega$</td>
<td>$\subset B(\Sigma^0_2)$</td>
</tr>
<tr>
<td>$\text{DTM}_\omega$ with Muller</td>
<td>$= B(\Sigma^0_2)$</td>
</tr>
<tr>
<td>$\text{CFL}_\omega$</td>
<td>$\subset \Sigma^1_1$</td>
</tr>
<tr>
<td>$\text{NTM}_\omega$ with Büchi</td>
<td>$= \Sigma^1_1$</td>
</tr>
</tbody>
</table>

**Theorem (Finkel, 2013)**

\[ \text{Det}(\text{CFL}_\omega) \iff \text{Det}(\Sigma^1_1). \]
Theorem

REG_\omega games are effectively determined with a computable winning strategy.

Theorem

DCFL_\omega games are effectively determined with a computable winning strategy.

To show the above theorem, recall that Walukiewicz (2001) and Serre (2003) proved

Theorem

The winning regions of a parity pushdown game (i.e., a special game on the infinite graph of configurations generated by a PDA) are regular.
The higher-order pushdown automata are generalization of pushdown automata, which are equipped with higher-order stacks. Higher-order stacks can be regarded as stacks of stacks structures.

Theorem (Carayol, Hague, Meyer, Ong, Serre, 2008)

The winning regions of a higher-order pushdown game are regular.

Theorem

\( \text{DHPDL}_\omega \) games are effectively determined with a computable winning strategy.
Finite chain of DPDA’s (Serre, 2006)

- Let $A_1, \ldots, A_n, A_{n+1}$ be deterministic real-time (i.e., no $\varepsilon$ transition) pushdown automata, and additionally $A_{n+1}$ be equipped with a parity condition. Moreover, the input alphabet of $A_{i+1}$ is the stack alphabet of $A_i$.

- Then an infinite word $\alpha_0$ (over the alphabet of $A_1$) belongs to $L(A_1 \Rightarrow \cdots \Rightarrow A_n \Rightarrow A_{n+1})$ iff there are $\alpha_1, \alpha_2, \ldots, \alpha_n$ such that
  - when $A_i$ reads $\alpha_{i-1}$, the sequence of its stack contents converges to $\alpha_i$ for all $i < n$,
  - finally, $A_{n+1}$ accepts $\alpha_n$. 

\[ \alpha_0 \xrightarrow{\text{copy}} A_1 \xrightarrow{} \alpha_1 \xrightarrow{\text{copy}} A_2 \xrightarrow{} \alpha_2 \xrightarrow{\text{copy}} \cdots \xrightarrow{\text{copy}} A_{n+1} \xrightarrow{} \alpha_n \]
We denote by $C_n(A)$ the class of languages $L(A_1 \triangleright \cdots \triangleright A_n \triangleright A_{n+1})$ on a finite alphabet $A$ for some deterministic real-time pushdown automata $A_1, \ldots, A_n, A_{n+1}$, where $A_{n+1}$ is equipped with a parity condition.
Theorem (Serre, 2006)

A language in $C_n(A)$ belongs to $B(\Sigma_{n+2})$. In addition, there is a language in $C_n(A)$, which is $\Pi_{n+2}$-complete.

Theorem (Serre, 2006)

$C_n(A)$ games are effectively determined with a computable winning strategy, which can be computed by higher-order pushdown automata.
Visibly Pushdown Automata (VPA)

- The alphabet $\Sigma$ is partitioned into **Push**, **Pop**, **Int**.
- The *transitions* are defined as follows.

![Diagram of Visibly Pushdown Automata](image-url)
2-Stack Visibly Pushdown Automata (2VPA)

- The alphabet $\Sigma$ is partitioned into $\text{Push}_1$, $\text{Pop}_1$, $\text{Push}_2$, $\text{Pop}_2$, $\text{Int}$. 

Example:

Given $\varepsilon = (f \ a \ g ; f \ c \ d ; d \ g ; f \ b \ g ; f \ x \ y ; y \ g ; \emptyset)$, the $\varepsilon$-language $f(ab)^n c d n \times n \times n; i^2 \mathbb{N}$ and $i \in n$ is recognized by a non-deterministic 2-stack visibly pushdown automaton (2VPA), but not 2DVPA. (Non-determinism is necessary to guess $i$.)

