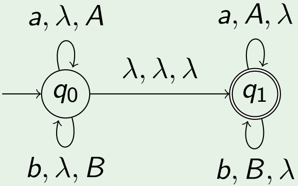
Valence automata as a generalization of automata with storage

Georg Zetzsche

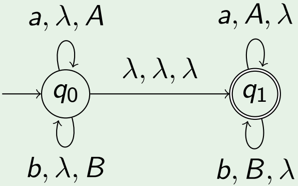
Technische Universität Kaiserslautern

D-CON 2013

Example (Pushdown automaton)

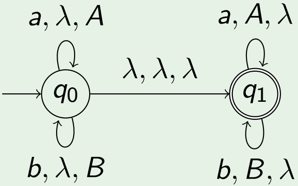


Example (Pushdown automaton)



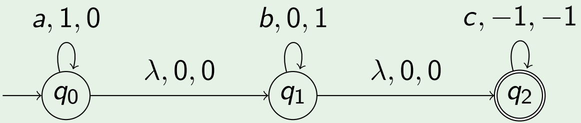
$$L = \{ww^{\text{rev}} \mid w \in \{a, b\}^*\}$$

Example (Pushdown automaton)

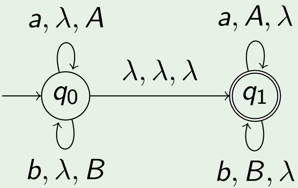


$$L = \{ww^{\text{rev}} \mid w \in \{a, b\}^*\}$$

Example (Blind counter automaton)

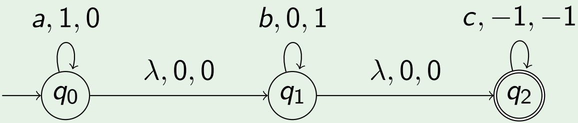


Example (Pushdown automaton)



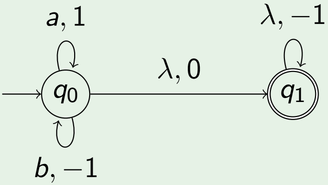
$$L = \{ww^{\text{rev}} \mid w \in \{a, b\}^*\}$$

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	$S = \Gamma^*$	$\mathrm{push}_a : w \mapsto wa,\ a \in \Gamma$ $\mathrm{pop}_a : wa \mapsto w,\ a \in \Gamma$

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
  - $m_1 \cdots m_n = 1$.

Classical results can now be generalized:

Questions

- Which storage mechanisms increase the expressive power?

Classical results can now be generalized:

Questions

- Which storage mechanisms increase the expressive power?
- For which storage mechanisms can we determinize?

Classical results can now be generalized:

Questions

- Which storage mechanisms increase the expressive power?
- For which storage mechanisms can we determinize?
- For which can we avoid silent transitions?

Classical results can now be generalized:

Questions

- Which storage mechanisms increase the expressive power?
- For which storage mechanisms can we determinize?
- For which can we avoid silent transitions?
- For which do we have a Parikh theorem?

Classical results can now be generalized:

Questions

- Which storage mechanisms increase the expressive power?
- For which storage mechanisms can we determinize?
- For which can we avoid silent transitions?
- For which do we have a Parikh theorem?
- For which can we compute the downward closure?

Classical results can now be generalized:

Questions

- Which storage mechanisms increase the expressive power?
- For which storage mechanisms can we determinize?
- For which can we avoid silent transitions?
- For which do we have a Parikh theorem?
- For which can we compute the downward closure?
- For which can we decide, for example, emptiness?

Classical results can now be generalized:

Questions

- Which storage mechanisms increase the expressive power?
- For which storage mechanisms can we determinize?
- For which can we avoid silent transitions?
- For which do we have a Parikh theorem?
- For which can we compute the downward closure?
- For which can we decide, for example, emptiness?

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 $m \in M$, $q \in Q$.

$\mathrm{detVA}(M)$ 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

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

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.

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

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

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

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

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

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

Lemma (Dichotomy)

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

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

Silent transitions

Definition

Transitions $p \xrightarrow{\lambda|m} q$ are called *silent* or λ-*transitions*.

Silent transitions

Definition

Transitions $p \xrightarrow{\lambda|m} q$ are called *silent* or λ-*transitions*.

