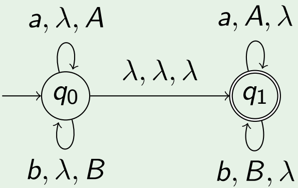
Valence automata as a generalization of automata with storage

Georg Zetzsche

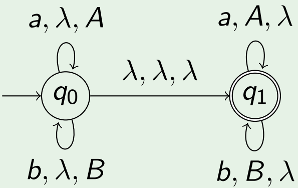
Technische Universität Kaiserslautern

ALFA 2013

Example (Pushdown automaton)

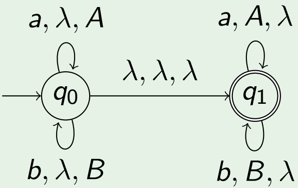


Example (Pushdown automaton)



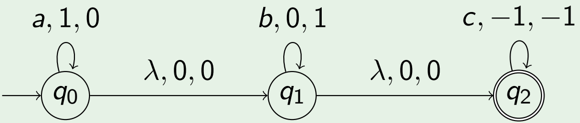
$$L = \{ww^{\text{rev}} \mid w \in \{a, b\}^*\}$$

Example (Pushdown automaton)

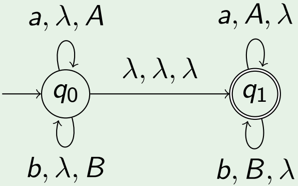


$$L = \{ww^{\text{rev}} \mid w \in \{a, b\}^*\}$$

Example (Blind counter automaton)

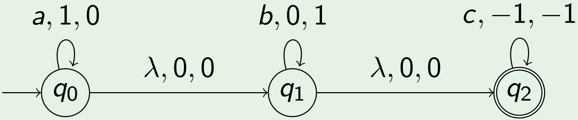


Example (Pushdown automaton)



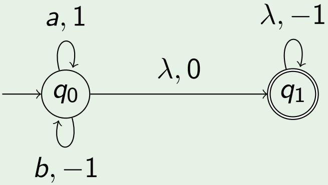
$$L = \{ww^{\text{rev}} \mid w \in \{a, b\}^*\}$$

Example (Blind counter automaton)



$$L = \{a^n b^n c^n \mid n \geqslant 0\}$$

Example (Partially blind counter automaton)



$$L = \{w \in \{a,b\}^* \quad | \quad |p|_a \geqslant |p|_b \text{ for any prefix } p \text{ of } w\}$$

Automata models that extend finite automata by some storage mechanism:

- Pushdown automata
- Blind counter automata
- Partially blind counter automata
- Turing machines

Automata models that extend finite automata by some storage mechanism:

- Pushdown automata
- Blind counter automata
- Partially blind counter automata
- Turing machines

Each storage mechanism consists of:

- States: set S of states
- Operations: partial maps $\alpha_1, \ldots, \alpha_n : S \to S$

Model	States	Operations
Pushdown automata	$S = \Gamma^*$	$\text{push}_a : w \mapsto wa,\ a \in \Gamma$ $\text{pop}_a : wa \mapsto w,\ a \in \Gamma$

Model	States	Operations
Pushdown automata	$S = \Gamma^*$	$\mathrm{push}_a : w \mapsto wa,\ a \in \Gamma$ $\mathrm{pop}_a : wa \mapsto w,\ a \in \Gamma$
Blind counter automata	$S = \mathbb{Z}^n$	$\mathrm{inc}_i : (x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_i + 1, \ldots, x_n)$ $\mathrm{dec}_i : (x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_i - 1, \ldots, x_n)$

Model	States	Operations
Pushdown automata	$S = \Gamma^*$	$\mathrm{push}_a : w \mapsto wa,\ a \in \Gamma$ $\mathrm{pop}_a : wa \mapsto w,\ a \in \Gamma$
Blind counter automata	$S = \mathbb{Z}^n$	$\mathrm{inc}_i : (x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_i + 1, \ldots, x_n)$ $\mathrm{dec}_i : (x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_i - 1, \ldots, x_n)$
Partially blind counter automata	$S = \mathbb{N}^n$	$\mathrm{inc}_i : (x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_i + 1, \ldots, x_n)$ $\mathrm{dec}_i : (x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_i - 1, \ldots, x_n)$

Model	States	Operations
Pushdown automata	$S = \Gamma^*$	$\mathrm{push}_a : w \mapsto wa,\ a \in \Gamma$ $\mathrm{pop}_a : wa \mapsto w,\ a \in \Gamma$
Blind counter automata	$S = \mathbb{Z}^n$	$\mathrm{inc}_i : (x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_i + 1, \ldots, x_n)$ $\mathrm{dec}_i : (x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_i - 1, \ldots, x_n)$
Partially blind counter automata	$S = \mathbb{N}^n$	$\mathrm{inc}_i : (x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_i + 1, \ldots, x_n)$ $\mathrm{dec}_i : (x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_i - 1, \ldots, x_n)$

Observation

Here, a sequence $\beta_1, \ldots, \beta_k$ of operations is valid if and only if

$$\beta_1 \circ \cdots \circ \beta_k = \mathrm{id}$$

Definition

A *monoid* is

- a set M together with
- an associative binary operation $\cdot : M \times M \to M$ and
- a neutral element $1 \in M$ ($a1 = 1a = a$ for any $a \in M$).

