Determinacy strength of infinite games in ω-languages recognized by variations of pushdown automata

Wenjuan Li
jointly with Prof. Kazuyuki Tanaka

Mathematical Institute, Tohoku University

Sep. 16, 2016

Workshop on Mathematical Logic and its Applications
JSPS Core-to-Core Program
Kyoto University, Kyoto
Infinite games
- Gale-Stewart game $\mathcal{G}(X)$, where $X(\subseteq A^\omega)$ is a winning set for player I

Determinacy
- With the usual convention, $\mathcal{C}$-Det denotes that “A Gale-Stewart game $\mathcal{G}(X)$ is determined (one of the two players has a winning strategy), if $X$ is contained in the class $\mathcal{C}$”.

$\omega$-languages accepted by automata
- $L(\mathcal{M})$, where $\mathcal{M}$ is some kind of automata

**Question**
If the winning sets are effectively given, i.e., winning sets are accepted by some kind of automata, how is the determinacy strength of such games?
Outline

1. Introduction
   - Pushdown automata, visibly pushdown automata, etc.

2. Determinacy strength and $\omega$-languages
   - $2\text{DVPL}_\omega$, $r\text{-PDL}_\omega$, $\text{PDL}_\omega$, etc.

3. Ongoing and future works
Pushdown automata on infinite words ($\omega$-PDA)

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

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

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.
For (1-stack) visibly pushdown automata (2VPA), the alphabet \( A \) is partitioned into \textbf{Push}, \textbf{Pop}, \textbf{Int}. The \textit{transitions} are as follows.

- If \( a \in \text{Pop} \):
  - From \( p \) to \( q \): \( a \) with \( b \):
  - From \( q \) to \( b \):
- If \( a \in \text{Push} \):
  - From \( q \) to \( c \):
  - From \( c \) to \( a \):
- If \( a \in \text{Int} \):

For 2-stack visibly pushdown automata (2VPA), the alphabet \( A \) is partitioned into \textbf{Push}_1, \textbf{Pop}_1, \textbf{Push}_2, \textbf{Pop}_2, \textbf{Int}.

Example

Given \( A = (\{a\}, \{\bar{a}\}, \{b\}, \{\bar{b}\}, \emptyset) \), the language \( \{(ab)^n\bar{a}^n\bar{b}^n \mid n \in \mathbb{N}\} \), is recognized by a deterministic 2-stack visibly pushdown automaton (2DVPA).
Outline

1 Introduction
   - Pushdown automata, visibly pushdown automata, etc.

2 Determinacy strength and $\omega$-languages
   - $2DVPL_\omega$, $r$-$PDL_\omega$, $PDL_\omega$, etc.

3 Ongoing and future works
Determinacy strength of infinite games in deterministic 2-stack visibly $\omega$-languages
Recall undecidability results of games in some $\omega$-languages.

$\Sigma_1^1 = \text{BTM}_\omega = \text{BCL}_\omega$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$

$\bigcup$
Question

How about other acceptance conditions of lower levels?

In this talk

we concentrates on determinacy strength of infinite games specified by nondeterministic pushdown automata and variants of it with various acceptance conditions, e.g., safety, reachability, co-Büchi conditions.
Acceptance conditions of infinite words

<table>
<thead>
<tr>
<th>Condition</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safety (or $\Pi_1$) acceptance condition</td>
<td>$L(M) = { \alpha \in \Sigma^\omega \mid \text{there is a run } r = (q_i, \gamma_i)_{i \geq 1} \text{ of } M \text{ on } \alpha \text{ such that } \forall i, q_i \in F }$</td>
</tr>
<tr>
<td>Reachability (or $\Sigma_1$) acceptance condition</td>
<td>$L(M) = { \alpha \in \Sigma^\omega \mid \text{there is a run } r = (q_i, \gamma_i)_{i \geq 1} \text{ of } M \text{ on } \alpha \text{ such that } \exists i, q_i \in F }$</td>
</tr>
</tbody>
</table>