Details:
The non-deterministic 2VPA can, on $a$ and $b$, push $\#$ onto both stacks, and nondeterministically switch to a mode where it pushes $\$\$ onto both stacks. Intuitively this switch corresponds to the guess of what $i$ is. On popping, it checks that $c'$s get popped on $\$$ and $d'$s on $\#$, and $y'$s on $\$$ and $x'$s on $\#$. 

Kazuyuki Tanaka (Tohoku University)
2-Stack Visibly Pushdown Automata (2VPA)

- The *alphabet* $\Sigma$ is partitioned into $\text{Push}_1$, $\text{Pop}_1$, $\text{Push}_2$, $\text{Pop}_2$, $\text{Int}$.

**Example**

Given $\Sigma = (\{a\}, \{c, d\}, \{b\}, \{x, y\}, \emptyset)$, the $\omega$-language

$$\{(ab)^n c^i d^{n-i} x^i y^{n-i} \mid n, i \in \mathbb{N} \text{ and } i \leq n\}$$

is recognized by a non-deterministic 2-stack visibly pushdown automaton (2VPA), but not 2DVPA. *(non-determinism is necessary to guess $i$)*

**Details:**

Kazuyuki Tanaka (Tohoku University)  Vari. of Infinite Games and Reverse Math  Nov. 18, 2014  34 / 39
The alphabet $\Sigma$ is partitioned into $\text{Push}_1$, $\text{Pop}_1$, $\text{Push}_2$, $\text{Pop}_2$, $\text{Int}$.

Example

Given $\Sigma = (\{a\}, \{c, d\}, \{b\}, \{x, y\}, \emptyset)$, the $\omega$-language

$$\{(ab)^n c^i d^{n-i} x^i y^{n-i} | n, i \in \mathbb{N} \text{ and } i \leq n\}$$

is recognized by a non-deterministic 2-stack visibly pushdown automaton (2VPA), but not 2DVPA. (non-determinism is necessary to guess $i$ )

Details: The non-deterministic 2VPA can, on $a$ and $b$, push $\#$ onto both stacks, and nondeterministically switch to a mode where it pushes $\$$ onto both stacks. Intuitively this switch corresponds to the guess of what $i$ is. On popping, it checks that $c$'s get popped on $\$$ and $d$'s on $\#$, and $y$'s on $\$$ and $x$'s on $\#$. 
Finite Languages

- **VPL**: the class of finite languages recognized by visibly pushdown automata.
- **DVPL**: the class of finite languages recognized by deterministic visibly pushdown automata.
- **CSL**: the class of context-sensitive languages.
Theorem (W.Li and K.T.)

There exists a deterministic 2-visibly pushdown automaton $\mathcal{M}$ such that

1. the $\omega$-language $L(\mathcal{M})$ is defined by an accepting condition (a Boolean combination of $\Sigma_1^0$-sets), and

2. Player II has a winning strategy in the game $G(L(\mathcal{M}))$, but has no computable winning strategies in this game.
Theorem (W.Li and K.T.)

There exists a deterministic 2-visibly pushdown automaton \( M \) such that

1. the \( \omega \)-language \( L(M) \) is defined by an accepting condition (a Boolean combination of \( \Sigma^0_1 \)-sets), and
2. Player II has a winning strategy in the game \( G(L(M)) \), but has no computable winning strategies in this game.

Theorem (Reverse Mathematics)

The determinacy of \( B(\Sigma^0_1) \) games defined by deterministic 2-visibly pushdown automata is equivalent to \( \text{ACA}_0 \).
Decidability of Computable Games

Question:

\[ \text{Det}(2\text{VPL}_\omega) \iff \text{Det}(\Sigma^1_1) \]

\[ 2\text{VPL}_\omega \cup 2\text{DVPL}_\omega \]

\[ \text{Undecidable} \]

\[ \Sigma^1_1 = \text{BTM}_\omega = \text{BCL}_\omega(2) \]

Note:

\[ \text{Det}(\text{CFL}_\omega) \iff \text{Det}(\Sigma^1_1) \]

\[ \text{DCFL}_\omega \subset \text{DHPDL}_\omega \quad \text{Decidable} \]

\[ \text{DVPL}_\omega \subset \text{VPL}_\omega \]

\[ \text{REG}_\omega \]

\[ \cup \]

\[ \cup \]
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, LICS’08, 23rd Annual IEEE Symposium on IEEE, 193-204.

O. Finkel,
The determinacy of context-free games.

O. Serre,
Games with winning conditions of high Borel complexity.

I. Walukiewicz,
Pushdown processes: Games and model-checking.
Congratulations on Prof. Fuchino’s Sixtieth Birthday!