Languages of Boundedly-Ambiguous Vector Addition Systems with States

Wojciech Czerwiński, **Łukasz Orlikowski**

University of Warsaw

29.08.2025

Plan

Plan

1. Introduction
2. Tools

Plan

Plan

Deterministic systems

- Deterministic systems are simple (we know what to do next)

Deterministic systems

- Deterministic systems are simple (we know what to do next)
- We often know efficient algorithms for deterministic systems

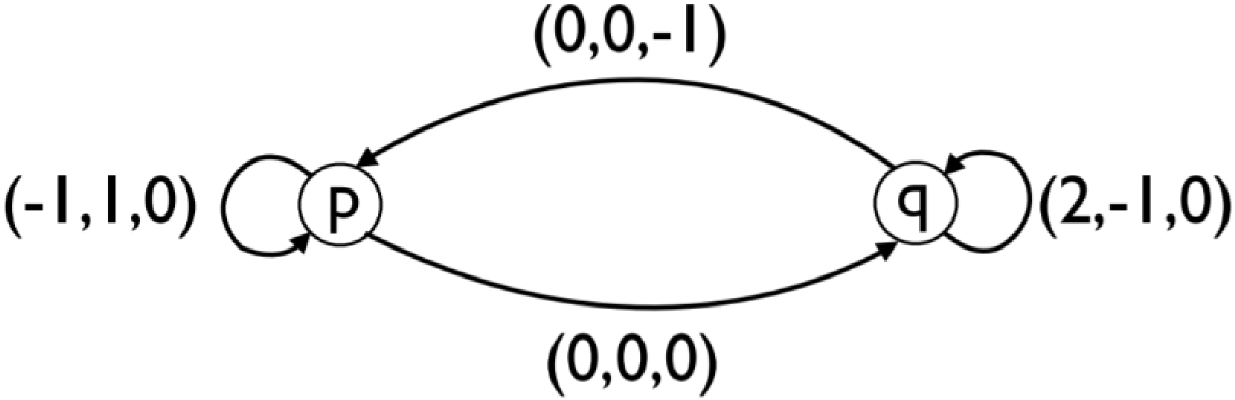
Deterministic systems

- Deterministic systems are simple (we know what to do next)
- We often know efficient algorithms for deterministic systems
- Determinization is often costly (exponential blow-up)

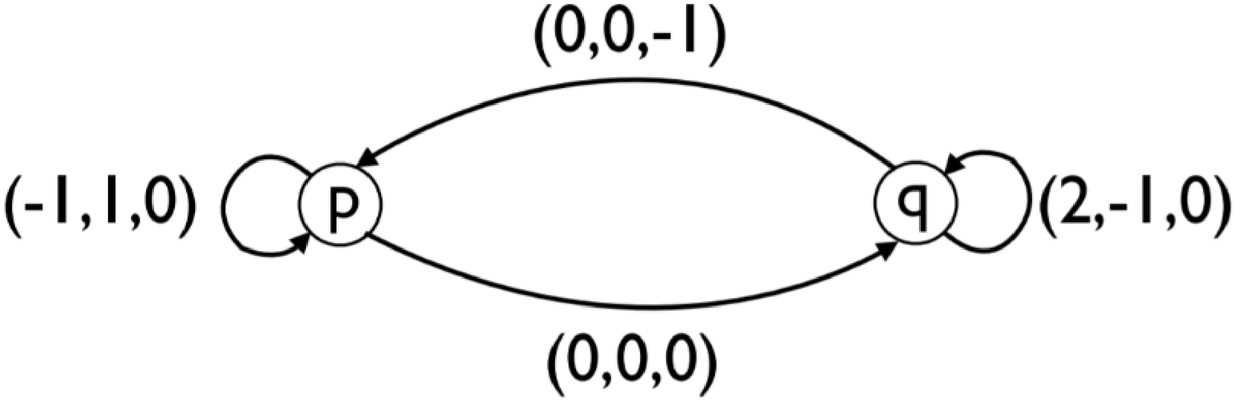
Deterministic systems

- Deterministic systems are simple (we know what to do next)
- We often know efficient algorithms for deterministic systems
- Determinization is often costly (exponential blow-up)
- Hence we often look for a middle-ground (some extensions of determinism)