Let $\text{Inf}(r)$ be the set of states that are visited infinite many times during the run $r$.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-Büchi (or $\Sigma_2$) acceptance condition</td>
<td>$L(M) = { \alpha \in \Sigma^\omega \mid \text{there is a run } r = (q_i, \gamma_i)_{i \geq 1} \text{ of } M \text{ on } \alpha \text{ such that } \text{Inf}(r) \subseteq F }$</td>
</tr>
</tbody>
</table>
Acceptance conditions of infinite words (continued)

\((\Sigma_1 \land \Pi_1)\) acceptance condition

There exist \(F_r, F_s \subseteq Q\),

\[ L(M) = \{ \alpha \in \Sigma^\omega \mid \text{there is a run } r = (q_i, \gamma_i)_{i \geq 1} \text{ of } M \text{ on } \alpha \]

\[ \text{such that } \exists i, q_i \in F_r \land \forall i, q_i \in F_s \}\].

\((\Sigma_1 \lor \Pi_1)\) acceptance condition

There exist \(F_r, F_s \subseteq Q\),

\[ L(M) = \{ \alpha \in \Sigma^\omega \mid \text{there is a run } r = (q_i, \gamma_i)_{i \geq 1} \text{ of } M \text{ on } \alpha \]

\[ \text{such that } \exists i, q_i \in F_r \lor \forall i, q_i \in F_s \}\].
Acceptance conditions of infinite words (continued)

\( \Delta_2 \) acceptance condition

There exist \( F_b, F_c \subset Q \),

\[
L(M) = \{ \alpha \in \Sigma^\omega | \text{there is a run } r \text{ of } M \text{ on } \alpha \text{ such that } \text{Inf}(r) \cap F_b \neq \emptyset \}
\]

\[
= \{ \alpha \in \Sigma^\omega | \text{there is a run } r \text{ of } M \text{ on } \alpha \text{ such that } \text{Inf}(r) \subset F_c \}.
\]
Various acceptance conditions of $\omega$-2DVPA

- We denote the $\omega$-languages accepted by $\omega$-2DVPA with different acceptance conditions as follows.

- $\omega$-languages accepted by deterministic Turing machines with safety (resp., reachability, co-Büchi, Büchi) condition is the collection of all arithmetical $\Pi_1^0$-sets (respectively, $\Sigma_1^0$-sets, $\Sigma_2^0$-sets, $\Pi_2^0$-sets).

<table>
<thead>
<tr>
<th>Acceptance conditions</th>
<th>Subclass of $2DVPL_\omega$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reachability</td>
<td>$2DVPL_\omega(\Sigma_1)$ $\subseteq \Sigma_1^0$</td>
</tr>
<tr>
<td>Safety</td>
<td>$2DVPL_\omega(\Pi_1)$ $\subseteq \Pi_1^0$</td>
</tr>
<tr>
<td>Co-Büchi</td>
<td>$2DVPL_\omega(\Sigma_2)$ $\subseteq \Sigma_2^0$</td>
</tr>
<tr>
<td>Büchi</td>
<td>$2DVPL_\omega(\Pi_2)$ $\subseteq \Pi_2^0$</td>
</tr>
</tbody>
</table>

Similarly, by $2DVPL_\omega(C)$ we denote the $\omega$-languages accepted by deterministic 2-stack visibly pushdown automata with an acceptance condition $C$. 
Theorem

There exists an infinite game in $2\text{DVPL}_\omega(\Sigma_1 \land \Pi_1)$ with only $\Sigma_1^0$-hard winning strategies.

Proof.

- Let $\mathcal{R}$ be a universal 2-counter automaton.
- We construct a game $G_\mathcal{R}$ such that the halting problem of $\mathcal{R}$ is computable in any winning strategies of player II, while player I has no winning strategy, and moreover the winning set for player II is accepted by a deterministic 2-stack visibly pushdown automaton with a $\Sigma_1 \lor \Pi_1$ acceptance condition.
A 2-counter automaton can be seen as a restricted 2-stack pushdown automaton with just one symbol for each stack: the number of the symbols in a stack is expresses as a nonnegative integer in a counter.

The input is a natural number $m$ which is initially store in one of the counter.

By the current state and the tests results on whether each counter is zero or not, the automaton goes to 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.