$\mathsf{VA}^+(M)$ Languages accepted without λ-transitions.

Silent transitions

Definition

Transitions $p \xrightarrow{\lambda|m} q$ are called *silent* or λ-*transitions*.

$\mathsf{VA}^+(M)$ Languages accepted without λ-transitions.

Important problem

- When can λ-transitions be eliminated?
- Without λ-transitions, decide membership using exponential number of queries to the word problem.
- Elimination can be regarded as a precomputation.

Monoids defined by graphs

By graphs, we mean undirected graphs with loops allowed.

Monoids defined by graphs

By graphs, we mean undirected graphs with loops allowed.

Notation

- $\mathbb{B}$: monoid for partially blind counter
- $\mathbb{Z}$: monoid for blind counter, i.e. the group of integers

Monoids defined by graphs

By graphs, we mean undirected graphs with loops allowed.

Notation

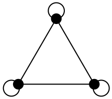
- $\mathbb{B}$: monoid for partially blind counter
- $\mathbb{Z}$: monoid for blind counter, i.e. the group of integers

Intuition

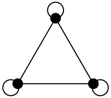
To each graph Γ, we associate a monoid $\mathbb{M}\Gamma$:

- For each unlooped vertex, we have a copy of $\mathbb{B}$
- For each looped vertex, we have a copy of $\mathbb{Z}$
- $\mathbb{M}\Gamma$ consists of sequences of such elements
- An edge between vertices means that elements can commute

Examples

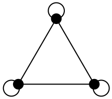


Examples



$\mathbb{Z}^3$

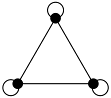
Examples



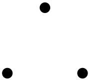
$\mathbb{Z}^3$

Blind multicounter

Examples

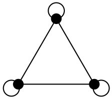


$\mathbb{Z}^3$

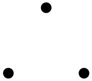


Blind multicounter

Examples



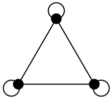
$\mathbb{Z}^3$



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

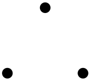
Blind multicounter

Examples



$\mathbb{Z}^3$

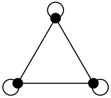
Blind multicounter



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

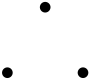
Pushdown

Examples



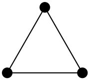
$\mathbb{Z}^3$

Blind multicounter

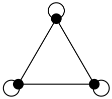


$\mathbb{B} * \mathbb{B} * \mathbb{B}$

Pushdown

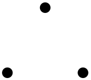


Examples



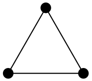
$\mathbb{Z}^3$

Blind multicounter



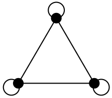
$\mathbb{B} * \mathbb{B} * \mathbb{B}$

Pushdown



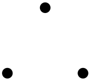
$\mathbb{B}^3$

Examples



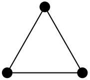
$\mathbb{Z}^3$

Blind multicounter



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

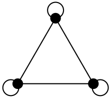
Pushdown



$\mathbb{B}^3$

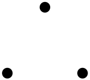
Partially blind multicounter

Examples



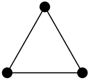
$\mathbb{Z}^3$

Blind multicounter



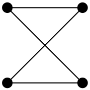
$\mathbb{B} * \mathbb{B} * \mathbb{B}$

Pushdown

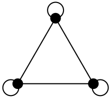


$\mathbb{B}^3$

Partially blind multicounter

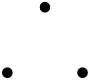


Examples



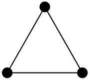
$\mathbb{Z}^3$

Blind multicounter



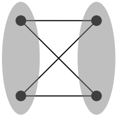
$\mathbb{B} * \mathbb{B} * \mathbb{B}$

Pushdown

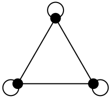


$\mathbb{B}^3$

Partially blind multicounter

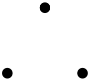


Examples



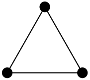
$\mathbb{Z}^3$

Blind multicounter



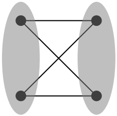
$\mathbb{B} * \mathbb{B} * \mathbb{B}$

Pushdown



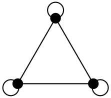
$\mathbb{B}^3$

Partially blind multicounter



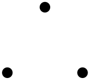
$(\mathbb{B} * \mathbb{B}) \times (\mathbb{B} * \mathbb{B})$