Definition

A *monoid* is

- a set M together with
- an associative binary operation $\cdot : M \times M \to M$ and
- a neutral element $1 \in M$ ($a1 = 1a = a$ for any $a \in M$).

Storage mechanisms as monoids

- Let S be a set of states and $\alpha_1, \ldots, \alpha_n : S \to S$ partial maps.
- The set of all compositions of $\alpha_1, \ldots, \alpha_n$ is a monoid M.
- The identity map is the neutral element of M.
- M is a decription of the storage mechanism.

Valence automata

Common generalization: Valence Automata

Valence automaton over M:

- Finite automaton with edges $p \xrightarrow{w|m} q$, $w \in \Sigma^*$, $m \in M$.

Valence automata

Common generalization: Valence Automata

Valence automaton over M:

- Finite automaton with edges $p \xrightarrow{w|m} q$, $w \in \Sigma^*$, $m \in M$.
- Run $q_0 \xrightarrow{w_1|m_1} q_1 \xrightarrow{w_2|m_2} \cdots \xrightarrow{w_n|m_n} q_n$ is *accepting for* $w_1 \cdots w_n$ if
  - q_0 is the initial state,
  - q_n is a final state, and

Valence automata

Common generalization: Valence Automata

Valence automaton over M:

- Finite automaton with edges $p \xrightarrow{w|m} q$, $w \in \Sigma^*$, $m \in M$.
- Run $q_0 \xrightarrow{w_1|m_1} q_1 \xrightarrow{w_2|m_2} \cdots \xrightarrow{w_n|m_n} q_n$ is *accepting for* $w_1 \cdots w_n$ if
  - q_0 is the initial state,
  - q_n is a final state, and
  - $m_1 \cdots m_n = 1$.

Valence automata

Common generalization: Valence Automata

Valence automaton over M:

- Finite automaton with edges $p \xrightarrow{w|m} q$, $w \in \Sigma^*$, $m \in M$.
- Run $q_0 \xrightarrow{w_1|m_1} q_1 \xrightarrow{w_2|m_2} \cdots \xrightarrow{w_n|m_n} q_n$ is *accepting for* $w_1 \cdots w_n$ if
  - q_0 is the initial state,
  - q_n is a final state, and
  - $m_1 \cdots m_n = 1$.

Language class

VA(M) languages accepted by valence automata over M.

Valence automata

- Studied throughout the last decades
- In connection with valence grammars (Fernau, Stiebe)
- Expressive power for concrete monoids (Render, Kambites, Corson)
- As acceptors for word problems of groups (Gilman, Elder, Kambites, Ostheimer)

Classical results can now be generalized:

Questions

- Which language classes can be obtained this way?

Classical results can now be generalized:

Questions

- Which language classes can be obtained this way?
- Which storage mechanisms increase the expressive power?

Classical results can now be generalized:

Questions

- Which language classes can be obtained this way?
- Which storage mechanisms increase the expressive power?
- For which storage mechanisms can we determinize?

Classical results can now be generalized:

Questions

- Which language classes can be obtained this way?
- Which storage mechanisms increase the expressive power?
- For which storage mechanisms can we determinize?
- For which are all languages context-free?

Classical results can now be generalized:

Questions

- Which language classes can be obtained this way?
- Which storage mechanisms increase the expressive power?
- For which storage mechanisms can we determinize?
- For which are all languages context-free?
- For which do we have semilinearity of all languages?

Classical results can now be generalized:

Questions

- Which language classes can be obtained this way?
- Which storage mechanisms increase the expressive power?
- For which storage mechanisms can we determinize?
- For which are all languages context-free?
- For which do we have semilinearity of all languages?
- For which can we avoid silent transitions?

Classical results can now be generalized:

Questions

- Which language classes can be obtained this way?
- Which storage mechanisms increase the expressive power?
- For which storage mechanisms can we determinize?
- For which are all languages context-free?
- For which do we have semilinearity of all languages?
- For which can we avoid silent transitions?
- For which can we decide, for example, emptiness?

Classical results can now be generalized:

Questions

- Which language classes can be obtained this way?
- Which storage mechanisms increase the expressive power?
- For which storage mechanisms can we determinize?
- For which are all languages context-free?
- For which do we have semilinearity of all languages?
- For which can we avoid silent transitions?
- For which can we decide, for example, emptiness?

Classical results can now be generalized:

Questions

- Which language classes can be obtained this way?
- Which storage mechanisms increase the expressive power?
- For which storage mechanisms can we determinize?
- For which are all languages context-free?
- For which do we have semilinearity of all languages?
- For which can we avoid silent transitions? ← tomorrow!
- For which can we decide, for example, emptiness? ← tomorrow!

Deterministic valence automata

A valence automaton is called deterministic, if

Deterministic valence automata

A valence automaton is called deterministic, if

- every edge is of the form $p \xrightarrow{a|m} q$ with $a \in \Sigma$.

Deterministic valence automata

A valence automaton is called deterministic, if

- every edge is of the form $p \xrightarrow{a|m} q$ with $a \in \Sigma$.
- for each state $p \in Q$ and each letter $a \in \Sigma$, there is at most one edge $p \xrightarrow{a|m} q$ for some $m \in M$, $q \in Q$.