It is known that a (deterministic) 2-counter automaton, is equivalent to a Turing machine. Thus the halting problem for a certain (universal deterministic) 2-counter automaton is $\Sigma^0_1$-complete.
Recall 2-counter automata (continued)

- A **configuration** \((q, m, n)\) of a 2-counter automaton \(\mathcal{R}\) is coded as \(qa^mb^n\), where \(q \in Q\), and \(m, n\) are non-negative integers in the two counters.

- A **run** for a natural number \(m\) on \(\mathcal{R}\):
  \[q_{\text{in}}a^mb^n \xrightarrow{} \mathcal{R} q_1a^{m_1}b^{n_1} \xrightarrow{} \mathcal{R} q_2a^{m_2}b^{n_2} \xrightarrow{} \mathcal{R} \cdots\], where \(q_{\text{in}}\) is the initial state, and \(m_0 = m, n_0 = 0\).

- A run is **halting** if it reaches a halting configuration.

- A natural number \(m \in L(\mathcal{R})\) iff there exists a run on \(m\) such that
  \[q_{\text{in}}a^mb^n \xrightarrow{} \mathcal{R} q_1a^{m_1}b^{n_1} \xrightarrow{} \mathcal{R} \cdots \xrightarrow{} \mathcal{R} q_sa^msb^ns\], where \(n_0 = 0\) and \(q_s\) is a halting state.
Back to the proof: construct a game $G_{\mathcal{R}}$

Let $\mathcal{R}$ be a universal 2-counter automaton.

**I**: $m_0 \in L(\mathcal{R})$?

**II**: Yes and writes a finite run of $\mathcal{R}$ on $m_0$.

**I**: challenges and finds an error.

**II**: challenges and finds no error.

**I** wins

**II** wins

**II** wins

**II** wins

**I** wins
Construct a game $G_{\mathcal{R}}$

I: $m_0 \in L(\mathcal{R})$?

II: Yes and writes a finite run of $\mathcal{R}$ on $m_0$.

II: No.

I: defends by writing a finite run of $\mathcal{R}$ on $m_0$.

I: challenges and finds no error.

II: challenges and finds an error.

II: challenges and finds no error.

II: never challenges.

I: never challenges.

II: I wins

I: II wins

II: II wins

II: II wins

I: II wins

II: I wins

II: I wins
If player II says “no”, how does she challenge?

Player II wants to make sure

1. the sequence of configurations provided by player I is a sequence of the form $qa^mb^n$ and connected by $\triangleright$,

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 conditions (1), (2) and (4) are easy to check with $\Sigma_1$ conditions (i.e., player I lose with $\Pi_1$). In the following we explain how player II challenges if she thinks player I cheated by disobeying the above rule (3).

Such a play can be checked by a deterministic 2-stack visibly pushdown automaton.
Assume player II has a winning strategy $\sigma$, then

$$L(\mathcal{R}) = \{ m : \text{player II follows her winning strategy } \sigma \text{ and answers “yes” to } m \text{ in the game } \mathcal{G}_\mathcal{R} \text{ in } 2DVPL_\omega(\Sigma_1 \land \Pi_1) \}.$$ 

Since the halting problem of $\mathcal{R}$ is $\Sigma^0_1$-complete, any winning strategy for player II is $\Sigma^0_1$-hard.
Corollary

The determinacy of games in $2\text{DVPL}_\omega(\Sigma_1 \land \Pi_1)$ with an oracle implies $\text{ACA}_0$. In fact, they are equivalent to each other over $\text{RCA}_0$.

Sketch.

1. We use a 2-counter automaton $\mathcal{R}$ with an oracle function $f : \mathbb{N} \to \mathbb{N}$, denoted as $\mathcal{R}^f$. $m_0 \in L(\mathcal{R}^f)$ iff there exists a run on $m$ s.t.

$$q_0 a^{m_0} b^{n_0} \triangleright q_1 a^{m_1} b^{n_1} \triangleright \cdots \triangleright q_s a^{m_s} b^{n_s},$$

where $n_0 = f(m_0)$ and $q_s$ is halting.