VASS

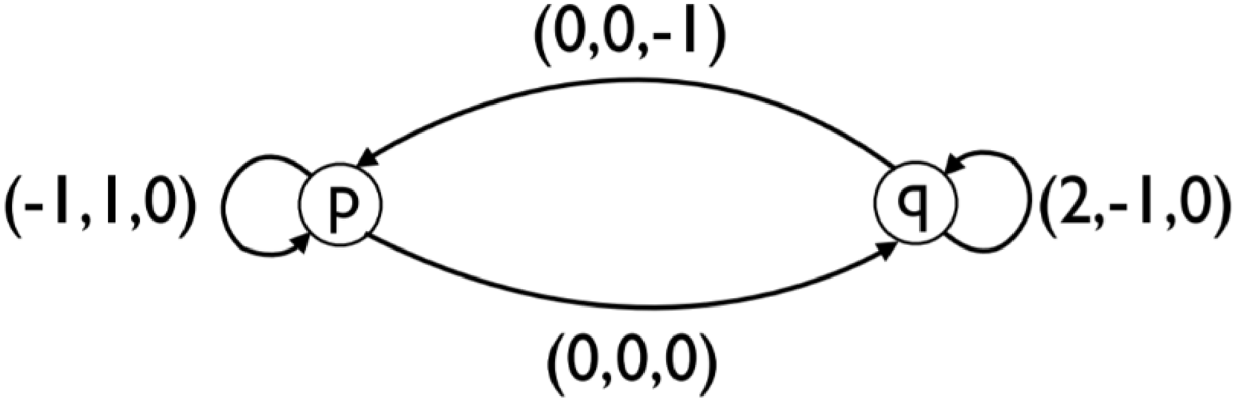


VASS



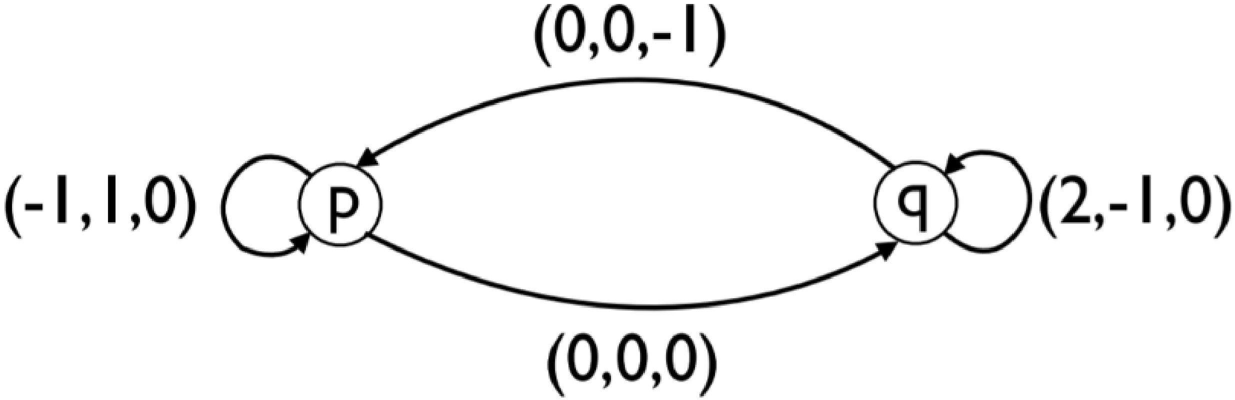
- We can equip a VASS with letters on transitions, initial and accepting configurations to recognize languages by it

