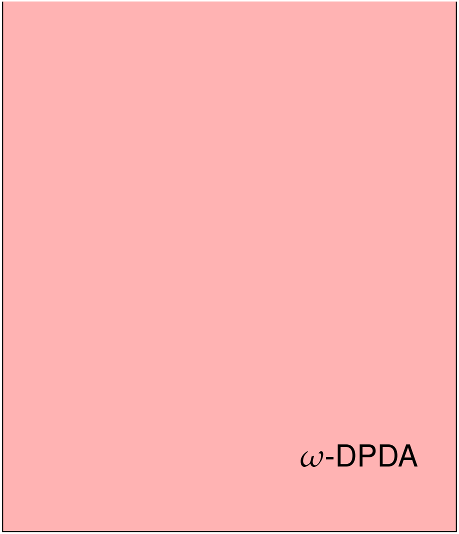
Decision Problems for Deterministic Pushdown Automata on Infinite Words

Christof Löding

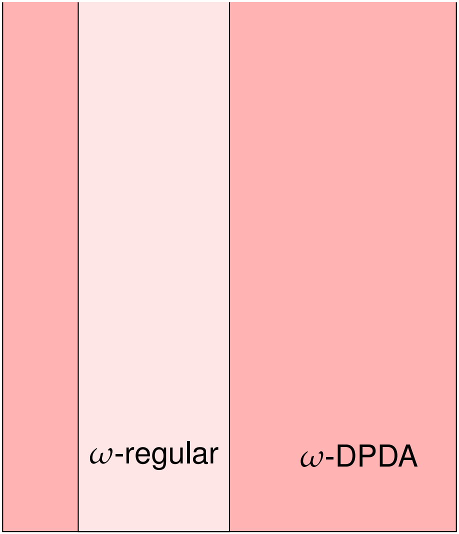
RWTH Aachen University, Germany

AFL 2014, 14th International Conference on Automata and Formal Languages, May 27–29, 2014

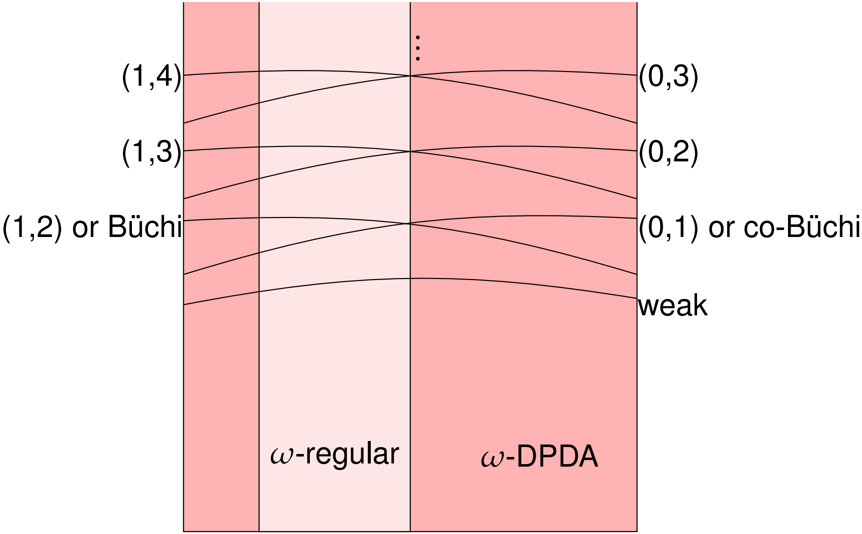
Descriptive Complexity of ω-DPDA Languages