2. The game $\mathcal{G}_{\mathcal{R}^f}$

\[ \begin{array}{ll}
\text{I} & m_0 \in L(\mathcal{R})? \quad q_0 a^{m_0} b^{f(m_0)} \triangleright \cdots \triangleright q_i a^{m_i} b^{n_i} \triangleright q_{i+1} a^{m_{i+1}} b \\
\text{II} & \text{Yes/No} \quad \text{Challenge with a witness:} \quad \check{c}b(a\check{b})^{m_{i+1}} \check{q} \triangleleft b^{n_i}(a\check{b})^{\min\{m_i, m_{i+1}\}} \check{q}
\end{array} \]

3. Such a play can be checked by a deterministic 2-stack visibly pushdown automaton with an oracle tape, in which the oracle tape is read-only, non-real-time and in the form $1^{f(0)}01^{f(1)}01^{f(2)}\ldots$. 

Check!
Corollary

For any \( n \), there exists an infinite game in \( 2DVPL_\omega(\mathcal{B}(\Sigma_1)) \) with only \( \Sigma^0_n \)-hard winning strategies.

The brief idea is as follows. Take the case \( n = 3 \) as an example. Let \( A \) be any \( \Sigma^0_3 \) set. Then there is a 2-counter automaton \( \mathcal{R} \) such that \( m_0 \in A \) if and only if

\[
\exists m_1 \forall m_2 (\mathcal{R} \text{ halts on } m = 2^{m_0} 3^{m_1} 5^{m_2}).
\]

We can construct game starts with player I by asking \( m_0 \) in \( A \) or not.
Theorem

The determinacy of games in $2DVPL_\omega(\Pi_1)$ implies $\Sigma^0_1$-SP.

Proof.

- Let $R_1$ and $R_2$ be two 2-counter automata such that $L(R_1) \cap L(R_2) = \emptyset$.
- Player I has a sequence of $m$’s, and for each $m$, he chooses $i$ such that $R_i$ does not halt with $m$.
- Player II may challenge player I’s choice $i$ at any $m$.
Then player I defends by producing an infinite sequence of configurations of $R_i$ on $m$, $q_0a_0^m b_0^n \triangleright q_1a_1^m b_1^n \cdots$, where $m_0 = m$ and $n_0 = 0$.

Player I defends by providing a sequence of configurations of $R_i$ on $m$.

Player II challenges at this $i$.

$1221\cdots q_0a_0^m b_0^n \triangleright q_1a_1^m b_1^n \triangleright \cdots \triangleright q_m a_m^s c \cdots$

Player II may also challenge at a transition if she thinks player I has cheated.

While player I is producing a sequence of configurations, player II may challenge at any point she thinks player cheated.

Player I’s winning set is accepted by a 2DVPA($\Pi_1$). Assume that player I has a winning strategy $\sigma$, then the desired separating set is $X = \{ m : \text{player I follows } \sigma \text{ and picks } R_2 \text{ for } m \}$.

Corollary

The determinacy of the games in $2DVPL_\omega(\Pi_1)$ with an oracle is equivalent to $WKL_0$ over $RCA_0$. 
Theorem

The determinacy of $2DVPL_\omega(\Delta_2)$ implies the determinacy of $\Delta^0_1$ games in $\omega^\omega$, which is equivalent to $ATR_0$.

- We mimic the proof of $\Delta^0_2$-Det (in $2^\omega$) $\rightarrow$ $\Delta^0_1$-Det (in $\omega^\omega$) in [NMT07] $^1$.
- By using their coding technique, we write $\tilde{\alpha} \in \omega^\omega$ for the unique sequence coded by $\alpha \in 2^\omega$. Note that not all sequences in $2^\omega$ code a sequence in $\omega^\omega$.

---

Then, a play $\tilde{\alpha}$ in $\Delta^0_1$ game in $\omega^\omega$ can be translated into a play $\alpha$ in $2^\omega$ and $\alpha$ is winning for player 0 (resp. player 1) iff

(a) $\tilde{\alpha}$ is a winning play (resp. $\tilde{\alpha}$ is not a winning play) in the $\Delta^0_1$ game in $\omega^\omega$ while both players obey the rules to produce a play $\alpha$, or

(b) while they are producing $\alpha$, player 1 (resp. player 0) breaks the rules. [a $\Sigma^0_2$ statement]

which constitutes a $\Sigma^0_2$ winning set for player 0 (resp. player 1). Thus the game is $\Delta^0_2$ in $2^\omega$.