Examples



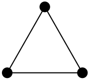
$\mathbb{Z}^3$

Blind multicounter



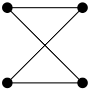
$\mathbb{B} * \mathbb{B} * \mathbb{B}$

Pushdown



$\mathbb{B}^3$

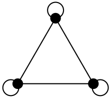
Partially blind multicounter



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

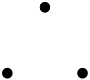
Infinite tape (TM)

Examples



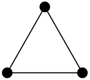
$\mathbb{Z}^3$

Blind multicounter



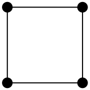
$\mathbb{B} * \mathbb{B} * \mathbb{B}$

Pushdown



$\mathbb{B}^3$

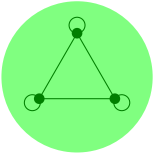
Partially blind multicounter



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

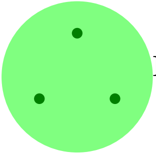
Infinite tape (TM)

Examples



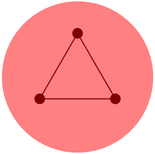
$\mathbb{Z}^3$

Blind multicounter



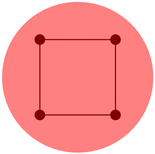
$\mathbb{B} * \mathbb{B} * \mathbb{B}$

Pushdown



$\mathbb{B}^3$

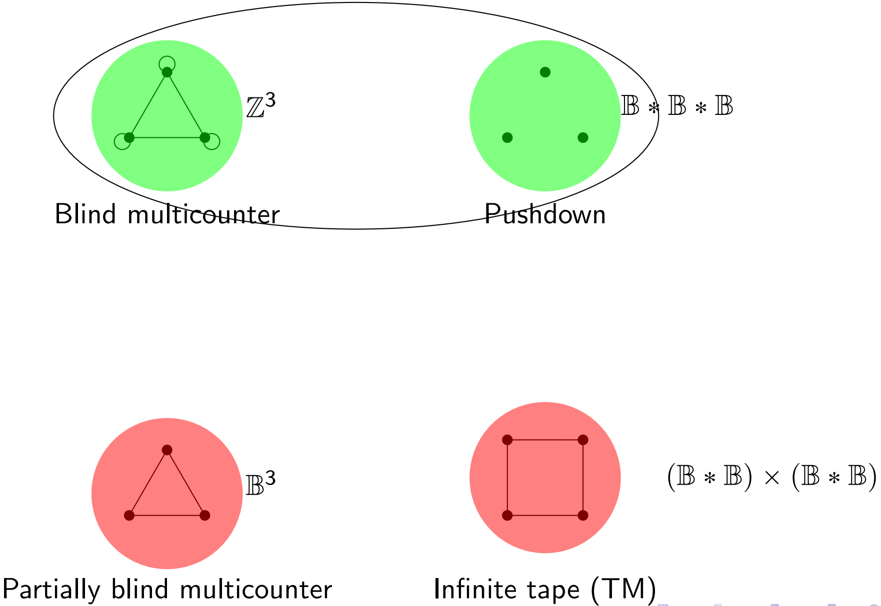
Partially blind multicounter



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

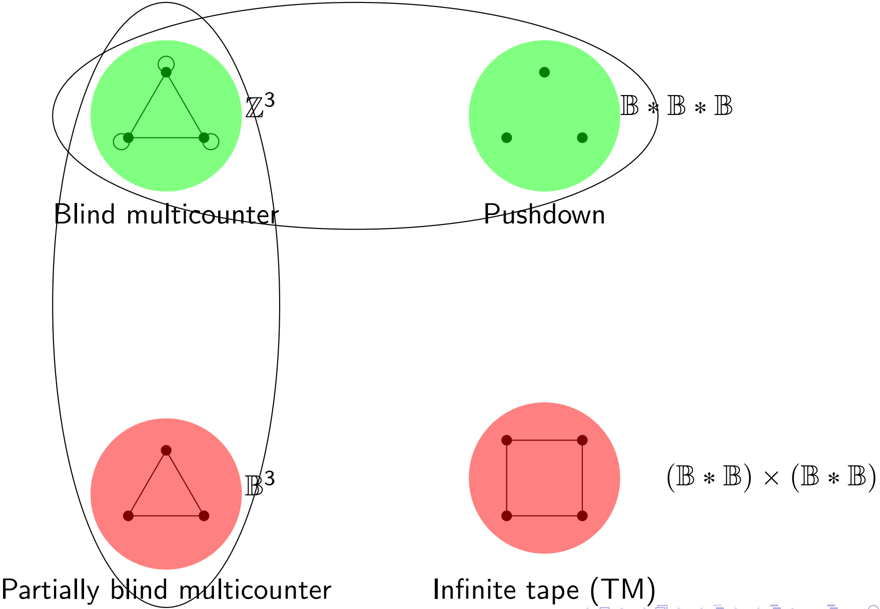
Infinite tape (TM)

