First-order logic with reachability for infinite-state systems

Emanuele D'Osualdo[1] Roland Meyer[1] Georg Zetzsche[2]

[1]University of Kaiserslautern
[2]LSV, CNRS & ENS Cachan, Université Paris-Saclay

LICS 2016

Goal: decidability frontier of FO[R]

$$S_A \models \varphi$$

Goal: decidability frontier of FO[R]

$$S_A \models \varphi$$

Automaton:
finite control + storage

Goal: decidability frontier of FO[R]

Reachability structure:
infinite graph of configurations + $\to^*$

$$S_A \models \varphi$$

Automaton:
finite control + storage

Goal: decidability frontier of FO[R]

Reachability structure:
infinite graph of configurations + $\to^*$

$$S_A \models \varphi$$

Automaton:
finite control + storage

FO[R] first-order with reachability:
middle ground between FO and MSO

Goal: decidability frontier of FO[R]

Reachability structure:
infinite graph of configurations + $\to^*$

$$S_A \models \varphi$$

Automaton:
finite control + storage

FO[R] first-order with reachability:
middle ground between FO and MSO

Decidable		**Undecidable**
1-stack automata	?	2-stacks automata
2-dimension VASS		VASS (aka Petri nets)

Key Question

Which features of storage mechanisms determine the decidability of FO[R]?

Key Question

Which features of storage mechanisms determine the decidability of FO[R]?

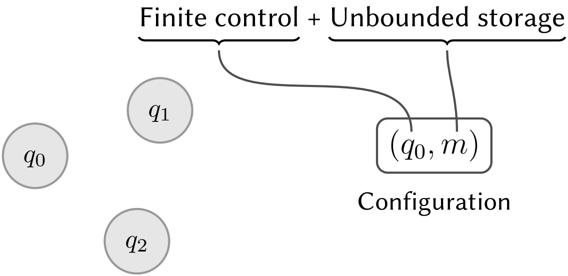
Main Result

We found a simple condition characterising storage mechanisms with decidable FO[R].

Valence Automata

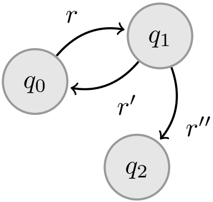
Finite control + Unbounded storage

Valence Automata



Valence Automata

Finite control + Unbounded storage

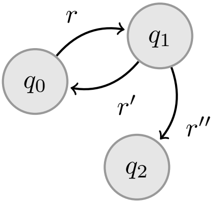


$$(q_0, m) \longrightarrow (q_1, m \odot r)$$

Apply action

Valence Automata

Finite control + Unbounded storage



$$(q_0, m) \longrightarrow (q_1, m \odot r)$$

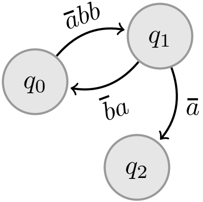
Apply action

Storage = Monoid $(M, \odot, \mathbf{1})$

- $m, r \in M$ set of storage contents and actions
- valid storage contents are the right-invertible elements:
  $$m \in \mathcal{R}_1(M) := \{x \in M \mid \exists r \in M : x \odot r = \mathbf{1}\}$$
- $\mathbf{1}$ is empty storage and no-op action

Valence Automata

Pushdown systems

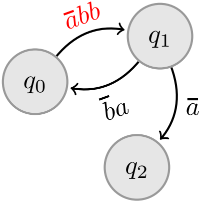


(q_0, ba)

Stack = Monoid $\left(\{a, b, \bar{a}, \bar{b}\}^*, \odot, \varepsilon\right)$

- a, b are push actions, $\bar{a}, \bar{b}$ are pop actions

Pushdown systems



$$(q_0, ba) \longrightarrow (q_1, ba \odot \bar{a}bb)$$

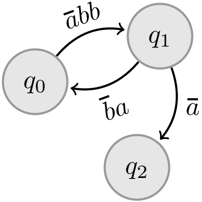
Apply action

Stack = Monoid $\left(\{a, b, \bar{a}, \bar{b}\}^*, \odot, \varepsilon\right)$

- a, b are push actions, $\bar{a}, \bar{b}$ are pop actions

Valence Automata

Pushdown systems



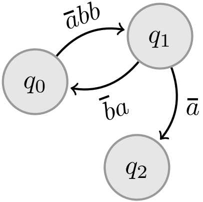
$$(q_0, ba) \quad \longrightarrow \quad (q_1, b\not{a} \odot \not{\bar{a}}bb)$$

Stack = Monoid $\left(\{a, b, \bar{a}, \bar{b}\}^*, \odot, \varepsilon\right)$

- a, b are push actions, $\bar{a}, \bar{b}$ are pop actions

Valence Automata

Pushdown systems

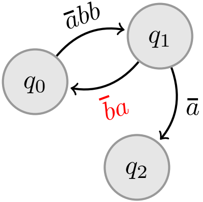


$$(q_0, ba) \longrightarrow (q_1, \quad bbb \quad)$$

Stack = Monoid $\left(\{a, b, \bar{a}, \bar{b}\}^*, \odot, \varepsilon\right)$

- a, b are push actions, $\bar{a}, \bar{b}$ are pop actions

Pushdown systems



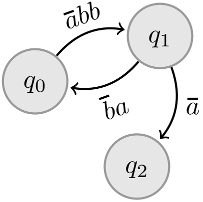
$$(q_0, ba) \longrightarrow (q_1, \quad bbb \quad)$$
$$\downarrow$$
$$(q_0, bbb \odot \bar{b}a)$$