Deterministic valence automata

A valence automaton is called deterministic, if

- every edge is of the form $p \xrightarrow{a|m} q$ with $a \in \Sigma$.
- for each state $p \in Q$ and each letter $a \in \Sigma$, there is at most one edge $p \xrightarrow{a|m} q$ for some $m \in M$, $q \in Q$.

detVA(M) languages accepted by deterministic valence automata over M.

Questions

- When does $\mathrm{VA}(M)$ contain non-regular languages?
- When does $\mathrm{detVA}(M) = \mathrm{VA}(M)$?

Questions

- When does $\mathsf{VA}(M)$ contain non-regular languages?
- When does $\mathsf{detVA}(M) = \mathsf{VA}(M)$?

Theorem (Render 2010, Z. 2011)

The following statements are equivalent:

1. $\mathsf{VA}(M) = \mathsf{REG}$.
2. $\mathsf{detVA}(M) = \mathsf{VA}(M)$.
3. *Every finitely generated submonoid of M possesses only finitely many right-invertible elements.*

Questions

- When does $\mathsf{VA}(M)$ contain non-regular languages?
- When does $\mathsf{detVA}(M) = \mathsf{VA}(M)$?

Theorem (Render 2010, Z. 2011)

The following statements are equivalent:

1. $\mathsf{VA}(M) = \mathsf{REG}$.
2. $\mathsf{detVA}(M) = \mathsf{VA}(M)$.
3. *Every finitely generated submonoid of M possesses only finitely many right-invertible elements.*

(1) $\Leftrightarrow$ (3) has been shown independently by Elaine Render in 2010.

Questions

- When does $\mathsf{VA}(M)$ contain non-regular languages?
- When does $\mathsf{detVA}(M) = \mathsf{VA}(M)$?

Theorem (Render 2010, Z. 2011)

The following statements are equivalent:

1. $\mathsf{VA}(M) = \mathsf{REG}$.
2. $\mathsf{detVA}(M) = \mathsf{VA}(M)$.
3. *Every finitely generated submonoid of M possesses only finitely many right-invertible elements.*

(1) $\Leftrightarrow$ (3) has been shown independently by Elaine Render in 2010.

Definition

A monoid with the above property is called *FRI-monoid*.

$$\mathsf{R}(M) = \{x \in M \mid \exists y \in M : xy = 1\} \quad \overline{\mathsf{R}}(x) = \{y \in M \mid xy = 1\}$$

$$\mathsf{R}(M) = \{x \in M \mid \exists y \in M : xy = 1\} \quad \overline{\mathsf{R}}(x) = \{y \in M \mid xy = 1\}$$
$$\mathsf{L}(M) = \{x \in M \mid \exists y \in M : yx = 1\} \quad \overline{\mathsf{L}}(x) = \{y \in M \mid yx = 1\}$$

$$\mathsf{R}(M) = \{x \in M \mid \exists y \in M : xy = 1\} \qquad \overline{\mathsf{R}}(x) = \{y \in M \mid xy = 1\}$$
$$\mathsf{L}(M) = \{x \in M \mid \exists y \in M : yx = 1\} \qquad \overline{\mathsf{L}}(x) = \{y \in M \mid yx = 1\}$$

$$\mathsf{J}(M) = \{x \in M \mid \exists y, z \in M : yxz = 1\}$$

$$\mathrm{R}(M) = \{x \in M \mid \exists y \in M : xy = 1\} \quad \overline{\mathrm{R}}(x) = \{y \in M \mid xy = 1\}$$
$$\mathrm{L}(M) = \{x \in M \mid \exists y \in M : yx = 1\} \quad \overline{\mathrm{L}}(x) = \{y \in M \mid yx = 1\}$$

$$\mathrm{J}(M) = \{x \in M \mid \exists y, z \in M : yxz = 1\}$$

Lemma (Dichotomy)

For each monoid M, exactly one of the following holds:

1. $\mathrm{R}(M) = \mathrm{L}(M) = \mathrm{J}(M)$ *is a finite group.*
2. *There are infinite subsets $S \subseteq \mathrm{R}(M)$, $S' \subseteq \mathrm{L}(M)$ such that*
    - $\overline{\mathrm{R}}(s) \cap \overline{\mathrm{R}}(t) = \emptyset$ *for any $s, t \in S$ and*
    - $\overline{\mathrm{L}}(s) \cap \overline{\mathrm{L}}(t) = \emptyset$ *for any $s, t \in S'$.*

Definition (Graph products)

Let $\Gamma = (V, E)$ be a simple graph and M_v a monoid for each $v \in V$ with a presentation (A_v, R_v). Then the graph product $\mathbb{M}(\Gamma, (M_v)_{v \in V})$ is given by

$$A = \bigcup_{v \in V} A_v, \quad R = \bigcup_{v \in V} R_v \quad .$$

Intuition:

- $M = \mathbb{M}(\Gamma, (M_v)_{v \in V})$ consists of sequences of elements in $\bigcup_v M_v$
- elements in the same monoid M_v are multiplied as defined in M_v
- an edge between v and w means elements from M_v and M_w commute

Definition (Graph products)

Let $\Gamma = (V, E)$ be a simple graph and M_v a monoid for each $v \in V$ with a presentation (A_v, R_v). Then the graph product $\mathbb{M}(\Gamma, (M_v)_{v \in V})$ is given by

$$A = \bigcup_{v \in V} A_v, \qquad R = \bigcup_{v \in V} R_v \cup \{xy = yx \mid x \in M_v, y \in M_w, \{v, w\} \in E\}.$$