VASS



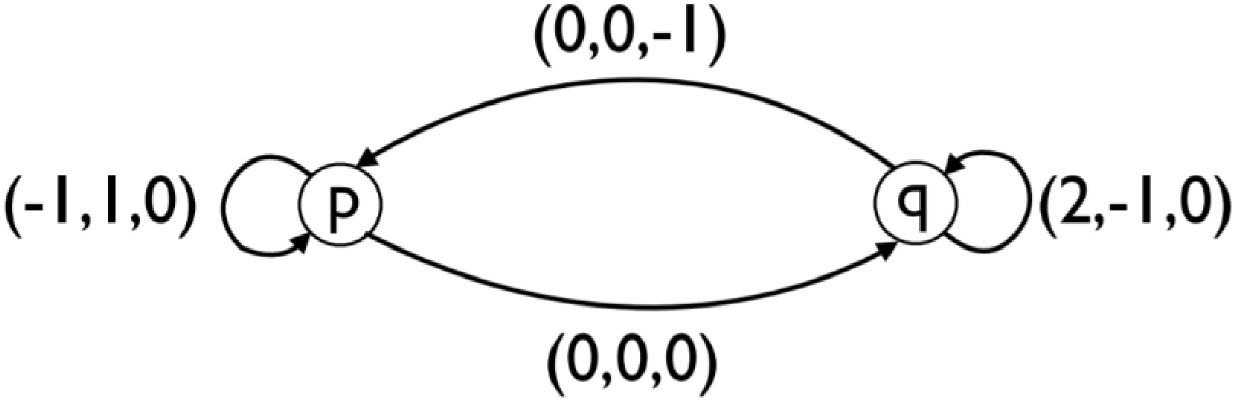
- We can equip a VASS with letters on transitions, initial and accepting configurations to recognize languages by it
- We consider coverability VASS where configuration is accepting if it covers configuration from a finite set

VASS



- We can equip a VASS with letters on transitions, initial and accepting configurations to recognize languages by it
- We consider coverability VASS where configuration is accepting if it covers configuration from a finite set
- VASS is k-ambiguous if for each word w there are at most k accepting runs over it

VASS



- We can equip a VASS with letters on transitions, initial and accepting configurations to recognize languages by it
- We consider coverability VASS where configuration is accepting if it covers configuration from a finite set
- VASS is k-ambiguous if for each word w there are at most k accepting runs over it
- VASS is unambiguous if for each word w there is at most one accepting run over it

Boundedly-ambiguous VASSs

- Boundedly-ambiguous VASSs are easier

Boundedly-ambiguous VASSs

- Boundedly-ambiguous VASSs are easier
- Deciding language equivalence is undecidable **(Jančar, 1995)**

Boundedly-ambiguous VASSs

- Boundedly-ambiguous VASSs are easier
- Deciding language equivalence is undecidable **(Jančar, 1995)**
- Deciding language equivalence is decidable if both VASSs are k-ambiguous **(Czerwiński and Hofman, 2022)**

Boundedly-ambiguous VASSs

- Boundedly-ambiguous VASSs are easier
- Deciding language equivalence is undecidable **(Jančar, 1995)**
- Deciding language equivalence is decidable if both VASSs are k-ambiguous **(Czerwiński and Hofman, 2022)**
- Universality for 1-VASS is Ackermann-complete **(Hofman, Totzke, 2014)**

Boundedly-ambiguous VASSs

- Boundedly-ambiguous VASSs are easier
- Deciding language equivalence is undecidable **(Jančar, 1995)**
- Deciding language equivalence is decidable if both VASSs are k-ambiguous **(Czerwiński and Hofman, 2022)**
- Universality for 1-VASS is Ackermann-complete **(Hofman, Totzke, 2014)**
- Universality for unambiguous VASS is ExpSpace-complete **(Czerwiński, Figueira, Hofman, 2020)**

How can one show language ambiguity?