Stack = Monoid $\left(\{a, b, \bar{a}, \bar{b}\}^*, \odot, \varepsilon\right)$

- a, b are push actions, $\bar{a}, \bar{b}$ are pop actions

Valence Automata

Pushdown systems



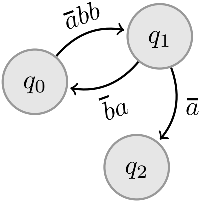
$$(q_0, ba) \longrightarrow (q_1, bbb)$$
$$\downarrow$$
$$(q_0, bba)$$

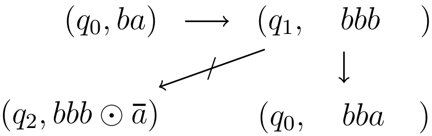
Stack = Monoid $\left(\{a, b, \bar{a}, \bar{b}\}^*, \odot, \varepsilon\right)$

- a, b are push actions, $\bar{a}, \bar{b}$ are pop actions

Valence Automata

Pushdown systems



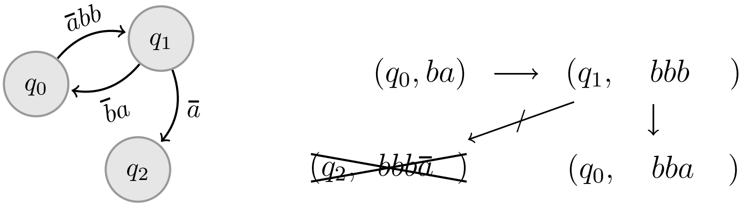


Stack = Monoid $\left(\{a, b, \bar{a}, \bar{b}\}^*, \odot, \varepsilon\right)$

- a, b are push actions, $\bar{a}, \bar{b}$ are pop actions
- valid storage contents are the right-invertible elements:

$$\mathcal{R}_1(M) = \{a, b\}^*$$

Pushdown systems



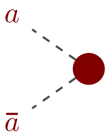
Stack = Monoid $(\{a, b, \bar{a}, \bar{b}\}^*, \odot, \varepsilon)$

- a, b are push actions, $\bar{a}, \bar{b}$ are pop actions
- valid storage contents are the right-invertible elements:
$$\mathcal{R}_1(M) = \{a, b\}^*$$
- ε is empty storage and no-op action

Graph monoids

- Graph $\Gamma = (V, E)$

Graph monoids



$a\bar{a} \equiv \varepsilon$

- Graph $\Gamma = (V, E)$
- Generators $X_\Gamma = \{a_v, \bar{a}_v \mid v \in V\}$

Graph monoids

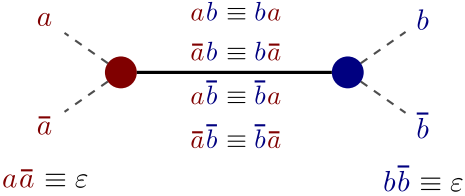
$a\bar{a} \equiv \varepsilon$ $\quad$ $b\bar{b} \equiv \varepsilon$

- Graph $\Gamma = (V, E)$
- Generators $X_\Gamma = \{a_v, \bar{a}_v \mid v \in V\}$

Graph monoids

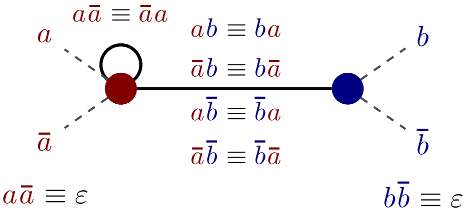
- Graph $\Gamma = (V, E)$
- Generators $X_\Gamma = \{a_v, \bar{a}_v \mid v \in V\}$
- Monoid generated by Γ is $\mathbb{M}\Gamma = X_\Gamma^* /_{\equiv}$

Graph monoids



- Graph $\Gamma = (V, E)$
- Generators $X_\Gamma = \{a_v, \bar{a}_v \mid v \in V\}$
- Monoid generated by Γ is $\mathbb{M}\Gamma = X_\Gamma^*/_\equiv$

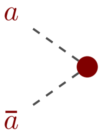
Graph monoids



- Graph $\Gamma = (V, E)$
- Generators $X_\Gamma = \{a_v, \bar{a}_v \mid v \in V\}$
- Monoid generated by Γ is $\mathbb{M}\Gamma = X_\Gamma^*/_\equiv$