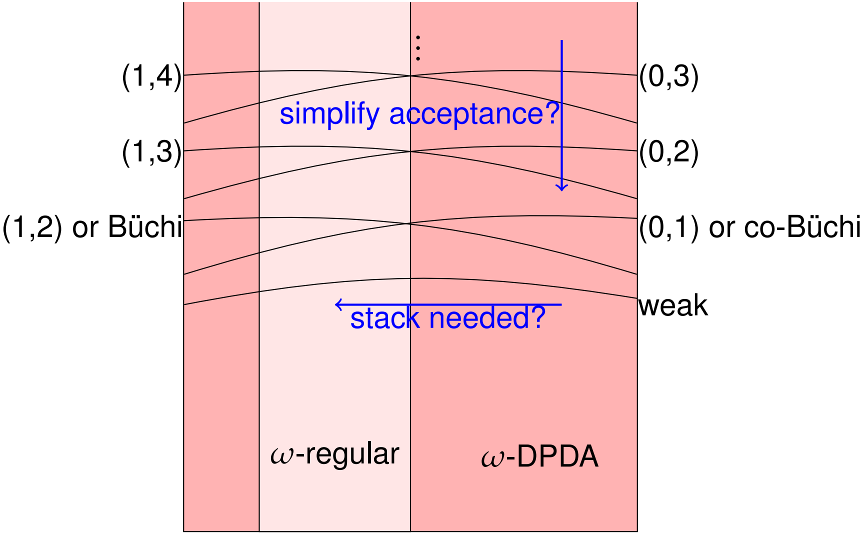
Descriptive Complexity of ω-DPDA Languages



Descriptive Complexity of ω-DPDA Languages



Descriptive Complexity of ω-DPDA Languages



Pushdown Automata

Finite state machine + unbounded pushdown store (stack)

In this talk: Only deterministic automata (with ε-transitions)

DPDA on finite words: Set F of final states, as usual.

$L_*(\mathcal{A})$ language of finite words accepted by $\mathcal{A}$

Pushdown Automata

Finite state machine + unbounded pushdown store (stack)

In this talk: Only deterministic automata (with ε-transitions)

DPDA on finite words: Set F of final states, as usual.

$L_*(\mathcal{A})$ language of finite words accepted by $\mathcal{A}$

ω-DPDA on infinite words:

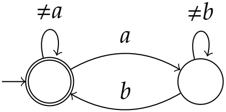
- Büchi condition: set F of accepting states
  run accepting if it visits F infinitely often
- Parity condition: mapping $\Omega : Q \to \mathbb{N}$
  run accepting if highest priority seen infinitely often is even

$L_\omega(\mathcal{A})$ language of infinite words accepted by $\mathcal{A} = (\cdots, F)$ or $\mathcal{A} = (\cdots, \Omega)$

Example for ω-Languages

alphabet A with $a, b \in A$, where a represents request, b grant

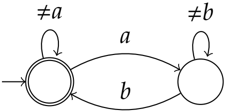
Regular ω-language: Whenever a occurs, then later b occurs.



Example for ω-Languages

alphabet A with $a, b \in A$, where a represents request, b grant

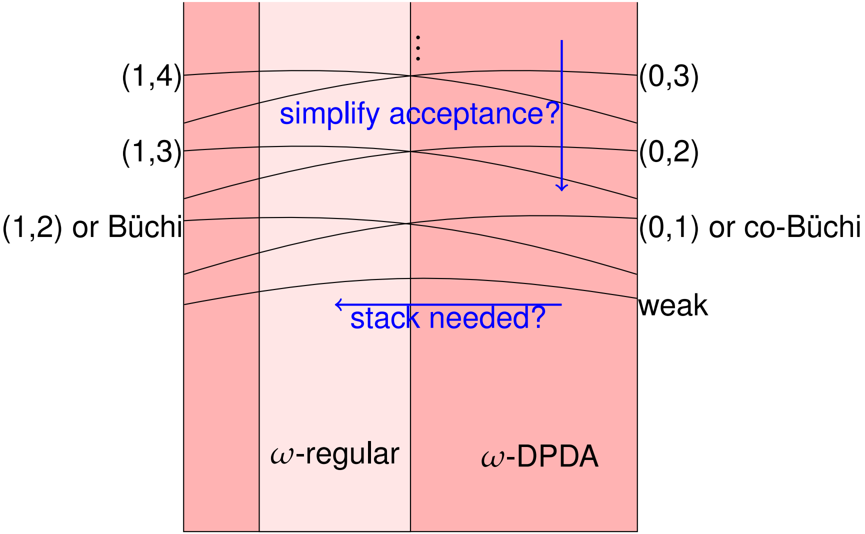
Regular ω-language: Whenever a occurs, then later b occurs.