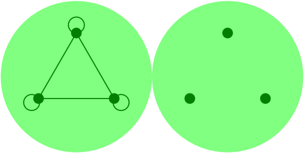
Examples



$\mathbb{Z}^3$
$\mathbb{B} * \mathbb{B} * \mathbb{B}$
Blind multicounter
Pushdown
$\mathbb{B}^3$
$(\mathbb{B} * \mathbb{B}) \times (\mathbb{B} * \mathbb{B})$
Partially blind multicounter
Infinite tape (TM)

Examples

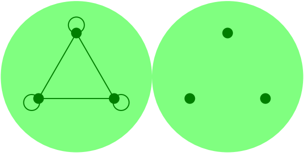

$\mathbb{Z}^3$
Blind multicounter
$\mathbb{B} * \mathbb{B} * \mathbb{B}$
Pushdown
$\mathbb{B}^3$
Partially blind multicounter
$(\mathbb{B} * \mathbb{B}) \times (\mathbb{B} * \mathbb{B})$
Infinite tape (TM)

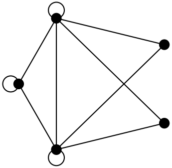


Theorem

Let Γ be a graph such that

- *between any two looped vertices, there is an edge and*
- *between any two unlooped vertices, there is no edge.*

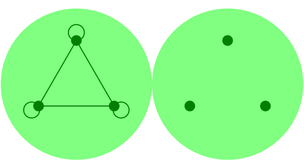


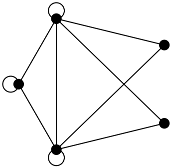


Theorem

Let Γ be a graph such that

- *between any two looped vertices, there is an edge and*
- *between any two unlooped vertices, there is no edge.*



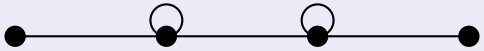


Theorem

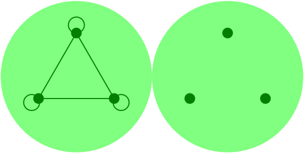
Let Γ be a graph such that

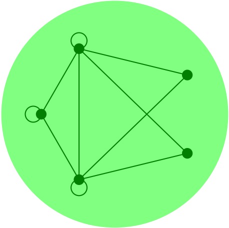
- *between any two looped vertices, there is an edge and*
- *between any two unlooped vertices, there is no edge.*

Then $\mathsf{VA}(\mathbb{M}\Gamma) = \mathsf{VA}^{+}(\mathbb{M}\Gamma)$ *if and only if* Γ *does not contain*



as an induced subgraph.





Theorem

Let Γ *be a graph such that*

- *between any two looped vertices, there is an edge and*
- *between any two unlooped vertices, there is no edge.*

Then $\mathsf{VA}(\mathbb{M}\Gamma) = \mathsf{VA}^{+}(\mathbb{M}\Gamma)$ *if and only if* Γ *does not contain*

as an induced subgraph.

By reduction to an undecidable problem from group theory, we obtain:

Lemma

Let Γ *be a graph with*

as an induced subgraph. Then $\mathsf{VA}(\mathbb{M}\Gamma)$ *contains an undecidable language. Hence,* $\mathsf{VA}^{+}(\mathbb{M}\Gamma) \subsetneq \mathsf{VA}(\mathbb{M}\Gamma)$.