Examples of Graph Monoids

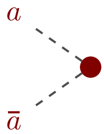
A partially blind counter: $\mathbb{B}$



$a\bar{a} \equiv \varepsilon$

Examples of Graph Monoids

A partially blind counter: $\mathbb{B}$

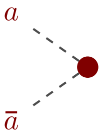


$a\bar{a} \equiv \varepsilon$

aaa

Examples of Graph Monoids

A partially blind counter: $\mathbb{B}$

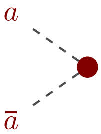


$$a\bar{a} \equiv \varepsilon$$

$$aaa \cdot \bar{a}$$

Examples of Graph Monoids

A partially blind counter: $\mathbb{B}$

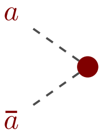


$$a\bar{a} \equiv \varepsilon$$

$$aa\cancel{a}\cancel{\bar{a}}$$

Examples of Graph Monoids

A partially blind counter: $\mathbb{B}$



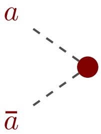
$$a\bar{a} \equiv \varepsilon$$

$$aa\cancel{a}\cancel{\bar{a}} \equiv aa$$

$$a$$

Examples of Graph Monoids

A partially blind counter: $\mathbb{B}$



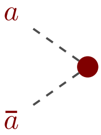
$$a\bar{a} \equiv \varepsilon$$

$$aa\cancel{a}\cancel{\bar{a}} \equiv aa$$

$$a \cdot \bar{a}\bar{a}$$

Examples of Graph Monoids

A partially blind counter: $\mathbb{B}$



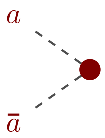
$a\bar{a} \equiv \varepsilon$

$$aa\cancel{a}\cancel{\bar{a}} \equiv aa$$

$$\cancel{a}\cancel{\bar{a}}\bar{a}$$

Examples of Graph Monoids

A partially blind counter: $\mathbb{B}$



$a\bar{a} \equiv \varepsilon$

$$aa\cancel{a}\cancel{\bar{a}} \equiv aa$$

$$\cancel{a}\cancel{\bar{a}}\bar{a} \equiv \bar{a}$$ not right-invertible!

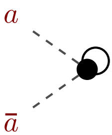
Can only represent positive integers

Examples of Graph Monoids

A blind counter: $\mathbb{Z}$

Examples of Graph Monoids

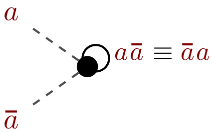
A blind counter: $\mathbb{Z}$



$a\bar{a} \equiv \varepsilon$

Examples of Graph Monoids

A blind counter: $\mathbb{Z}$

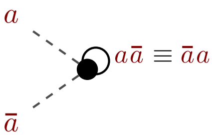


$a\bar{a} \equiv \varepsilon$

aaa

Examples of Graph Monoids

A blind counter: $\mathbb{Z}$

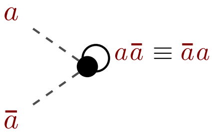


$a\bar{a} \equiv \varepsilon$

$$aaa \cdot \bar{a}$$

Examples of Graph Monoids

A blind counter: $\mathbb{Z}$

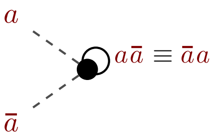


$$a\bar{a} \equiv \varepsilon$$

$$aa\cancel{a}\cancel{\bar{a}} \equiv aa$$

Examples of Graph Monoids

A blind counter: $\mathbb{Z}$



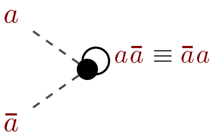
$a\bar{a} \equiv \varepsilon$

$$aa\not{a}\not{\bar{a}} \equiv aa$$

a

Examples of Graph Monoids

A blind counter: $\mathbb{Z}$



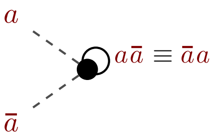
$a\bar{a} \equiv \varepsilon$

$$aa\cancel{a}\cancel{\bar{a}} \equiv aa$$

$$a \cdot \bar{a}\bar{a}$$

Examples of Graph Monoids

A blind counter: $\mathbb{Z}$



$a\bar{a} \equiv \varepsilon$

$$aa\cancel{a}\cancel{\bar{a}} \equiv aa$$

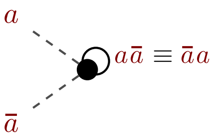
$$\cancel{a}\cancel{\bar{a}}\bar{a} \equiv \bar{a}$$

now right-invertible!

($\bar{a}a \equiv a\bar{a} \equiv \varepsilon$)

Examples of Graph Monoids

A blind counter: $\mathbb{Z}$



$a\bar{a} \equiv \varepsilon$

$$aa\cancel{a}\cancel{\bar{a}} \equiv aa$$

$$\cancel{a}\cancel{\bar{a}}\bar{a} \equiv \bar{a}$$

now right-invertible!
($\bar{a}a \equiv a\bar{a} \equiv \varepsilon$)

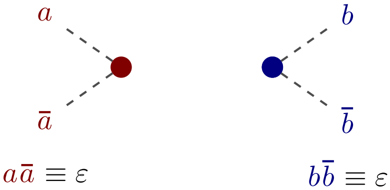
Can also represent negative integers

Examples of Graph Monoids

A stack of two symbols: $\mathbb{B} * \mathbb{B}$

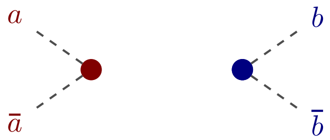
Examples of Graph Monoids

A stack of two symbols: $\mathbb{B} * \mathbb{B}$



Examples of Graph Monoids

A stack of two symbols: $\mathbb{B} * \mathbb{B}$

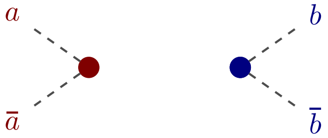


$a\bar{a} \equiv \varepsilon$ $b\bar{b} \equiv \varepsilon$

$aaab$

Examples of Graph Monoids

A stack of two symbols: $\mathbb{B} * \mathbb{B}$



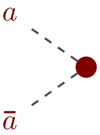
$a\bar{a} \equiv \varepsilon$

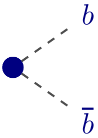
$b\bar{b} \equiv \varepsilon$

$$aaab \cdot \bar{b}\bar{a}b$$

Examples of Graph Monoids

A stack of two symbols: $\mathbb{B} * \mathbb{B}$





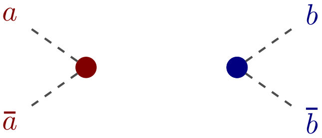
$a\bar{a} \equiv \varepsilon$

$b\bar{b} \equiv \varepsilon$

$aaa\cancel{b}\cancel{\bar{b}}\bar{a}b$

Examples of Graph Monoids

A stack of two symbols: $\mathbb{B} * \mathbb{B}$

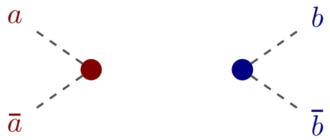


$a\bar{a} \equiv \varepsilon$ $\quad$ $b\bar{b} \equiv \varepsilon$

$$aaa\cancel{b}\cancel{\bar{b}}\bar{a}b \equiv aaa\bar{a}b$$

Examples of Graph Monoids

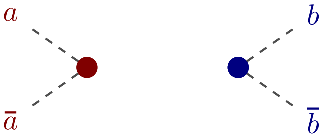
A stack of two symbols: $\mathbb{B} * \mathbb{B}$