DPDA ω-language: For every a there is a matching b later.

Büchi DPDA: Use stack to count the number of “unanswered” a. Go to accepting state whenever this number is 0

Descriptive Complexity of ω-DPDA Languages



Outline

Problem Setting

Regularity problem for DPDA:

Given: DPDA $\mathcal{A}$

Question: Is $L_*(\mathcal{A})$ regular?

Theorem (Stearns 1967, Valiant 1975). The regularity problem for DPDAs is decidable.

Problem Setting

Regularity problem for DPDA:

Given: DPDA $\mathcal{A}$

Question: Is $L_*(\mathcal{A})$ regular?

Theorem (Stearns 1967, Valiant 1975). The regularity problem for DPDAs is decidable.

Regularity problem for ω-DPDA:

Given: ω-DPDA $\mathcal{A}$

Question: Is $L_w(\mathcal{A})$ regular?

Open problem (Cohen/Gold 1978): Is the regularity problem for ω-DPDAs decidable?

Main Difficulty

Decision procedure for finite words uses characterization of regular languages in terms of Myhill/Nerode congruence:

- Configurations with large stack must be equivalent to smaller configurations if the language is regular.
- A finite automaton uses the configurations up to some bound and redirects transitions to larger configurations to the equivalent smaller ones.

There is no such characterization of regular ω-languages in terms of congruences.

Regularity vs. ω-Regularity

Idea: Can we use the results on regularity for DPDA also for ω-DPDA?

Regularity vs. ω-Regularity

Idea: Can we use the results on regularity for DPDA also for ω-DPDA?

A Büchi-DPDA $\mathcal{A}$ defines the language $L_\omega(\mathcal{A})$.

But it also can be seen as a DPDA defining $L_*(\mathcal{A})$.

Question: Is $L_\omega(\mathcal{A})$ regular iff $L_*(\mathcal{A})$ is regular?

Regularity vs. ω-Regularity

Idea: Can we use the results on regularity for DPDA also for ω-DPDA?

A Büchi-DPDA $\mathcal{A}$ defines the language $L_\omega(\mathcal{A})$.

But it also can be seen as a DPDA defining $L_*(\mathcal{A})$.

Question: Is $L_\omega(\mathcal{A})$ regular iff $L_*(\mathcal{A})$ is regular?

Answer:

- If $L_*(\mathcal{A})$ is regular, then $L_\omega(\mathcal{A})$ is:
  $L_*(\mathcal{A}) = L_*(\mathcal{B}) \;\; \Rightarrow \;\; L_\omega(\mathcal{A}) = L_\omega(\mathcal{B})$

Regularity vs. ω-Regularity

Idea: Can we use the results on regularity for DPDA also for ω-DPDA?

A Büchi-DPDA $\mathcal{A}$ defines the language $L_\omega(\mathcal{A})$.

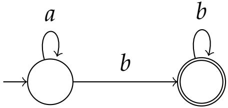
But it also can be seen as a DPDA defining $L_*(\mathcal{A})$.

Question: Is $L_\omega(\mathcal{A})$ regular iff $L_*(\mathcal{A})$ is regular?

Answer:

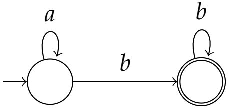
- If $L_*(\mathcal{A})$ is regular, then $L_\omega(\mathcal{A})$ is:
  $L_*(\mathcal{A}) = L_*(\mathcal{B}) \;\Rightarrow\; L_\omega(\mathcal{A}) = L_\omega(\mathcal{B})$
- The other direction does not hold, in general:
  $L_\omega(\mathcal{A}) = L_\omega(\mathcal{B}) \;\not\Rightarrow\; L_*(\mathcal{A}) = L_*(\mathcal{B})$

Regularity vs. ω-Regularity – Example

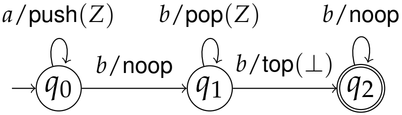
$L = a^* b^\omega$ is obviously regular: 