Intuition:

- $M = \mathbb{M}(\Gamma, (M_v)_{v \in V})$ consists of sequences of elements in $\bigcup_v M_v$
- elements in the same monoid M_v are multiplied as defined in M_v
- an edge between v and w means elements from M_v and M_w commute

Specialization: Monoids defined by graphs

Notation

- $\mathbb{B}$: monoid for partially blind counter, $\mathbb{B} = \{a, \bar{a}\}^*/\{a\bar{a} = 1\}$.
- $\mathbb{Z}$: monoid for blind counter, i.e. the group of integers

Specialization: Monoids defined by graphs

Notation

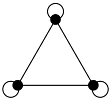
- $\mathbb{B}$: monoid for partially blind counter, $\mathbb{B} = \{a, \bar{a}\}^*/\{a\bar{a} = 1\}$.
- $\mathbb{Z}$: monoid for blind counter, i.e. the group of integers

Monoids $\mathbb{M}\Gamma$

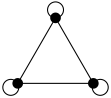
To each graph Γ, we associate the monoid $\mathbb{M}\Gamma$:

- For each unlooped vertex, we have a copy of $\mathbb{B}$
- For each looped vertex, we have a copy of $\mathbb{Z}$
- $\mathbb{M}\Gamma$ is the corresponding graph product

Examples

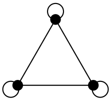


Examples



$\mathbb{Z}^3$

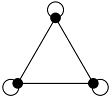
Examples

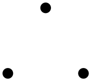


$\mathbb{Z}^3$

Blind multicounter

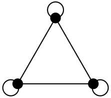
Examples


$\mathbb{Z}^3$

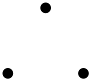


Blind multicounter

Examples



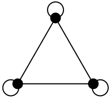
$\mathbb{Z}^3$



$\mathbb{B} * \mathbb{B} * \mathbb{B}$

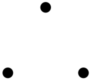
Blind multicounter

Examples



$\mathbb{Z}^3$

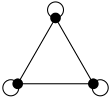
Blind multicounter



$\mathbb{B} * \mathbb{B} * \mathbb{B}$

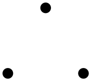
Pushdown

Examples



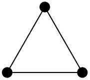
$\mathbb{Z}^3$

Blind multicounter

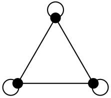


$\mathbb{B} * \mathbb{B} * \mathbb{B}$

Pushdown

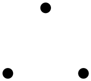


Examples



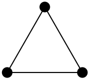
$\mathbb{Z}^3$

Blind multicounter



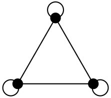
$\mathbb{B} * \mathbb{B} * \mathbb{B}$

Pushdown



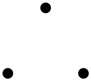
$\mathbb{B}^3$

Examples



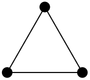
$\mathbb{Z}^3$

Blind multicounter



$\mathbb{B} * \mathbb{B} * \mathbb{B}$

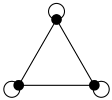
Pushdown



$\mathbb{B}^3$

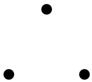
Partially blind multicounter

Examples



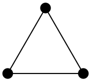
$\mathbb{Z}^3$

Blind multicounter



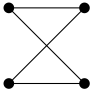
$\mathbb{B} * \mathbb{B} * \mathbb{B}$

Pushdown

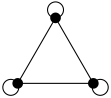


$\mathbb{B}^3$

Partially blind multicounter

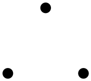


Examples



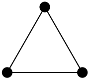
$\mathbb{Z}^3$

Blind multicounter



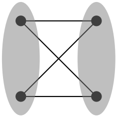
$\mathbb{B} * \mathbb{B} * \mathbb{B}$

Pushdown

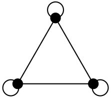


$\mathbb{B}^3$

Partially blind multicounter

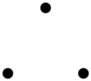


Examples



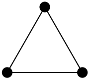
$\mathbb{Z}^3$

Blind multicounter



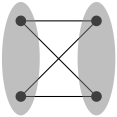
$\mathbb{B} * \mathbb{B} * \mathbb{B}$

Pushdown



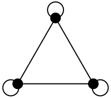
$\mathbb{B}^3$

Partially blind multicounter



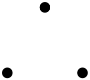
$(\mathbb{B} * \mathbb{B}) \times (\mathbb{B} * \mathbb{B})$

Examples



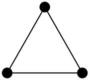
$\mathbb{Z}^3$

Blind multicounter



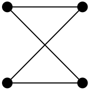
$\mathbb{B} * \mathbb{B} * \mathbb{B}$

Pushdown



$\mathbb{B}^3$

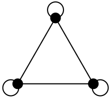
Partially blind multicounter



$(\mathbb{B} * \mathbb{B}) \times (\mathbb{B} * \mathbb{B})$

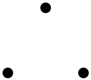
Infinite tape (TM)

Examples



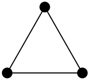
$\mathbb{Z}^3$

Blind multicounter



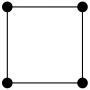
$\mathbb{B} * \mathbb{B} * \mathbb{B}$

Pushdown



$\mathbb{B}^3$

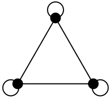
Partially blind multicounter



$(\mathbb{B} * \mathbb{B}) \times (\mathbb{B} * \mathbb{B})$

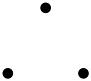
Infinite tape (TM)