$a\bar{a} \equiv \varepsilon$ $\quad$ $b\bar{b} \equiv \varepsilon$

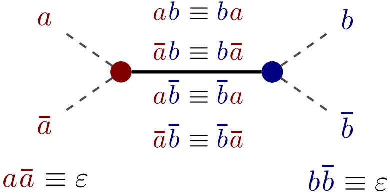
$$aaa\cancel{b}\cancel{\bar{b}}\bar{a}b \equiv aa\cancel{a}\cancel{\bar{a}}b \equiv aab$$

Examples of Graph Monoids

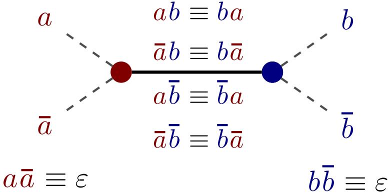


$a\bar{a} \equiv \varepsilon$ $\qquad$ $b\bar{b} \equiv \varepsilon$

Examples of Graph Monoids

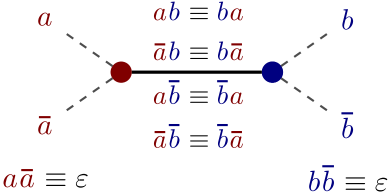


Examples of Graph Monoids



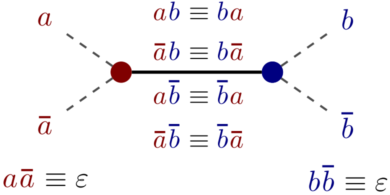
$$abaaba$$

Examples of Graph Monoids



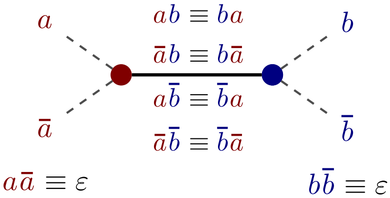
$$abaaab$$

Examples of Graph Monoids



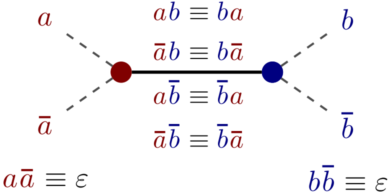
$$aabaab$$

Examples of Graph Monoids



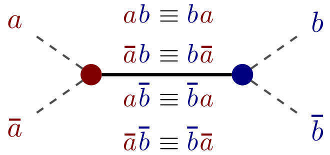
$$aaabab$$

Examples of Graph Monoids



$$aaaabb$$

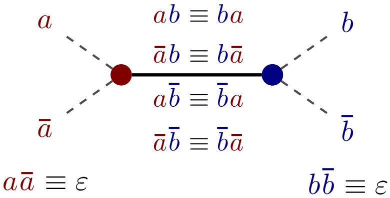
Examples of Graph Monoids



$a\bar{a} \equiv \varepsilon$ $\quad$ $b\bar{b} \equiv \varepsilon$

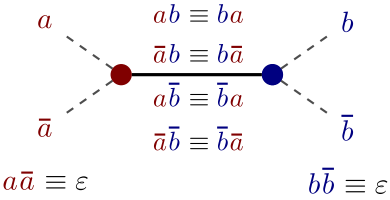
$$aaaabb \cdot \bar{a}$$

Examples of Graph Monoids



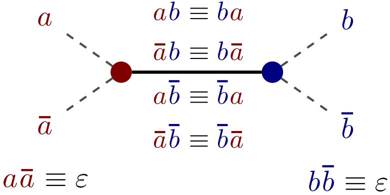
$$aaaabb\bar{a}$$

Examples of Graph Monoids



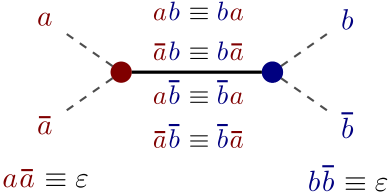
$$aaaab\bar{a}b$$

Examples of Graph Monoids



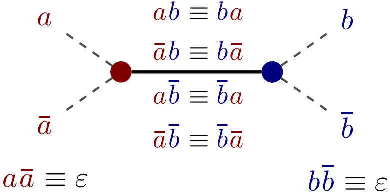
$$aaaa\bar{a}bb$$

Examples of Graph Monoids



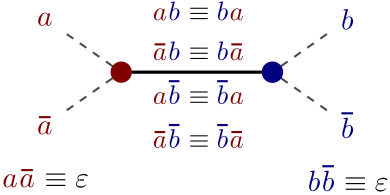
$$aaa\not{a}\not{\bar{a}}bb$$

Examples of Graph Monoids



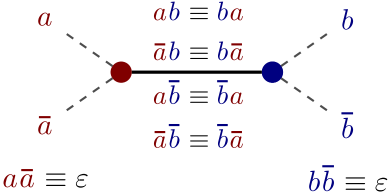
$$aaabb$$

Examples of Graph Monoids



$$aaabb \cdot \bar{b}$$

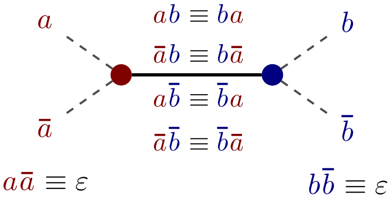
Examples of Graph Monoids



$$aaab\cancel{b}\cancel{\bar{b}}$$

Examples of Graph Monoids

Two partially blind counters: $\mathbb{B} \times \mathbb{B}$



$$aaab$$

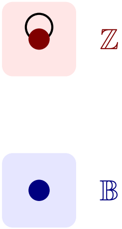
Direct product

This works in general: $\mathbb{M}\Gamma_1 \times \mathbb{M}\Gamma_2 = \mathbb{M}\Gamma_{1,2}$:

$\mathbb{Z}$

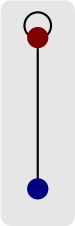
Direct product

This works in general: $\mathbb{M}\Gamma_1 \times \mathbb{M}\Gamma_2 = \mathbb{M}\Gamma_{1,2}$:



Direct product

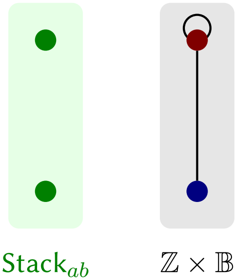
This works in general: $\mathbb{M}\Gamma_1 \times \mathbb{M}\Gamma_2 = \mathbb{M}\Gamma_{1,2}$:



$\mathbb{Z} \times \mathbb{B}$

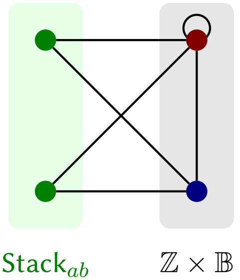
Direct product

This works in general: $\mathbb{M}\Gamma_1 \times \mathbb{M}\Gamma_2 = \mathbb{M}\Gamma_{1,2}$:



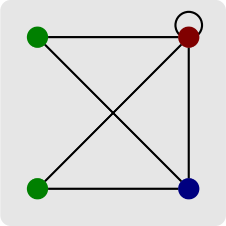
Direct product

This works in general: $\mathbb{M}\Gamma_1 \times \mathbb{M}\Gamma_2 = \mathbb{M}\Gamma_{1,2}$:



Direct product

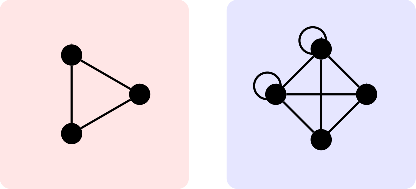
This works in general: $\mathbb{M}\Gamma_1 \times \mathbb{M}\Gamma_2 = \mathbb{M}\Gamma_{1,2}$:



$\mathsf{Stack}_{ab} \times \mathbb{Z} \times \mathbb{B}$

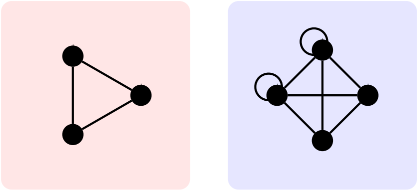
Free product

Free product: $\mathbb{M}\Gamma_1 * \mathbb{M}\Gamma_2 = \mathbb{M}(\Gamma_1 \uplus \Gamma_2)$.



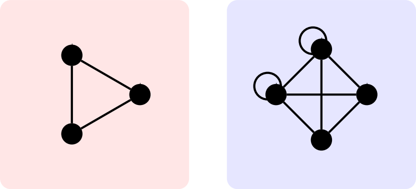
Free product

Free product: $\mathbb{M}\Gamma_1 * \mathbb{M}\Gamma_2 = \mathbb{M}(\Gamma_1 \uplus \Gamma_2)$.



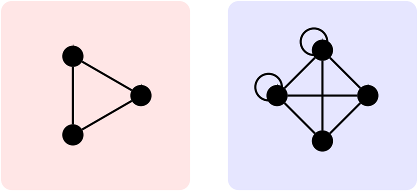
Free product

Free product: $\mathbb{M}\Gamma_1 * \mathbb{M}\Gamma_2 = \mathbb{M}(\Gamma_1 \uplus \Gamma_2)$.



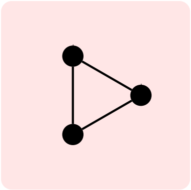
Free product

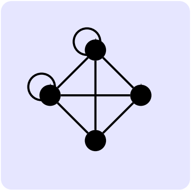
Free product: $\mathbb{M}\Gamma_1 * \mathbb{M}\Gamma_2 = \mathbb{M}(\Gamma_1 \uplus \Gamma_2)$.



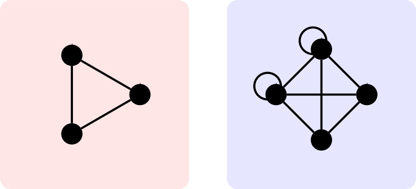
Free product

Free product: $\mathbb{M}\Gamma_1 * \mathbb{M}\Gamma_2 = \mathbb{M}(\Gamma_1 \uplus \Gamma_2)$.



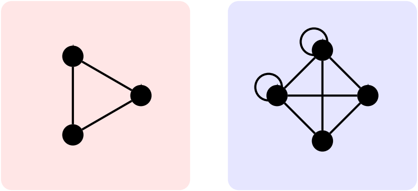


Free product: $\mathbb{M}\Gamma_1 * \mathbb{M}\Gamma_2 = \mathbb{M}(\Gamma_1 \uplus \Gamma_2)$.