Regularity vs. ω-Regularity – Example

$L = a^* b^\omega$ is obviously regular:



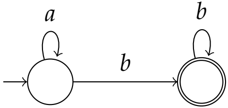
Consider the following Büchi-DPDA $\mathcal{A}$ (informal notation on transitions: letter/stack operation):



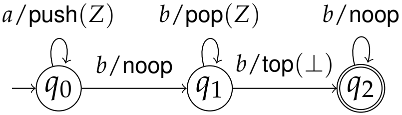
$L_\omega(\mathcal{A}) = a^* b^\omega$ but $L_*(\mathcal{A}) = \{a^m b^n \mid n > m + 1\}$ is non-regular.

Regularity vs. ω-Regularity – Example

$L = a^* b^\omega$ is obviously regular:



Consider the following Büchi-DPDA $\mathcal{A}$ (informal notation on transitions: letter/stack operation):



$L_\omega(\mathcal{A}) = a^* b^\omega$ but $L_*(\mathcal{A}) = \{a^m b^n \mid n > m + 1\}$ is non-regular.

Solution in this case: Make q_1 accepting.

Then $L_\omega(\mathcal{A}) = a^* b^\omega$ and $L_*(\mathcal{A}) = a^* b b^*$.

General Solution for Weak Büchi DPDA

A Büchi DPDA is called weak if there there is a bound on the number of possible alternations between accepting and rejecting states.

General Solution for Weak Büchi DPDA

A Büchi DPDA is called weak if there there is a bound on the number of possible alternations between accepting and rejecting states.

Normalize weak Büchi DPDA as follows:

- There are configurations that cannot appear infinitely often in a run.
- Those can be made accepting or rejecting without changing the accepted ω-language.
- Modify the DPDA such that from each configuration the number of alternations between accepting and rejecting becomes minimal.
- This transformation might cause an exponential blow-up in the worst case.

Regularity for Weak Büchi DPDA

Theorem (L./Repke 2012)

1. For a weak Büchi DPDA $\mathcal{A}$ in normal form, the language $L_\omega(\mathcal{A})$ is regular if, and only if, $L_*(\mathcal{A})$ is regular.
2. Given two weak Büchi DPDAs $\mathcal{A}$ and $\mathcal{B}$ in normal form, $L_\omega(\mathcal{A}) = L_\omega(\mathcal{B})$ if, and only if, $L_*(\mathcal{A}) = L_*(\mathcal{B})$.

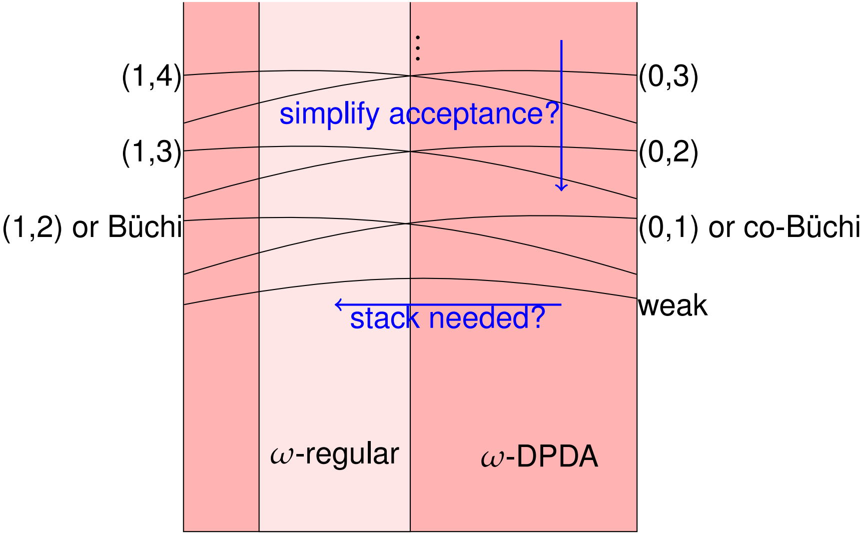
Corollary (L./Repke 2012) The regularity problem for weak ω-DPDAs is decidable.