Examples



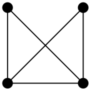
$\mathbb{Z}^3$

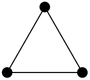
Blind multicounter



$\mathbb{B} * \mathbb{B} * \mathbb{B}$

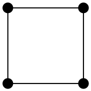
Pushdown





$\mathbb{B}^3$

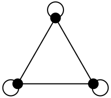
Partially blind multicounter



$(\mathbb{B} * \mathbb{B}) \times (\mathbb{B} * \mathbb{B})$

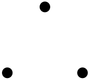
Infinite tape (TM)

Examples



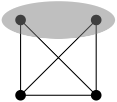
$\mathbb{Z}^3$

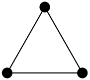
Blind multicounter



$\mathbb{B} * \mathbb{B} * \mathbb{B}$

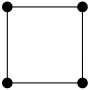
Pushdown





$\mathbb{B}^3$

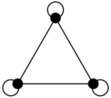
Partially blind multicounter



$(\mathbb{B} * \mathbb{B}) \times (\mathbb{B} * \mathbb{B})$

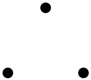
Infinite tape (TM)

Examples



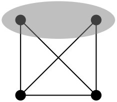
$\mathbb{Z}^3$

Blind multicounter

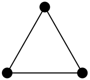


$\mathbb{B} * \mathbb{B} * \mathbb{B}$

Pushdown

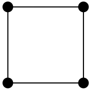


$(\mathbb{B} * \mathbb{B}) \times \mathbb{B} \times \mathbb{B}$



$\mathbb{B}^3$

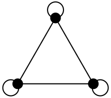
Partially blind multicounter



$(\mathbb{B} * \mathbb{B}) \times (\mathbb{B} * \mathbb{B})$

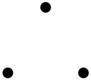
Infinite tape (TM)

Examples



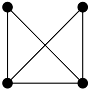
$\mathbb{Z}^3$

Blind multicounter



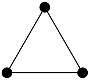
$\mathbb{B} * \mathbb{B} * \mathbb{B}$

Pushdown



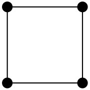
$(\mathbb{B} * \mathbb{B}) \times \mathbb{B} \times \mathbb{B}$

Pushdown + partially blind counters



$\mathbb{B}^3$

Partially blind multicounter



$(\mathbb{B} * \mathbb{B}) \times (\mathbb{B} * \mathbb{B})$

Infinite tape (TM)

Semilinearity I

For which monoids M are all languages in $\mathsf{VA}(M)$ semilinear?

- Parikh's Theorem: Pushdown storage
- Ibarra + Greibach: Blind counter automata

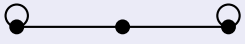
Semilinearity I

For which monoids M are all languages in $\mathsf{VA}(M)$ semilinear?

- Parikh's Theorem: Pushdown storage
- Ibarra + Greibach: Blind counter automata

Theorem (Buckheister, Z. 2013)

All languages in $\mathsf{VA}(\mathbb{M}\Gamma)$ *are semilinear if and only if*

1. Γ *contains neither*  *nor*  *as an induced subgraph and*
2. Γ*'s underlying loop-free graph is a transitive forest.*

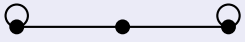
Semilinearity I

For which monoids M are all languages in $\mathsf{VA}(M)$ semilinear?

- Parikh's Theorem: Pushdown storage
- Ibarra + Greibach: Blind counter automata

Theorem (Buckheister, Z. 2013)

All languages in $\mathsf{VA}(\mathbb{M}\Gamma)$ *are semilinear if and only if*

1. Γ *contains neither*  *nor*  *as an induced subgraph and*
2. Γ*'s underlying loop-free graph is a transitive forest.*

- For 4 forbidden induced subgraphs, non-semilinear languages from Petri net and trace theory

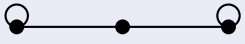
Semilinearity I

For which monoids M are all languages in $\mathsf{VA}(M)$ semilinear?

- Parikh's Theorem: Pushdown storage
- Ibarra + Greibach: Blind counter automata

Theorem (Buckheister, Z. 2013)

All languages in $\mathsf{VA}(\mathbb{M}\Gamma)$ *are semilinear if and only if*

1. Γ *contains neither*  *nor*  *as an induced subgraph and*
2. Γ*'s underlying loop-free graph is a transitive forest.*

- For 4 forbidden induced subgraphs, non-semilinear languages from Petri net and trace theory
- $\mathsf{VA}(\mathbb{B}) \subseteq \mathsf{CF}$
- $M \mapsto M \times \mathbb{Z}$, $(M, M') \mapsto M * M'$ preserve semilinearity

Semilinearity II

A group G is called a *torsion group* if for every $g \in G$, there is a $k \in \mathbb{N} \setminus \{0\}$ with $g^k = 1$.

Theorem (Render 2010)

For every monoid M, at least one of the following holds:

- $\mathsf{VA}(M) = \mathsf{REG}$
- $\mathsf{VA}(M) = \mathsf{VA}(G)$ *for a torsion group G (which is not locally finite)*
- $\mathsf{VA}(M)$ *contains the blind one-counter languages*
- $\mathsf{VA}(M)$ *contains the partially blind one-counter languages*

Semilinearity II

A group G is called a *torsion group* if for every $g \in G$, there is a $k \in \mathbb{N}\setminus\{0\}$ with $g^k = 1$.