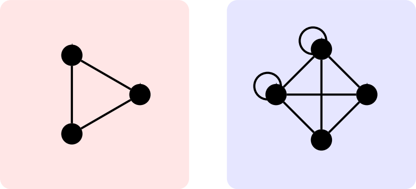
Free product

Free product: $\mathbb{M}\Gamma_1 * \mathbb{M}\Gamma_2 = \mathbb{M}(\Gamma_1 \uplus \Gamma_2)$.



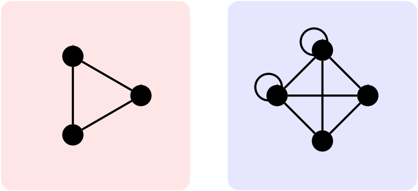
Free product

Free product: $\mathbb{M}\Gamma_1 * \mathbb{M}\Gamma_2 = \mathbb{M}(\Gamma_1 \uplus \Gamma_2)$.

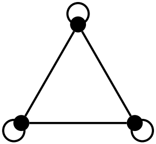


Free product

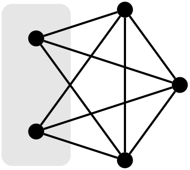
Free product: $\mathbb{M}\Gamma_1 * \mathbb{M}\Gamma_2 = \mathbb{M}(\Gamma_1 \uplus \Gamma_2)$.



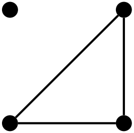
Graph monoids



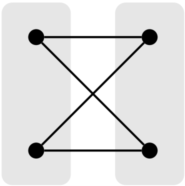
$\mathbb{Z}^3$



$(\mathbb{B} * \mathbb{B}) \times \mathbb{B}^3$

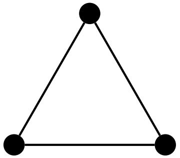


$\mathbb{B} * \mathbb{B}^3$

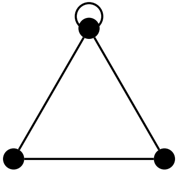


$(\mathbb{B} * \mathbb{B}) \times (\mathbb{B} * \mathbb{B})$

$\mathbb{B}^2$-triangles



$\mathbb{B}^2 \times \mathbb{B}$



$\mathbb{B}^2 \times \mathbb{Z}$

Def A graph is $\mathbb{B}^2$*-triangle-free* if it does not contain a $\mathbb{B}^2$-triangle as induced subgraph.

Main Result

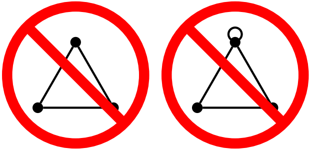
Theorem

FO[R] for valence systems over $\mathbb{M}\Gamma$ is decidable

if and only if

Γ is a disjoint union of $\mathbb{B}^2$-triangle-free cliques.

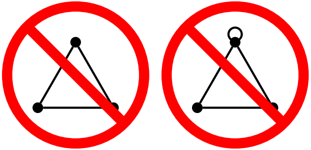
Characterisation of decidable cases



Γ is a disjoint union of $\mathbb{B}^2$-triangle-free cliques

Allowed cliques:

Characterisation of decidable cases

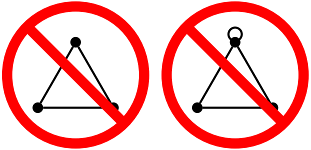


Γ is a disjoint union of $\mathbb{B}^2$-triangle-free cliques

Allowed cliques:

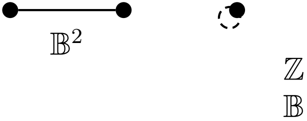
$\mathbb{B}^2$

Characterisation of decidable cases

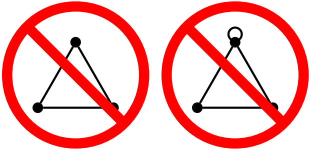


Γ is a disjoint union of $\mathbb{B}^2$-triangle-free cliques

Allowed cliques:



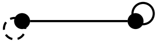
Characterisation of decidable cases



Γ is a disjoint union of $\mathbb{B}^2$-triangle-free cliques

Allowed cliques:

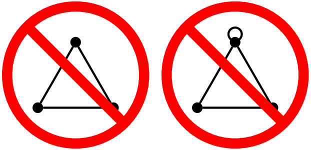
$\mathbb{B}^2$



$\mathbb{Z} \times \mathbb{Z}$

$\mathbb{B} \times \mathbb{Z}$

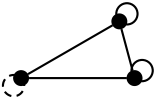
Characterisation of decidable cases



Γ is a disjoint union of $\mathbb{B}^2$-triangle-free cliques

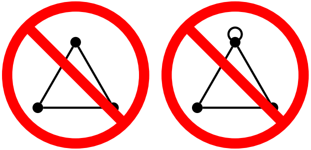
Allowed cliques:

$\mathbb{B}^2$



$\mathbb{Z} \times \mathbb{Z}^2$
$\mathbb{B} \times \mathbb{Z}^2$

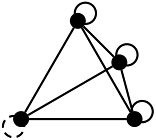
Characterisation of decidable cases



Γ is a disjoint union of $\mathbb{B}^2$-triangle-free cliques

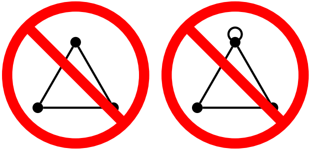
Allowed cliques:

$\mathbb{B}^2$



$\mathbb{Z} \times \mathbb{Z}^3$
$\mathbb{B} \times \mathbb{Z}^3$

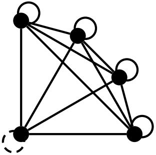
Characterisation of decidable cases



Γ is a disjoint union of $\mathbb{B}^2$-triangle-free cliques

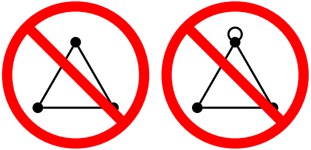
Allowed cliques:

$\mathbb{B}^2$



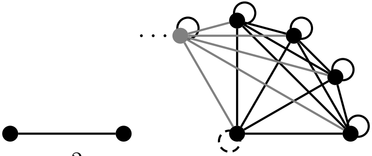
$\mathbb{Z} \times \mathbb{Z}^4$
$\mathbb{B} \times \mathbb{Z}^4$

Characterisation of decidable cases



Γ is a disjoint union of $\mathbb{B}^2$-triangle-free cliques

Allowed cliques:



$\mathbb{B}^2$

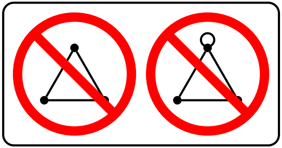
$\mathbb{Z} \times \mathbb{Z}^n$
$\mathbb{B} \times \mathbb{Z}^n$

$\mathbb{M}\Gamma_1 * \mathbb{M}\Gamma_2 * \cdots * \mathbb{M}\Gamma_k$ where Γ_i are cliques as above.

Operationally: Stack with as entries either

- 2 partially blind counters, or
- a partially blind counter with n blind counters.

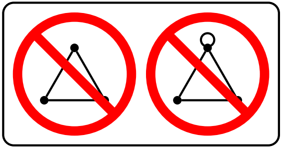
Decidability

 $\Longrightarrow$ FO[R] is decidable

Proof By showing **automaticity of the reachability structure**: the step and reachability relations can be represented by *regular relations*.

By (Khoussainov & Nerode 1995) the first-order theory of an automatic structure is decidable.

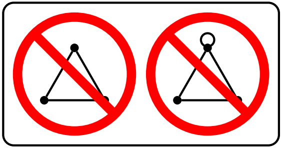
Decidability

 $\Longrightarrow$ FO[R] is decidable

We show automaticity for the reachability structures over:

- $\mathbb{B} \times \mathbb{B}$ a consequence of Presburger definability of reachability for 2-dimension VASS (Leroux & Sutre 2004)
- $\mathbb{B} \times \mathbb{Z}^n$ direct construction showing Presburger definability of reachability via Parikh images for 1-counter automata
- $M_0 * M_1$ when established for M_0 and M_1

Decidability

 $\Longrightarrow$ FO[R] is decidable

We show automaticity for the reachability structures over:

- $\mathbb{B} \times \mathbb{B}$ a consequence of Presburger definability of reachability for 2-dimension VASS (Leroux & Sutre 2004)
- $\mathbb{B} \times \mathbb{Z}^n$ direct construction showing Presburger definability of reachability via Parikh images for 1-counter automata
- $M_0 * M_1$ when established for M_0 and M_1

Decidability

Goal: $M_0 * M_1$ is automatic when M_0 and M_1 are automatic

Theorem If M has *automatic rational multiplication* then M has an effectively automatic reachability structure.

Decidability

Goal: $M_0 * M_1$ is automatic when M_0 and M_1 are automatic

Theorem If M has *automatic rational multiplication* then M has an effectively automatic reachability structure.

M has automatic rational multiplication if

- given a finite automaton representing $R \subseteq M$
- we can compute a synchronous automaton representing

$$R^{\odot} := \{(u, v) \in M \times M \mid \exists r \in R \colon u \odot r = v\}$$

Decidability

Goal: $M_0 * M_1$ has aut. mult. when M_0 and M_1 have aut. mult.

Theorem If M has *automatic rational multiplication* then M has an effectively automatic reachability structure.

M has automatic rational multiplication if

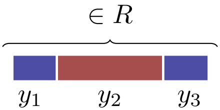
- given a finite automaton representing $R \subseteq M$
- we can compute a synchronous automaton representing

$$R^{\odot} := \{(u, v) \in M \times M \mid \exists r \in R \colon u \odot r = v\}$$

Decidability

Goal: $M_0 * M_1$ has aut. mult. when M_0 and M_1 have aut. mult.

Assume $R \subseteq M_0 * M_1$ is represented by finite automaton A.



Decidability

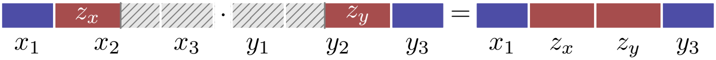
Goal: $M_0 * M_1$ has aut. mult. when M_0 and M_1 have aut. mult.

Assume $R \subseteq M_0 * M_1$ is represented by finite automaton A.

Decidability

Goal: $M_0 * M_1$ has aut. mult. when M_0 and M_1 have aut. mult.

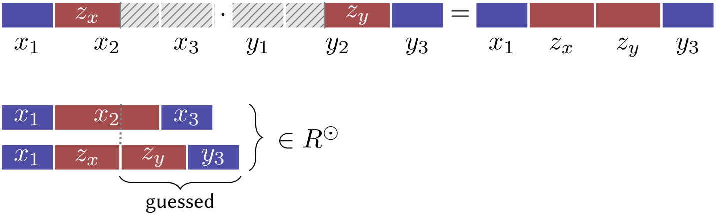
Assume $R \subseteq M_0 * M_1$ is represented by finite automaton A.