Consequence for the Equivalence Problem

Theorem (Senizergues 2001). The equivalence problem for DPDAs is decidable.

Corollary (L./Repke 2012). The equivalence problem for weak ω-DPDAs is decidable.

Descriptive Complexity of ω-DPDA Languages



Outline

Parity Index Problem

Given: Parity DPDA $\mathcal{A}$, finite $P \subseteq \mathbb{N}$

Question: Does there exist a P-parity DPDA $\mathcal{B}$ with $L(\mathcal{A}) = L(\mathcal{B})$?

Note: P can always be an interval of $\mathbb{N}$ starting in 0 or 1.

Classification Game

Given parity DPDA $\mathcal{A}$ and target set P of priorites.

Classification game $G(\mathcal{A}, P)$:

- Player AUTOMATON chooses input letters
- Player CLASSIFIER chooses priority from P

Classification Game

Given parity DPDA $\mathcal{A}$ and target set P of priorites.

Classification game $G(\mathcal{A}, P)$:

- Player AUTOMATON chooses input letters
- Player CLASSIFIER chooses priority from P

AUTOMATON	a_0
CLASSIFIER	

Classification Game

Given parity DPDA $\mathcal{A}$ and target set P of priorites.

Classification game $G(\mathcal{A}, P)$:

- Player Automaton chooses input letters
- Player Classifier chooses priority from P

Automaton	a_0
Classifier	p_0

Classification Game

Given parity DPDA $\mathcal{A}$ and target set P of priorites.

Classification game $G(\mathcal{A}, P)$:

- Player Automaton chooses input letters
- Player Classifier chooses priority from P

Automaton	$a_0\ a_1$
Classifier	p_0

Classification Game

Given parity DPDA $\mathcal{A}$ and target set P of priorites.

Classification game $G(\mathcal{A}, P)$:

- Player AUTOMATON chooses input letters
- Player CLASSIFIER chooses priority from P

AUTOMATON	$a_0\ a_1$
CLASSIFIER	$p_0\ p_1$

Classification Game

Given parity DPDA $\mathcal{A}$ and target set P of priorites.

Classification game $G(\mathcal{A}, P)$:

- Player AUTOMATON chooses input letters
- Player CLASSIFIER chooses priority from P

AUTOMATON	$a_0\ a_1\ a_2$
CLASSIFIER	$p_0\ p_1$

Classification Game

Given parity DPDA $\mathcal{A}$ and target set P of priorites.

Classification game $G(\mathcal{A}, P)$:

- Player AUTOMATON chooses input letters
- Player CLASSIFIER chooses priority from P

AUTOMATON	$a_0\ a_1\ a_2$
CLASSIFIER	$p_0\ p_1\ p_2$

Classification Game

Given parity DPDA $\mathcal{A}$ and target set P of priorites.

Classification game $G(\mathcal{A}, P)$:

- Player AUTOMATON chooses input letters
- Player CLASSIFIER chooses priority from P

AUTOMATON	$a_0\ a_1\ a_2 \cdots$
CLASSIFIER	$p_0\ p_1\ p_2$

Classification Game

Given parity DPDA $\mathcal{A}$ and target set P of priorites.

Classification game $G(\mathcal{A}, P)$:

- Player AUTOMATON chooses input letters
- Player CLASSIFIER chooses priority from P

AUTOMATON	$a_0\ a_1\ a_2 \cdots$
CLASSIFIER	$p_0\ p_1\ p_2 \cdots$

Classification Game

Given parity DPDA $\mathcal{A}$ and target set P of priorites.

Classification game $G(\mathcal{A}, P)$:

- Player AUTOMATON chooses input letters
- Player CLASSIFIER chooses priority from P

AUTOMATON	$a_0\ a_1\ a_2 \cdots$
CLASSIFIER	$p_0\ p_1\ p_2 \cdots$

Winning condition for CLASSIFIER in infinite plays:

- The word played by AUTOMATON is in $L(\mathcal{A})$ if, and only if, the priority sequence chosen by CLASSIFIER satisfies the parity condition.