Definition

Let $\mathcal{C}$ be the smallest class of monoids such that

- $1 \in \mathcal{C}$
- if $M \in \mathcal{C}$, then $M \times \mathbb{Z} \in \mathcal{C}$
- if $M \in \mathcal{C}$, then $M * \mathbb{B} \in \mathcal{C}$

Definition

Let $\mathcal{C}$ be the smallest class of monoids such that

- $1 \in \mathcal{C}$
- if $M \in \mathcal{C}$, then $M \times \mathbb{Z} \in \mathcal{C}$
- if $M \in \mathcal{C}$, then $M * \mathbb{B} \in \mathcal{C}$

Lemma

Let Γ be a graph such that

- *between any two looped vertices, there is an edge*
- *between any two unlooped vertices, there is no edge*
-  *does not appear as an induced subgraph*

Then, $\mathbb{M}\Gamma \in \mathcal{C}$.

Definition

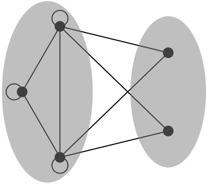
Let $\mathcal{C}$ be the smallest class of monoids such that

- $1 \in \mathcal{C}$
- if $M \in \mathcal{C}$, then $M \times \mathbb{Z} \in \mathcal{C}$
- if $M \in \mathcal{C}$, then $M * \mathbb{B} \in \mathcal{C}$

Interpretation of $\mathcal{C}$

$\mathcal{C}$ corresponds to the class of storage mechanisms obtained by

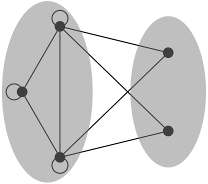
- adding a blind counter ($M \times \mathbb{Z}$) and
- building stacks ($M * \mathbb{B}$).



Proof.

- Let $\Gamma = (V, E)$, $V = L \cup U$, looped and unlooped vertices.
- For each $x \in L$, let $\nu(x) = N(x) \cap U$

$\square$

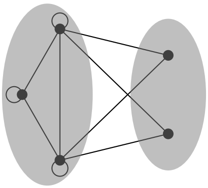


Proof.

- Let $\Gamma = (V, E)$, $V = L \cup U$, looped and unlooped vertices.
- For each $x \in L$, let $\nu(x) = N(x) \cap U$ and

$$\text{for } x, y \in L: \quad x \leqslant y \;\text{ iff }\; \nu(x) \subseteq \nu(y).$$

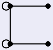
$\square$



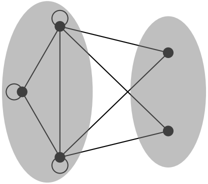
Proof.

- Let $\Gamma = (V, E)$, $V = L \cup U$, looped and unlooped vertices.
- For each $x \in L$, let $\nu(x) = N(x) \cap U$ and

$$\text{for } x, y \in L: \quad x \leqslant y \ \text{ iff } \ \nu(x) \subseteq \nu(y).$$

- Since  is not an induced subgraph, $\leqslant$ is a total preorder.

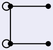
$\square$



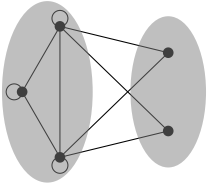
Proof.

- Let $\Gamma = (V, E)$, $V = L \cup U$, looped and unlooped vertices.
- For each $x \in L$, let $\nu(x) = N(x) \cap U$ and

$$\text{for } x, y \in L: \quad x \leqslant y \ \text{ iff } \ \nu(x) \subseteq \nu(y).$$

- Since  is not an induced subgraph, $\leqslant$ is a total preorder.
- Let $m \in L$ be maximal.

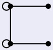
$\square$



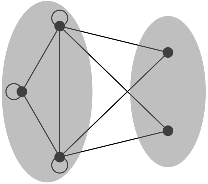
Proof.

- Let $\Gamma = (V, E)$, $V = L \cup U$, looped and unlooped vertices.
- For each $x \in L$, let $\nu(x) = N(x) \cap U$ and

$$\text{for } x, y \in L: \quad x \leqslant y \ \text{ iff } \ \nu(x) \subseteq \nu(y).$$

- Since  is not an induced subgraph, $\leqslant$ is a total preorder.
- Let $m \in L$ be maximal.
- If $\nu(m) = U$, then $\Gamma = (\Gamma \setminus \{m\}) \times m$.