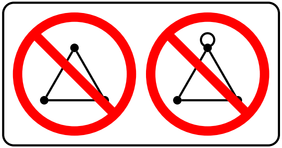
Decidability

Goal: $M_0 * M_1$ has aut. mult. when M_0 and M_1 have aut. mult.

Assume $R \subseteq M_0 * M_1$ is represented by finite automaton A.

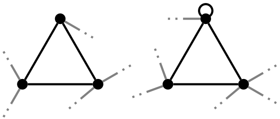


Decidability



$\Longrightarrow$ FO[R] is decidable

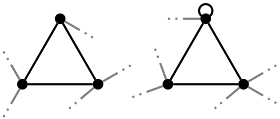
Undecidability



Γ is **not** a disjoint union of $\mathbb{B}^2$-triangle-free cliques

$\Longrightarrow$ FO[R] is undecidable

Undecidability

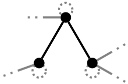


Γ is **not** a disjoint union of $\mathbb{B}^2$-triangle-free cliques

$\Longrightarrow$ FO[R] is undecidable

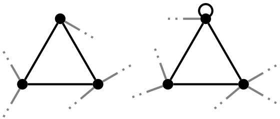
↓

Case 1: not all cliques



Submonoid $\{a, b\}^* \times \{c\}^*$

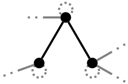
Undecidability



Γ is **not** a disjoint union of $\mathbb{B}^2$-triangle-free cliques

$\Longrightarrow$ FO[R] is undecidable

Case 1: not all cliques



Submonoid $\{a, b\}^* \times \{c\}^*$

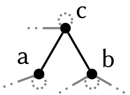
Case 2: contains a $\mathbb{B}^2$-triangle

Submonoid

$\mathbb{B}^2 \times \mathbb{B}$ or $\mathbb{B}^2 \times \mathbb{Z}$

Undecidability

Case 1: not all cliques



Submonoid $\{a, b\}^* \times \{c\}^*$

We can prove undecidability without barred symbols:

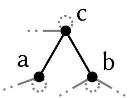
1. Use a and b as in a stack without popping.
2. Use c in a counter never decrementing.

We have two proofs:

- There is a *fixed* FO[R]-formula that cannot be checked for valence systems over $\mathbb{M}\Gamma$.
- There is a *fixed* valence system over $\mathbb{M}\Gamma$ with an undecidable first-order theory with reachability.

Undecidability

Case 1: not all cliques



Submonoid $\{a,b\}^* \times \{c\}^*$

We can prove undecidability without barred symbols:

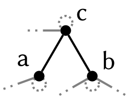
1. Use a and b as in a stack without popping.
2. Use c in a counter never decrementing.

We have two proofs:

- There is a *fixed* FO[R]-formula that cannot be checked for valence systems over $\mathbb{M}\Gamma$.
  By reducing universality of rational subsets of $\{a,b\}^* \times \{c\}^*$ which is undecidable (Sakarovitch 1992)
- There is a *fixed* valence system over $\mathbb{M}\Gamma$ with an undecidable first-order theory with reachability.

Undecidability

Case 1: not all cliques



Submonoid $\{a, b\}^* \times \{c\}^*$

We can prove undecidability without barred symbols:

1. Use a and b as in a stack without popping.
2. Use c in a counter never decrementing.

We have two proofs:

- There is a *fixed* FO[R]-formula that cannot be checked for valence systems over $\mathbb{M}\Gamma$.
  By reducing universality of rational subsets of $\{a, b\}^* \times \{c\}^*$ which is undecidable (Sakarovitch 1992)
- There is a *fixed* valence system over $\mathbb{M}\Gamma$ with an undecidable first-order theory with reachability.
  By reducing a variant of PCP

Undecidability

Case 2: contains a $\mathbb{B}^2$-triangle

Submonoid
$\mathbb{B}^2 \times \mathbb{B}$ or $\mathbb{B}^2 \times \mathbb{Z}$

We can use
the submonoid $\mathbb{B}^2 \times \mathbb{N}$

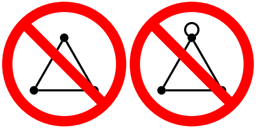
We can prove undecidability by using:

1. Two partially blind counters
2. A positive counter that we can only increment

The proof is by showing that there is a *fixed* valence automaton $A_{\mathbb{N}}$ on which $(\mathbb{N}, +, \cdot)$ can be interpreted:

- The Σ_1 fragment of arithmetic with addition and multiplication is undecidable (Matiyasevich 1993) $\Longrightarrow$ Σ_2 over $\mathbb{M}\Gamma$ is undecidable.
- Key trick:
  - squaring is enough $(a+b)^2 = a^2 + 2ab + b^2$
  - implement weak squaring by using $n^2 = \sum_{i=0}^{n-1} 2i + 1$

Conclusions

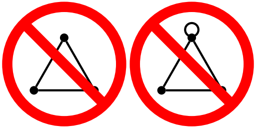


Γ is a disjoint union of $\mathbb{B}^2$-triangle-free cliques

iff

FO[R] for valence systems over $\mathbb{M}\Gamma$ is decidable

Conclusions



Γ is a disjoint union of $\mathbb{B}^2$-triangle-free cliques

iff

FO[R] for valence systems over $\mathbb{M}\Gamma$ is decidable

As an application, undecidability of FO[R] on 3-dimension VASS is a special case.

Thank you!