Lemma. There is a P-parity DPDA accepting $L(\mathcal{A})$ if, and only if, CLASSIFIER has a winning strategy in $G(\mathcal{A}, P)$.

Decidability

From a general theorem on pushdown games (Walukiewicz 1998), it follows that it is decidable whether Classifier has a winning strategy in $G(\mathcal{A}, P)$.

Corollary. The parity index problem for ω-DPDA is decidable.

Remark: In 1977 it was already shown by Linna that it is decidable whether a given ω-DPDA is equivalent to a Büchi-DPDA.

Outline

Visibly Pushdown Automata – Motivation

In some applications of pushdown automata, the input letter determines the stack operation:

- Analysis of recursive programs: calls and returns of procedures
- XML documents processing: opening and closing tags

Visibly Pushdown Automata – Definition

Partitioned alphabet $A = A_c \cup A_i \cup A_r$ with

- $A_c =$ calls: push one letter onto the stack
- $A_r =$ returns: pop one letter from the stack
- $A_i =$ internal actions: stack remains unchanged

Deterministic visibly pushdown automaton (DVPA): uses three transition functions

- Transition function according to above constraints
- No ε-transitions

Visibly Pushdown Automata – Definition

Partitioned alphabet $A = A_{\mathrm{c}} \cup A_{\mathrm{i}} \cup A_{\mathrm{r}}$ with

- $A_{\mathrm{c}} =$ calls: push one letter onto the stack
- $A_{\mathrm{r}} =$ returns: pop one letter from the stack
- $A_{\mathrm{i}} =$ internal actions: stack remains unchanged

Deterministic visibly pushdown automaton (DVPA): uses three transition functions

- Transition function according to above constraints
- No ε-transitions

Examples:

- $a^n b^n$ is a visibly pushdown language (of finite words) if $a \in A_c$ and $b \in A_r$

Visibly Pushdown Automata – Definition

Partitioned alphabet $A = A_\mathrm{c} \cup A_\mathrm{i} \cup A_\mathrm{r}$ with

- A_c = calls: push one letter onto the stack
- A_r = returns: pop one letter from the stack
- A_i = internal actions: stack remains unchanged

Deterministic visibly pushdown automaton (DVPA): uses three transition functions

- Transition function according to above constraints
- No ε-transitions

Examples:

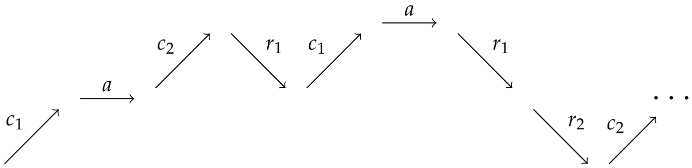
- $a^n b^n$ is a visibly pushdown language (of finite words) if $a \in A_c$ and $b \in A_r$
- $a^n b a^n$ is not a visibly pushdown language, no matter how the partition of the alphabet looks like

Evolution of the Stack

The evolution of the stack only depends on the input, not on the specific automaton.

Illustration:

$A_c = \{c_1, c_2\}$, $A_i = \{a\}$, $A_r = \{r_1, r_2\}$



- Natural notion of matching call and return.
- Definition of VPA in this talk enforces that every return has a matching call.
- There can be calls without matching return.

Closure and Decidability

For a fixed partition $A = A_c \cup A_i \cup A_r$:

Theorem (Alur/Madhusudan 2004).

- On finite words, visibly pushdown automata are closed under union, intersection, and complement. Nondeterministic VPAs can be determinized.

Closure and Decidability

For a fixed partition $A = A_c \cup A_i \cup A_r$:

Theorem (Alur/Madhusudan 2004).

- On finite words, visibly pushdown automata are closed under union, intersection, and complement. Nondeterministic VPAs can be determinized.
- Nondeterministic Büchi VPAs are closed under union, intersection, and complement, but can, in general, not be determinized.