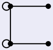
□



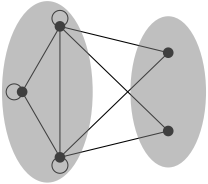
Proof.

- Let $\Gamma = (V, E)$, $V = L \cup U$, looped and unlooped vertices.
- For each $x \in L$, let $\nu(x) = N(x) \cap U$ and

$$\text{for } x, y \in L: \quad x \leqslant y \ \text{ iff } \ \nu(x) \subseteq \nu(y).$$

- Since  is not an induced subgraph, $\leqslant$ is a total preorder.
- Let $m \in L$ be maximal.
- If $\nu(m) = U$, then $\Gamma = (\Gamma \setminus \{m\}) \times m$. $\mathbb{M}\Gamma \cong \mathbb{M}(\Gamma \setminus \{m\}) \times \mathbb{Z}$.

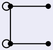
$\square$



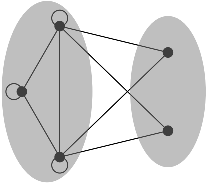
Proof.

- Let $\Gamma = (V, E)$, $V = L \cup U$, looped and unlooped vertices.
- For each $x \in L$, let $\nu(x) = N(x) \cap U$ and

$$\text{for } x, y \in L: \quad x \leqslant y \ \text{ iff } \ \nu(x) \subseteq \nu(y).$$

- Since  is not an induced subgraph, $\leqslant$ is a total preorder.
- Let $m \in L$ be maximal.
- If $\nu(m) = U$, then $\Gamma = (\Gamma \setminus \{m\}) \times m$. $\mathbb{M}\Gamma \cong \mathbb{M}(\Gamma \setminus \{m\}) \times \mathbb{Z}$.
- If $\nu(m) \subsetneq U$, then $\Gamma = (\Gamma \setminus \{u\}) * u$ for some $u \in U$.

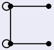
$\square$



Proof.

- Let $\Gamma = (V, E)$, $V = L \cup U$, looped and unlooped vertices.
- For each $x \in L$, let $\nu(x) = N(x) \cap U$ and

$$\text{for } x, y \in L: \quad x \leqslant y \;\text{ iff }\; \nu(x) \subseteq \nu(y).$$

- Since  is not an induced subgraph, $\leqslant$ is a total preorder.
- Let $m \in L$ be maximal.
- If $\nu(m) = U$, then $\Gamma = (\Gamma \setminus \{m\}) \times m$. $\mathbb{M}\Gamma \cong \mathbb{M}(\Gamma \setminus \{m\}) \times \mathbb{Z}$.
- If $\nu(m) \subsetneq U$, then $\Gamma = (\Gamma \setminus \{u\}) * u$ for some $u \in U$. $\mathbb{M}\Gamma \cong \mathbb{M}(\Gamma \setminus \{u\}) * \mathbb{B}$.

□

Elimination of λ-transitions

Definition

A subset $S \subseteq M$ is called *rational* if it is the homomorphic image of a regular language.

Elimination of λ-transitions

Approach:

- Between a given pair of non-λ-transitions, the set of $x \in M$ applied in between is a rational set.
- Transform the automaton so as to simulate the application of a rational set in one step.

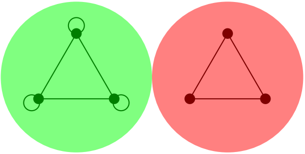
Proof ingredients

Ingredients

- Semilinearity of languages in $\mathsf{VA}(\mathbb{M}\Gamma)$
- Normal form for rational subsets of $\mathbb{M}\Gamma$

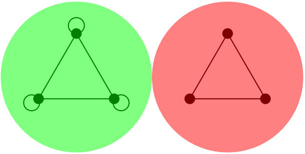
Construction for $\mathsf{VA}^+(\mathbb{M}\Gamma) = \mathsf{VA}(\mathbb{M}\Gamma)$

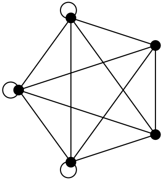
- Actual construction quite involved.
- Stronger claim to make induction work.
- Separate constructions for $\mathbb{B}$, $M \times \mathbb{Z}$, and $M * \mathbb{B}$.
- Representations of rational sets are encoded into the state or the monoid elements.