Theorem (Render 2010)

For every monoid M, at least one of the following holds:

- $\mathsf{VA}(M) = \mathsf{REG}$
- $\mathsf{VA}(M) = \mathsf{VA}(G)$ *for a torsion group G (which is not locally finite)*
- $\mathsf{VA}(M)$ *contains the blind one-counter languages*
- $\mathsf{VA}(M)$ *contains the partially blind one-counter languages*

Theorem (Buckheister, Z. 2013)

For every torsion group G, all languages in $\mathsf{VA}(G)$ are semilinear.

Semilinearity II

A group G is called a *torsion group* if for every $g \in G$, there is a $k \in \mathbb{N}\setminus\{0\}$ with $g^k = 1$.

Theorem (Render 2010)

For every monoid M, at least one of the following holds:

- $\mathsf{VA}(M) = \mathsf{REG}$
- $\mathsf{VA}(M) = \mathsf{VA}(G)$ *for a torsion group G (which is not locally finite)*
- $\mathsf{VA}(M)$ *contains the blind one-counter languages*
- $\mathsf{VA}(M)$ *contains the partially blind one-counter languages*

Theorem (Buckheister, Z. 2013)

For every torsion group G, all languages in $\mathsf{VA}(G)$ are semilinear.

- Non-effective construction!

Semilinearity II

A group G is called a *torsion group* if for every $g \in G$, there is a $k \in \mathbb{N} \setminus \{0\}$ with $g^k = 1$.

Theorem (Render 2010)

For every monoid M, at least one of the following holds:

- $\mathsf{VA}(M) = \mathsf{REG}$
- $\mathsf{VA}(M) = \mathsf{VA}(G)$ *for a torsion group G (which is not locally finite)*
- $\mathsf{VA}(M)$ *contains the blind one-counter languages*
- $\mathsf{VA}(M)$ *contains the partially blind one-counter languages*

Theorem (Buckheister, Z. 2013)

For every torsion group G, all languages in $\mathsf{VA}(G)$ are semilinear.

- Non-effective construction!
- Decompose computations into loops (and rest of bounded length).

Semilinearity II

A group G is called a *torsion group* if for every $g \in G$, there is a $k \in \mathbb{N}\setminus\{0\}$ with $g^k = 1$.

Theorem (Render 2010)

For every monoid M, at least one of the following holds:

- $\mathsf{VA}(M) = \mathsf{REG}$
- $\mathsf{VA}(M) = \mathsf{VA}(G)$ *for a torsion group G (which is not locally finite)*
- $\mathsf{VA}(M)$ *contains the blind one-counter languages*
- $\mathsf{VA}(M)$ *contains the partially blind one-counter languages*

Theorem (Buckheister, Z. 2013)

For every torsion group G, all languages in $\mathsf{VA}(G)$ are semilinear.

- Non-effective construction!
- Decompose computations into loops (and rest of bounded length).
- Set of vectors counting loops is upward-closed w.r.t. some WQO.

Semilinearity II

Theorem (Buckheister, Z. 2013)

For every monoid M, at least one of the following holds:

- *All languages in* $\mathrm{VA}(M)$ *are semilinear.*
- $\mathrm{VA}(M)$ *contains the blind one-counter languages.*
- $\mathrm{VA}(M)$ *contains the partially blind one-counter languages.*

Semilinearity II

Theorem (Buckheister, Z. 2013)

For every monoid M, at least one of the following holds:

- *All languages in* $\mathrm{VA}(M)$ *are semilinear.*
- $\mathrm{VA}(M)$ *contains the blind one-counter languages.*
- $\mathrm{VA}(M)$ *contains the partially blind one-counter languages.*

Observation

$\mathrm{VA}(M)$ is always a full semi-AFL (closed under rational transductions, union).

Semilinearity II

Theorem (Buckheister, Z. 2013)

For every monoid M, at least one of the following holds:

- *All languages in* $\mathsf{VA}(M)$ *are semilinear.*
- $\mathsf{VA}(M)$ *contains the blind one-counter languages.*
- $\mathsf{VA}(M)$ *contains the partially blind one-counter languages.*

Observation

$\mathsf{VA}(M)$ is always a full semi-AFL (closed under rational transductions, union).

Consequence

Let $S = \{a^n \mid n \text{ is a square}\}$. Let $\mathcal{T}$ be the smallest full semi-AFL containing S. Then $\mathcal{T}$ does not arise as $\mathsf{VA}(M)$ from a monoid M.

Semilinearity II

Theorem (Buckheister, Z. 2013)

For every monoid M, at least one of the following holds:

- *All languages in* $\mathsf{VA}(M)$ *are semilinear.*
- $\mathsf{VA}(M)$ *contains the blind one-counter languages.*
- $\mathsf{VA}(M)$ *contains the partially blind one-counter languages.*

Observation

$\mathsf{VA}(M)$ is always a full semi-AFL (closed under rational transductions, union).

Consequence

Let $S = \{a^n \mid n \text{ is a square}\}$. Let $\mathcal{T}$ be the smallest full semi-AFL containing S. Then $\mathcal{T}$ does not arise as $\mathsf{VA}(M)$ from a monoid M.