Closure and Decidability

For a fixed partition $A = A_c \cup A_i \cup A_r$:

Theorem (Alur/Madhusudan 2004).

- On finite words, visibly pushdown automata are closed under union, intersection, and complement. Nondeterministic VPAs can be determinized.
- Nondeterministic Büchi VPAs are closed under union, intersection, and complement, but can, in general, not be determinized.

Example: All infinite words containing infinitely many unmatched calls over any alphabet with at least one call and one return, e.g., $A_\mathrm{c} = \{c\}$ and $A_\mathrm{r} = \{r\}$.

Can be accepted by a nondeterminsitic Büchi VPA but not by any parity DVPA.

Determinization – Stair Conditions

A more powerful acceptance condition:

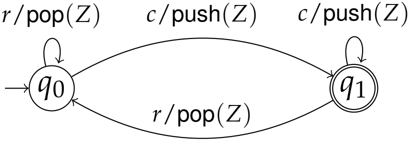
- A configuration in a run is called step if no later configuration has smaller stack height.
- A stair condition (Büchi or parity) is only evaluated on the states occurring on steps

Determinization – Stair Conditions

A more powerful acceptance condition:

- A configuration in a run is called step if no later configuration has smaller stack height.
- A stair condition (Büchi or parity) is only evaluated on the states occurring on steps

Stair Büchi DVPA for "infinitely many unmatched calls":

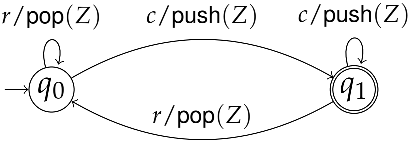


Determinization – Stair Conditions

A more powerful acceptance condition:

- A configuration in a run is called step if no later configuration has smaller stack height.
- A stair condition (Büchi or parity) is only evaluated on the states occurring on steps

Stair Büchi DVPA for "infinitely many unmatched calls":



Theorem (L./Madhusudan/Serre 2004). For each nondeterministic Büchi VPA there is an equivalent deterministic stair parity DVPA.

The Stair Problem

General stair problem:

Given: Stair parity DVPA $\mathcal{A}$

Question: Is there a parity DVPA equivalent to $\mathcal{A}$?

The Stair Problem

General stair problem:

Given: Stair parity DVPA $\mathcal{A}$

Question: Is there a parity DVPA equivalent to $\mathcal{A}$?

Büchi stair problem:

Given: Stair Büchi DVPA $\mathcal{A}$

Question: Is there a parity DVPA equivalent to $\mathcal{A}$?

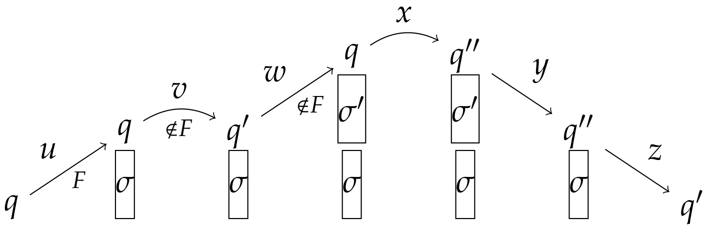
Forbidden Pattern

Theorem. There is a pattern with the following property. A stair Büchi DVPA $\mathcal{A}$ is equivalent to some parity DPDA if, and only if, $\mathcal{A}$ does not contain this pattern. The existence of such a pattern can be decided in polynomial time.

Forbidden Pattern

Theorem. There is a pattern with the following property. A stair Büchi DVPA $\mathcal{A}$ is equivalent to some parity DPDA if, and only if, $\mathcal{A}$ does not contain this pattern. The existence of such a pattern can be decided in polynomial time.

Illlustration of the pattern:



Outline

Conclusion

Three decidability results in this talk:

- Regularity for weak Büchi-DPDA: reduction to finite words
- Parity index for ω-DPDA: classification game
- Stair Büchi-DVPA to parity DVPA: forbidden pattern

Some open problems:

- Regularity for general parity DPDA
- Is a given parity DPDA equivalent to a weak DPDA?
- Removal of stair condition for general parity DVPAs
- Equivalence for general parity DPDA