- How can one show language ambiguity?

How can one show language ambiguity?

- How can one show language ambiguity?
- For regular or context-free languages we have Pumping Lemma

How can one show language ambiguity?

- How can one show language ambiguity?
- For regular or context-free languages we have Pumping Lemma
- Is language $\{a^n b a^m b a^k \mid n \geq m \vee n \geq k\}$ unambiguous?

How can one show language ambiguity?

- How can one show language ambiguity?
- For regular or context-free languages we have Pumping Lemma
- Is language $\{a^n b a^m b a^k \mid n \geq m \vee n \geq k\}$ unambiguous?
- Do we have an algorithm deciding if the language of a given VASS is k-ambiguous?

Deciding unambiguity

Deciding whether there exists equivalent unambiguous automata or grammar is:

Deciding unambiguity

Deciding whether there exists equivalent unambiguous automata or grammar is:

- Decidable for weighted automata **(Bell and Smertnig)**

Deciding unambiguity

Deciding whether there exists equivalent unambiguous automata or grammar is:

- Decidable for weighted automata **(Bell and Smertnig)**
- Undecidable for context-free languages **(Ginsburg and Ullian)**

Tools

First tool (domination block)

Lemma

For every $k \in \mathbb{N}_+$ the language

$$L_k = \{a^{n_1}ba^{n_2}ba^{n_3}b\ldots a^{n_{k+2}} \mid \exists_{i\in[k+1]}\ n_i \geq n_{i+1}\}$$

is not recognised by a k-ambiguous VASS.

First tool (dominating block)

Lemma

Let Σ be an alphabet such that $b \notin \Sigma$ and let L be a language over Σ. For each function $f : L \to \mathbb{N}_\omega$ such that $\sup f = \omega$ language $L_1 = \{a^{n_1}ba^{n_2}ba^{n_3}b \ldots a^{n_{k+2}}bw \mid w \in L, \exists_{1 \leq i \leq k+1} n_i \geq n_{i+1} \vee n_{k+2} \geq f(w)\}$ is not recognised by a k-ambiguous VASS.

Second tool

For any language $L \subseteq \{a, b\}^*$ such that for each $w \in L$ we have $\#_b(w) = l$ for some $l \in \mathbb{N}$ we define

$$im(L) = \{(a_1, a_2, \ldots, a_{l+1}) \mid a_1, a_2, \ldots, a_{l+1} \in \mathbb{N}, a^{a_1} b a^{a_2} b \ldots b a^{a_{l+1}} \in L\}$$

Second tool

For any language $L \subseteq \{a,b\}^*$ such that for each $w \in L$ we have $\#_b(w) = l$ for some $l \in \mathbb{N}$ we define

$$im(L) = \{(a_1, a_2, \ldots, a_{l+1}) \mid a_1, a_2, \ldots, a_{l+1} \in \mathbb{N}, a^{a_1}ba^{a_2}b\ldots ba^{a_{l+1}} \in L\}$$

Lemma

- *L is recognised by k-ambiguous VASS V.*
- *There exists $l \in \mathbb{N}$ such that for each $w \in L$ we have $\#_b(w) = l$.*

Then $im(L)$ is a semilinear set.

Second tool

Corollary

Let $a, b \in \Sigma$, $L \subseteq \Sigma^*$ be a language recognised by a k-ambiguous VASS and for $n \in \mathbb{N}$ let $K_n \subseteq \{a, b\}^*$ be the language of words containing exactly n letters b. Then for any $n \in N$ we have that $im(L \cap K_n)$ is a semilinear set.

Second tool

Corollary

Let $a, b \in \Sigma$, $L \subseteq \Sigma^*$ be a language recognised by a k-ambiguous VASS and for $n \in \mathbb{N}$ let $K_n \subseteq \{a, b\}^*$ be the language of words containing exactly n letters b. Then for any $n \in N$ we have that $im(L \cap K_n)$ is a semilinear set.