Theorem

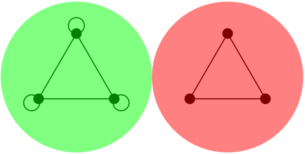
Let Γ be a graph such that between any two distinct vertices, there is an edge.

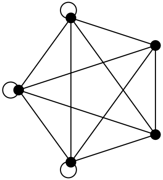




Theorem

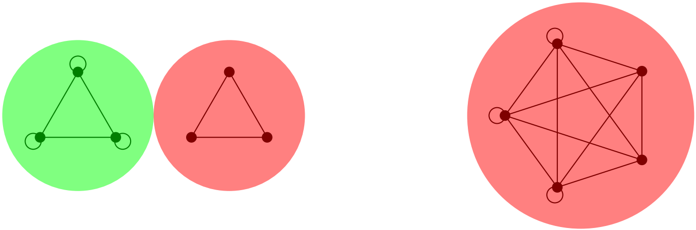
Let Γ be a graph such that between any two distinct vertices, there is an edge.





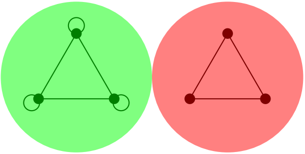
Theorem

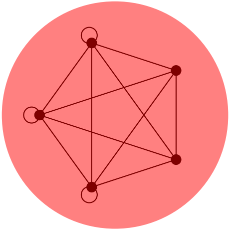
Let Γ be a graph such that between any two distinct vertices, there is an edge. Then $\mathsf{VA}(\mathbb{M}\Gamma) = \mathsf{VA}^+(\mathbb{M}\Gamma)$ if and only if the number of unlooped nodes is $\leqslant 1$.



Theorem

Let Γ *be a graph such that between any two distinct vertices, there is an edge. Then* $\mathsf{VA}(\mathbb{M}\Gamma) = \mathsf{VA}^{+}(\mathbb{M}\Gamma)$ *if and only if the number of unlooped nodes is* $\leqslant 1$.

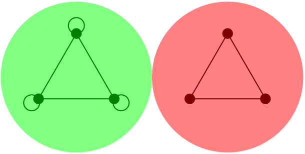


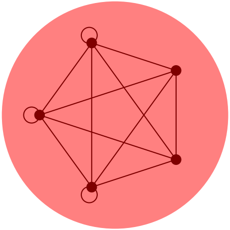


Theorem

Let Γ *be a graph such that between any two distinct vertices, there is an edge. Then* $\mathsf{VA}(\mathbb{M}\Gamma) = \mathsf{VA}^+(\mathbb{M}\Gamma)$ *if and only if the number of unlooped nodes is* $\leqslant 1$*. In other words:*

$$\mathsf{VA}(\mathbb{B}^r \times \mathbb{Z}^s) = \mathsf{VA}^+(\mathbb{B}^r \times \mathbb{Z}^s) \quad \textit{iff} \quad r \leqslant 1.$$





Theorem

Let Γ be a graph such that between any two distinct vertices, there is an edge. Then $\mathsf{VA}(\mathbb{M}\Gamma) = \mathsf{VA}^+(\mathbb{M}\Gamma)$ if and only if the number of unlooped nodes is $\leqslant 1$. In other words:

$$\mathsf{VA}(\mathbb{B}^r \times \mathbb{Z}^s) = \mathsf{VA}^+(\mathbb{B}^r \times \mathbb{Z}^s) \quad \textit{iff} \quad r \leqslant 1.$$

Observation

$\mathsf{VA}(\mathbb{B} \times \mathbb{Z}^s) = \mathsf{VA}^+(\mathbb{B} \times \mathbb{Z}^s)$ already follows from the first theorem.

More classical results can be generalized:

Work in Progress

- For which storage mechanisms do we have a Parikh theorem?

More classical results can be generalized:

Work in Progress

- For which storage mechanisms do we have a Parikh theorem?
- For which can we perform model checking?

More classical results can be generalized:

Work in Progress

- For which storage mechanisms do we have a Parikh theorem?
- For which can we perform model checking?
- For which can we compute the downward closure?