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 $\Delta^0_1$ game in $\omega^\omega$ to a $2\text{DVPL}_\omega(\Delta_2)$ using the coding rules given by the above $\Delta^0_2$ game in $2^\omega$.

- The coding rule does not need any modification for $2\text{DVPL}_\omega(\Delta_2)$. So, for simplicity, the players are assumed to obey this rule and we just treat the above case (a).

- Given a $\Delta^0_1$ game, there exist two 2-counter automata $R_0$ and $R_1$ such that

$$s \text{ is a winning play } \iff \exists n \#s[n] \in L(R_0) \iff \neg \exists n \#s[n] \in L(R_1),$$

where $\#s[n]$ denotes a code of the initial $n$-segment of $s$. 
We construct a game $G_{R_1,R_2}$ as follows.

- When a player produces a finite sequence $\alpha[n]$ in the $\Delta^0_2$ game in $2^\omega$ such that $\#\tilde{\alpha}[n] \in L(R_i) (i \in \{0, 1\})$, player $i$ in the game $G_{R_0,R_1}$ starts providing a sequence of configurations of $R_i$ on $\#\tilde{\alpha}[n]$, which player $i$ claims to halt in finite steps.

- While player $i$ is making such a sequence of configuration of $R_i$ on $\#\tilde{\alpha}[n]$, the player $1 - i$ may challenge at any point.

- We can see that the winning set for player 0 in the constructed game $G_{R_1,R_2}$ is in $2\mathbf{DVPL}_\omega(\Delta_2)$. Moreover, if player $i$ has a winning strategy in $2\mathbf{DVPL}_\omega(\Delta_2)$, then player $i$ also has a winning strategy in the original $\Delta^0_1$ game in $\omega^\omega$. 

Corollary

The determinacy of games $2\text{DVPL}_\omega(\Sigma_2)$ is equivalent to $\text{ATR}_0$ over $\text{RCA}_0$.

Proof.

By the above theorem,

$$2\text{DVPL}_\omega(\Sigma_2)-\text{Det} \rightarrow 2\text{DVPL}_\omega(\Delta_2)-\text{Det} \rightarrow \text{ATR}_0.$$  

By [NMT07],

$$\text{ATR}_0 \rightarrow \Sigma^0_2-\text{Det} \text{ in } 2^\omega \rightarrow 2\text{DVPL}_\omega(\Sigma_2)-\text{Det}.$$
Determinacy strength of infinite games in pushdown $\omega$-languages
We treat \( \omega \)-languages accepted by nondeterministic \( \omega \)-PDA with different acceptance conditions.

<table>
<thead>
<tr>
<th>Accepting conditions</th>
<th>Subclass of ((r-)PDL_\omega^2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reachability</td>
<td>((r-)PDL_\omega(\Sigma_1))</td>
</tr>
<tr>
<td>Safety</td>
<td>((r-)PDL_\omega(\Pi_1))</td>
</tr>
<tr>
<td>Co-Büchi</td>
<td>((r-)PDL_\omega(\Sigma_2))</td>
</tr>
<tr>
<td>Büchi</td>
<td>((r-)PDL_\omega(\Pi_2))</td>
</tr>
</tbody>
</table>

Recall:

- \(PDL_\omega(\Pi_2) = CFL_\omega\).
- \(DPDL_\omega(\Pi_2) \subsetneq DPDL_\omega(B(\Sigma_2)) = DCFL_\omega\).
- \(PDL_\omega(\Pi_1)\) (respectively, \(PDL_\omega(\Sigma_1)\), \(PDL_\omega(\Sigma_2)\), \(PDL_\omega(\Pi_2)\)) is a subclass of arithmetical \(\Pi^0_1\) (respectively, \(\Sigma^0_1\), \(\Sigma^0_2\), \(\Sigma^1_1\)) class.

\(^2\)The symbol \(r\) denotes the real-time case.
Intuitively...