- Hence $a^n b a^{\leq 2^n}$ is not recognised by a k-ambiguous VASS for any $k \in \mathbb{N}_+$.

Main result

Main result

Theorem

For any $k \in \mathbb{N}_+$ it is undecidable to check whether the language of a given 1-VASS accepting by coverability condition is k-ambiguous.

Main result

Theorem

For any $k \in \mathbb{N}_+$ it is undecidable to check whether the language of a given 1-VASS accepting by coverability condition is k-ambiguous.

- The proof goes by a reduction from the finitness problem of the reachability set of a lossy counter machine

Main result

Theorem

For any $k \in \mathbb{N}_+$ it is undecidable to check whether the language of a given 1-VASS accepting by coverability condition is k-ambiguous.

- The proof goes by a reduction from the finitness problem of the reachability set of a lossy counter machine
- If the reachability set is finite then language of a produced 1-VASS is regular (hence k-ambiguous)

Main result

Theorem

For any $k \in \mathbb{N}_+$ it is undecidable to check whether the language of a given 1-VASS accepting by coverability condition is k-ambiguous.

- The proof goes by a reduction from the finitness problem of the reachability set of a lossy counter machine
- If the reachability set is finite then language of a produced 1-VASS is regular (hence k-ambiguous)
- If the reachability set is infinite then language of a produced 1-VASS is not k-ambiguous

Main result

Theorem

For any $k \in \mathbb{N}_+$ it is undecidable to check whether the language of a given 1-VASS accepting by coverability condition is k-ambiguous.

- The proof goes by a reduction from the finitness problem of the reachability set of a lossy counter machine
- If the reachability set is finite then language of a produced 1-VASS is regular (hence k-ambiguous)
- If the reachability set is infinite then language of a produced 1-VASS is not k-ambiguous
- We use domination block technique ($L_k = \{a^{n_1}ba^{n_2}ba^{n_3}b \ldots a^{n_{k+2}} \mid \exists_{i \in [k+1]}\ n_i \geq n_{i+1}\}$)

Main result

For any class $\mathcal{C}$ of languages containing all regular languages and contained in the class of all boundedly-ambiguous VASS languages it is undecidable to check whether the language of a given 1-VASS accepting by coverability condition belongs to $\mathcal{C}$.

Main result

For any class $\mathcal{C}$ of languages containing all regular languages and contained in the class of all boundedly-ambiguous VASS languages it is undecidable to check whether the language of a given 1-VASS accepting by coverability condition belongs to $\mathcal{C}$.

- unambiguous VASS

Main result

For any class $\mathcal{C}$ of languages containing all regular languages and contained in the class of all boundedly-ambiguous VASS languages it is undecidable to check whether the language of a given 1-VASS accepting by coverability condition belongs to $\mathcal{C}$.

- unambiguous VASS
- k-ambiguous VASS for given $k \in \mathbb{N}$

Main result

For any class $\mathcal{C}$ of languages containing all regular languages and contained in the class of all boundedly-ambiguous VASS languages it is undecidable to check whether the language of a given 1-VASS accepting by coverability condition belongs to $\mathcal{C}$.

- unambiguous VASS
- k-ambiguous VASS for given $k \in \mathbb{N}$
- boundedly ambiguous VASS

Main result

For any class $\mathcal{C}$ of languages containing all regular languages and contained in the class of all boundedly-ambiguous VASS languages it is undecidable to check whether the language of a given 1-VASS accepting by coverability condition belongs to $\mathcal{C}$.

- unambiguous VASS
- k-ambiguous VASS for given $k \in \mathbb{N}$
- boundedly ambiguous VASS
- **deterministic VASS**

Main result

For any class $\mathcal{C}$ of languages containing all regular languages and contained in the class of all boundedly-ambiguous VASS languages it is undecidable to check whether the language of a given 1-VASS accepting by coverability condition belongs to $\mathcal{C}$.