- S is not semilinear
- $\mathcal{T}$ does not contain $\{a^n b^n \mid n \geqslant 0\}$.

Context-Freeness

For which monoids M is $\mathsf{VA}(M) \subseteq \mathsf{CF}$?

Context-Freeness

For which monoids M is $\mathsf{VA}(M) \subseteq \mathsf{CF}$?

Theorem (Muller, Schupp, Dunwoody)

For finitely generated groups G, $\mathsf{VA}(G) \subseteq \mathsf{CF}$ iff G is virtually free.

Context-Freeness

For which monoids M is $\mathsf{VA}(M) \subseteq \mathsf{CF}$?

Theorem (Muller, Schupp, Dunwoody)

For finitely generated groups G, $\mathsf{VA}(G) \subseteq \mathsf{CF}$ iff G is virtually free.

Theorem (Lohrey, Sénizergues 2007)

Let $\Gamma = (V, E)$ and let $G_v \neq \{1\}$ be a f.g. group for any $v \in V$.
$G = \mathbb{M}(\Gamma, (G_v)_{v \in V})$ is virtually free iff

1. *for each $v \in V$, G_v is virtually free,*

Context-Freeness

For which monoids M is $\mathsf{VA}(M) \subseteq \mathsf{CF}$?

Theorem (Muller, Schupp, Dunwoody)

For finitely generated groups G, $\mathsf{VA}(G) \subseteq \mathsf{CF}$ iff G is virtually free.

Theorem (Lohrey, Sénizergues 2007)

Let $\Gamma = (V, E)$ and let $G_v \neq \{1\}$ be a f.g. group for any $v \in V$.
$G = \mathbb{M}(\Gamma, (G_v)_{v \in V})$ is virtually free iff

1. *for each $v \in V$, G_v is virtually free,*
2. *if G_v and G_w are infinite and $v \neq w$, then $\{v, w\} \notin E$,*

Context-Freeness

For which monoids M is $\mathrm{VA}(M) \subseteq \mathrm{CF}$?

Theorem (Muller, Schupp, Dunwoody)

For finitely generated groups G, $\mathrm{VA}(G) \subseteq \mathrm{CF}$ iff G is virtually free.

Theorem (Lohrey, Sénizergues 2007)

Let $\Gamma = (V, E)$ and let $G_v \neq \{1\}$ be a f.g. group for any $v \in V$. $G = \mathbb{M}(\Gamma, (G_v)_{v \in V})$ is virtually free iff

1. *for each $v \in V$, G_v is virtually free,*
2. *if G_v and G_w are infinite and $v \neq w$, then $\{v, w\} \notin E$,*
3. *if G_v is infinite, G_u and G_w are finite and $\{v, u\}, \{v, w\} \in E$, then $\{u, w\} \in E$, and*

Context-Freeness

For which monoids M is $\mathsf{VA}(M) \subseteq \mathsf{CF}$?

Theorem (Muller, Schupp, Dunwoody)

For finitely generated groups G, $\mathsf{VA}(G) \subseteq \mathsf{CF}$ iff G is virtually free.

Theorem (Lohrey, Sénizergues 2007)

Let $\Gamma = (V, E)$ and let $G_v \neq \{1\}$ be a f.g. group for any $v \in V$. $G = \mathbb{M}(\Gamma, (G_v)_{v \in V})$ is virtually free iff

1. *for each $v \in V$, G_v is virtually free,*
2. *if G_v and G_w are infinite and $v \neq w$, then $\{v, w\} \notin E$,*
3. *if G_v is infinite, G_u and G_w are finite and $\{v, u\}, \{v, w\} \in E$, then $\{u, w\} \in E$, and*
4. *the graph Γ is chordal.*

Context-Freeness

Theorem (Buckheister, Z. 2013)

Let $\Gamma = (V, E)$ *and let* $\mathsf{J}(M_v) \neq \{1\}$ *for any* $v \in V$. $M = \mathbb{M}(\Gamma, (M_v)_{v \in V})$ *is context-free iff*

1. *for each* $v \in V$, M_v *is context-free,*

Context-Freeness

Theorem (Buckheister, Z. 2013)

Let $\Gamma = (V, E)$ *and let* $\mathsf{J}(M_v) \neq \{1\}$ *for any* $v \in V$. $M = \mathbb{M}(\Gamma, (M_v)_{v \in V})$ *is context-free iff*

1. *for each* $v \in V$, M_v *is context-free,*
2. *if* M_v *and* M_w *are not FRI-monoids and* $v \neq w$, *then* $\{v, w\} \notin E$,

Context-Freeness

Theorem (Buckheister, Z. 2013)

Let $\Gamma = (V, E)$ *and let* $\mathsf{J}(M_v) \neq \{1\}$ *for any* $v \in V$. $M = \mathbb{M}(\Gamma, (M_v)_{v \in V})$ *is context-free iff*

1. *for each* $v \in V$, M_v *is context-free,*
2. *if* M_v *and* M_w *are not FRI-monoids and* $v \neq w$, *then* $\{v, w\} \notin E$,
3. *if* M_v *is not an FRI-monoid,* M_u *and* M_w *are FRI-monoids and* $\{v, u\}, \{v, w\} \in E$, *then* $\{u, w\} \in E$, *and*