- A play in an infinite game in $2\text{DVPL}_\omega$:

$$q_0a^{m_0}b^{n_0} \triangleright \cdots \triangleright q_i a^{m_i}b^{n_i} \triangleright q_{i+1} a^{m_{i+1}} b \triangleright b(\bar{a}s)^{m_{i+1}} q \triangleright b^{n_i}(\bar{a}s)^{\min\{m_i, m_{i+1}\}\bar{a}} \bar{q} \cdots$$

To be compared by two stack, which is provoked by a challenge of the other player in the game.

- A play in an infinite game in $\text{PDL}_\omega$:

$$q_0a^{m_0}b^{n_0} \triangleright \cdots \triangleright q_i a^{m_i}b^{n_i} \triangleright q_{i+1} a^{m_{i+1}} b \cdots$$

Has been compared due to the nondeterminism of the pushdown automata.
Theorem

The determinacy of games in \( r\text{-PDL}_\omega(\Sigma_1) \) implies \( \Sigma^0_1\text{-SP} \).

Sketch.

Instead of checking by a challenge of player II, a pushdown automaton itself can nondeterministically check whether player I makes a mistake or not.

Remark

Safety condition is not the complement of reachability condition for nondeterministic pushdown languages.

Corollary

The determinacy of the games in \( r\text{-PDL}_\omega(\Sigma_1) \) with an oracle is equivalent to \( \text{WKL}_0 \) over \( \text{RCA}_0 \).
Theorem

\[ \text{RCA}_0 \vdash \text{PDL}_\omega(\Pi_1)-\text{Det}. \]

Proof idea

- Assume a pushdown automaton \( M \) with a \( \Pi_1 \) acceptance condition. We can construct a pushdown game \( G_P \) such that
  - if there exists a computable winning strategy \( \sigma \) for player \( i \) in \( G(L(M)) \), then there exists a winning strategy \( \sigma' \) for player \( i \) in \( G_P \) which is computable from \( \sigma \), and vice versa.

- From Walukiewicz (1996, 2001), we can show that there is a winning strategy in \( G_P \) and it is computable.

Note that a pushdown game is played on an infinite graph, which is generated by a pushdown process.
Theorem

\[ \begin{align*}
\text{r-PDL}_\omega(\Sigma_2)-\text{Det} & \iff \text{ATR}_0 \implies 2\text{DVPL}_\omega(\Sigma_2)-\text{Det} \iff 2\text{DVPL}_\omega(\Pi_2)-\text{Det} \\
\downarrow & \\
\text{r-PDL}_\omega(\Delta_2)-\text{Det} & \implies \Delta_1^0-\text{Det} \iff 2\text{DVPL}_\omega(\Delta_2)-\text{Det} \\
\downarrow & \\
\text{r-PDL}_\omega(\Sigma_1 \land \Pi_1)-\text{Det} & \iff \text{ACA}_0 \iff 2\text{DVPL}_\omega(\Sigma_1 \land \Pi_1)-\text{Det} \\
\downarrow & \\
\text{r-PDL}_\omega(\Sigma_1)-\text{Det} & \iff \text{WKL}_0 \iff 2\text{DVPL}_\omega(\Sigma_1)-\text{Det} \iff 2\text{DVPL}_\omega(\Pi_1)-\text{Det}
\end{align*} \]

Corollary

For an acceptance condition \( \mathcal{C} \in \{\Sigma_1, \Sigma_1 \land \Pi_1, \Delta_2, \Sigma_2\} \),

\[ \text{r-PDL}_\omega(\mathcal{C})-\text{Det} \iff \text{PDL}_\omega(\mathcal{C})-\text{Det}. \]
We can easily observe that all the arguments about pushdown automata, 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.

**Theorem**

For an acceptance condition \( C \in \{ \Sigma_1, \Sigma_1 \land \Pi_1, \Delta_2, \Sigma_2 \} \),

\[
\text{r-CL}_\omega(C)\text{-Det} \iff \text{CL}_\omega(C)\text{-Det} \iff \text{PDL}_\omega(C)\text{-Det}.
\]
Ongoing and future works

- Study the quantitative analysis of concurrent games in pushdown \( \omega \)-languages.

- Investigate the determinacy strength of the Blackwell-type games in \( \omega \)-languages, and its relation to probabilistic automata and other stochastic systems.


Thank you!