- unambiguous VASS
- k-ambiguous VASS for given $k \in \mathbb{N}$
- boundedly ambiguous VASS
- **deterministic VASS**
- **regular languages**

Deciding regularity

Deciding regularity

Theorem

For every $k \in \mathbb{N}$ it is decidable whether for a given k-ambiguous VASS A language $L(A)$ is regular.

Deciding regularity

Theorem

For every $k \in \mathbb{N}$ it is decidable whether for a given k-ambiguous VASS A language $L(A)$ is regular.

- Regularity is undecidable in general for VASS and for other notions of restricted nondeterminism (like history-deterministic VASS)

Deciding regularity

Theorem

For every $k \in \mathbb{N}$ it is decidable whether for a given k-ambiguous VASS A language $L(A)$ is regular.

- Regularity is undecidable in general for VASS and for other notions of restricted nondeterminism (like history-deterministic VASS)
- For coverability k-ambiguous VASS V one can construct reachability VASS V' recognising complement of $L(V)$

Deciding regularity

Theorem

For every $k \in \mathbb{N}$ it is decidable whether for a given k-ambiguous VASS A language $L(A)$ is regular.

- Regularity is undecidable in general for VASS and for other notions of restricted nondeterminism (like history-deterministic VASS)
- For coverability k-ambiguous VASS V one can construct reachability VASS V' recognising complement of $L(V)$
- Hence $L(V)$ is regular if and only if $L(V)$ and $L(V')$ are regular separable (there exists regular language L such that $L(V) \subseteq L$ and $L \cap L(V') = \emptyset$)

Summary and conclusions

- Two tools for showing, that language is not a language of a k-ambiguous VASS

Summary and conclusions

- Two tools for showing, that language is not a language of a k-ambiguous VASS
- Deciding whether there exists equivalent k-ambiguous VASS is undecidable

Summary and conclusions

- Two tools for showing, that language is not a language of a k-ambiguous VASS
- Deciding whether there exists equivalent k-ambiguous VASS is undecidable
- Deciding whether language of a k-ambiguous VASS is regular is decidable

Summary and conclusions

- Two tools for showing, that language is not a language of a k-ambiguous VASS
- Deciding whether there exists equivalent k-ambiguous VASS is undecidable
- Deciding whether language of a k-ambiguous VASS is regular is decidable
- Closure properties and other properties can be found in the paper

Summary and conclusions

- Two tools for showing, that language is not a language of a k-ambiguous VASS
- Deciding whether there exists equivalent k-ambiguous VASS is undecidable
- Deciding whether language of a k-ambiguous VASS is regular is decidable
- Closure properties and other properties can be found in the paper
- Future: Consider VASS accepting by configurations

Summary and conclusions

- Two tools for showing, that language is not a language of a k-ambiguous VASS
- Deciding whether there exists equivalent k-ambiguous VASS is undecidable
- Deciding whether language of a k-ambiguous VASS is regular is decidable
- Closure properties and other properties can be found in the paper
- Future: Consider VASS accepting by configurations
- Future: Consider determinisation of boundedly-ambiguous VASS

Summary and conclusions

- Two tools for showing, that language is not a language of a k-ambiguous VASS
- Deciding whether there exists equivalent k-ambiguous VASS is undecidable
- Deciding whether language of a k-ambiguous VASS is regular is decidable
- Closure properties and other properties can be found in the paper
- Future: Consider VASS accepting by configurations
- Future: Consider determinisation of boundedly-ambiguous VASS

Summary and conclusions

- Two tools for showing, that language is not a language of a k-ambiguous VASS
- Deciding whether there exists equivalent k-ambiguous VASS is undecidable
- Deciding whether language of a k-ambiguous VASS is regular is decidable
- Closure properties and other properties can be found in the paper
- Future: Consider VASS accepting by configurations
- Future: Consider determinisation of boundedly-ambiguous VASS

Thank You!