Context-Freeness

Theorem (Buckheister, Z. 2013)

Let $\Gamma = (V, E)$ *and let* $\mathsf{J}(M_v) \neq \{1\}$ *for any* $v \in V$. $M = \mathbb{M}(\Gamma, (M_v)_{v \in V})$ *is context-free iff*

1. *for each* $v \in V$, M_v *is context-free,*
2. *if* M_v *and* M_w *are not FRI-monoids and* $v \neq w$, *then* $\{v, w\} \notin E$,
3. *if* M_v *is not an FRI-monoid,* M_u *and* M_w *are FRI-monoids and* $\{v, u\}, \{v, w\} \in E$, *then* $\{u, w\} \in E$, *and*
4. *the graph* Γ *is chordal.*

Context-Freeness

Theorem (Buckheister, Z. 2013)

Let $\Gamma = (V, E)$ *and let* $\mathrm{J}(M_v) \neq \{1\}$ *for any* $v \in V$. $M = \mathbb{M}(\Gamma, (M_v)_{v \in V})$ *is context-free iff*

1. *for each* $v \in V$, M_v *is context-free,*
2. *if* M_v *and* M_w *are not FRI-monoids and* $v \neq w$, *then* $\{v, w\} \notin E$,
3. *if* M_v *is not an FRI-monoid,* M_u *and* M_w *are FRI-monoids and* $\{v, u\}, \{v, w\} \in E$, *then* $\{u, w\} \in E$, *and*
4. *the graph* Γ *is chordal.*

- $M \times M'$ context-free iff one factor is FRI and one is context-free.

Context-Freeness

Theorem (Buckheister, Z. 2013)

Let $\Gamma = (V, E)$ *and let* $\mathsf{J}(M_v) \neq \{1\}$ *for any* $v \in V$. $M = \mathbb{M}(\Gamma, (M_v)_{v \in V})$ *is context-free iff*

1. *for each* $v \in V$, M_v *is context-free,*
2. *if* M_v *and* M_w *are not FRI-monoids and* $v \neq w$, *then* $\{v, w\} \notin E$,
3. *if* M_v *is not an FRI-monoid,* M_u *and* M_w *are FRI-monoids and* $\{v, u\}, \{v, w\} \in E$, *then* $\{u, w\} \in E$, *and*
4. *the graph* Γ *is chordal.*

- $M \times M'$ context-free iff one factor is FRI and one is context-free.
- Context-freeness is preserved by products $M *_F M'$, F a finite subgroup with $1 \in F$.

Context-Freeness

Theorem (Buckheister, Z. 2013)

Let $\Gamma = (V, E)$ *and let* $\mathsf{J}(M_v) \neq \{1\}$ *for any* $v \in V$. $M = \mathbb{M}(\Gamma, (M_v)_{v \in V})$ *is context-free iff*

1. *for each* $v \in V$, M_v *is context-free,*
2. *if* M_v *and* M_w *are not FRI-monoids and* $v \neq w$*, then* $\{v, w\} \notin E$,
3. *if* M_v *is not an FRI-monoid,* M_u *and* M_w *are FRI-monoids and* $\{v, u\}, \{v, w\} \in E$*, then* $\{u, w\} \in E$*, and*
4. *the graph* Γ *is chordal.*

- $M \times M'$ context-free iff one factor is FRI and one is context-free.
- Context-freeness is preserved by products $M *_F M'$, F a finite subgroup with $1 \in F$.
- For induced cycles of finite groups, relied on Lohrey and Sénizergues' result.

Conclusion

- Regularity

Conclusion

- Regularity
- Semilinearity for graph products of $\mathbb{B}$ and $\mathbb{Z}$

Conclusion

- Regularity
- Semilinearity for graph products of $\mathbb{B}$ and $\mathbb{Z}$
- Semilinearity for torsion groups

Conclusion

- Regularity
- Semilinearity for graph products of $\mathbb{B}$ and $\mathbb{Z}$
- Semilinearity for torsion groups
- Context-freeness for graph products

Conclusion

- Regularity
- Semilinearity for graph products of $\mathbb{B}$ and $\mathbb{Z}$
- Semilinearity for torsion groups
- Context-freeness for graph products
- Language classes arising as $\mathsf{VA}(M)$

Conclusion

- Regularity
- Semilinearity for graph products of $\mathbb{B}$ and $\mathbb{Z}$
- Semilinearity for torsion groups
- Context-freeness for graph products
- Language classes arising as $\mathsf{VA}(M)$

More classical results can be generalized:

Ongoing work

- Computability of the downward closure (scattered subwords)?

Conclusion

- Regularity
- Semilinearity for graph products of $\mathbb{B}$ and $\mathbb{Z}$
- Semilinearity for torsion groups
- Context-freeness for graph products
- Language classes arising as $\mathrm{VA}(M)$

More classical results can be generalized:

Ongoing work

- Computability of the downward closure (scattered subwords)?
- Decidability of questions for Büchi variants.

Conclusion

- Regularity
- Semilinearity for graph products of $\mathbb{B}$ and $\mathbb{Z}$
- Semilinearity for torsion groups
- Context-freeness for graph products
- Language classes arising as $\mathsf{VA}(M)$

More classical results can be generalized:

Ongoing work

- Computability of the downward closure (scattered subwords)?
- Decidability of questions for Büchi variants.
- Boolean closure