Synthesis for Structure Rewriting Systems

Łukasz Kaiser

Mathematische Grundlagen der Informatik
RWTH Aachen

MFCS

High Tatras, 2009

Structure Rewriting History

Relational Structures and Dynamics of Certain Discrete Systems

Václav Rajlich, 2nd MFCS, High Tatras, *summer* 1973

Structure Rewriting History

Relational Structures and Dynamics of Certain Discrete Systems

Václav Rajlich, 2nd MFCS, High Tatras, *summer* 1973

On Oriented Hypergraphs and on Dynamics of some Discrete Systems

Václav Rajlich (abstract), vol. 1 of LNCS, *autumn* 1973

Structure Rewriting History

Relational Structures and Dynamics of Certain Discrete Systems

Václav Rajlich, 2nd MFCS, High Tatras, *summer* 1973

On Oriented Hypergraphs and on Dynamics of some Discrete Systems

Václav Rajlich (abstract), vol. 1 of LNCS, *autumn* 1973

This structure is well suited for our purposes, namely for its intuitive appeal, immense generality and flexibility, and also for its potential in description of the real world, consisting of interrelated objects.

Structure Rewriting History

Relational Structures and Dynamics of Certain Discrete Systems

Václav Rajlich, 2nd MFCS, High Tatras, *summer* 1973

On Oriented Hypergraphs and on Dynamics of some Discrete Systems

Václav Rajlich (abstract), vol. 1 of LNCS, *autumn* 1973

This structure is well suited for our purposes, namely for its intuitive appeal, immense generality and flexibility, and also for its potential in description of the real world, consisting of interrelated objects.

Since then …

- Theory of **graph grammars**, **tree decompositions**, connections to **MSO**
- Applications to **software engineering and verification**

Structure Rewriting Rules

Relational Structures and Embeddings

$$\sigma \ : \quad \mathfrak{A} = (A, R_1^{\mathfrak{A}}, R_2^{\mathfrak{A}}, \dots, R_k^{\mathfrak{A}}) \quad \to \quad (B, R_1^{\mathfrak{B}}, R_2^{\mathfrak{B}}, \dots, R_k^{\mathfrak{B}}) = \mathfrak{B}$$

Embedding: σ is **injective** and $R_i^{\mathfrak{A}}(a_1, \dots, a_{r_i}) \Leftrightarrow R_i^{\mathfrak{B}}(\sigma(a_1), \dots, \sigma(a_{r_i}))$

Structure Rewriting Rules

Relational Structures and Embeddings

$$\sigma \;:\quad \mathfrak{A} = (A, R_1^{\mathfrak{A}}, R_2^{\mathfrak{A}}, \ldots, R_k^{\mathfrak{A}}) \quad\to\quad (B, R_1^{\mathfrak{B}}, R_2^{\mathfrak{B}}, \ldots, R_k^{\mathfrak{B}}) = \mathfrak{B}$$

Embedding: σ is **injective** and $R_i^{\mathfrak{A}}(a_1, \ldots, a_{r_i}) \Leftrightarrow R_i^{\mathfrak{B}}(\sigma(a_1), \ldots, \sigma(a_{r_i}))$

Rewriting Definition

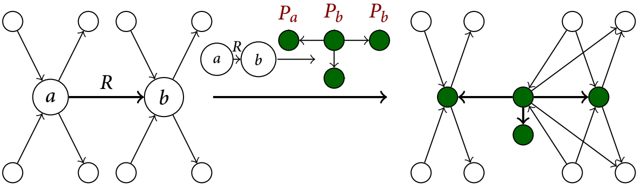
$\mathfrak{B} = \mathfrak{A}[\mathfrak{L} \to \mathfrak{R}/\sigma]$ iff $B = (A \setminus \sigma(L))\dot\cup R$ and,

for $M = \{(r, a) \mid a = \sigma(l), r \in P_l^{\mathfrak{R}}$ for some $l \in L\} \cup \{(a, a) \mid a \in A\}$,

$$(b_1, \ldots, b_{r_i}) \in R_i^{\mathfrak{B}} \;\Leftrightarrow\; (b_1, \ldots, b_{r_i}) \in R_i^{\mathfrak{R}} \textbf{ or } (b_1 M \times \ldots \times b_{r_i} M) \cap R_i^{\mathfrak{A}} \neq \varnothing.$$

(in the second case at least one $b_j \notin \mathfrak{A}$)

Rewriting Example



Structure Rewriting Games

Game arena is a **directed graph** with:

- vertices partitioned into positions of **Player 0** and **Player 1**
- edges **labelled by rewriting rules**

Structure Rewriting Games

Game arena is a **directed graph** with:

- vertices partitioned into positions of **Player 0** and **Player 1**
- edges **labelled by rewriting rules**

Two interpretations of $\mathfrak{L} \to \mathfrak{R}$:

- **Existential**: $\mathfrak{A}_{\text{next}} = \mathfrak{A}[\mathfrak{L} \to \mathfrak{R}/\sigma]$, the player chooses the embedding σ
- **Universal**: $\mathfrak{A}_{\text{next}} = \mathfrak{A}[\mathfrak{L} \to \mathfrak{R}]$, **all** occurrences of $\mathfrak{L}$ are rewritten to $\mathfrak{R}$

Structure Rewriting Games

Game arena is a **directed graph** with:

- vertices partitioned into positions of **Player 0** and **Player 1**
- edges **labelled by rewriting rules**

Two interpretations of $\mathfrak{L} \to \mathfrak{R}$:

- **Existential**: $\mathfrak{A}_{\text{next}} = \mathfrak{A}[\mathfrak{L} \to \mathfrak{R}/\sigma]$, the player chooses the embedding σ
- **Universal**: $\mathfrak{A}_{\text{next}} = \mathfrak{A}[\mathfrak{L} \to \mathfrak{R}]$, **all** occurrences of $\mathfrak{L}$ are rewritten to $\mathfrak{R}$

Winning condition:

- L_μ (or temporal) formula ψ with **MSO sentences** for predicates, or
- MSO formula φ to be evaluated on the **limit** of the play
  Limit of $\mathfrak{A}_0\mathfrak{A}_1\mathfrak{A}_2 \ldots \ = \ \left(\bigcup_{n\in\mathbb{N}} \bigcap_{i\geq n} A_i,\ \bigcup_{n\in\mathbb{N}} \bigcap_{i\geq n} R^{\mathfrak{A}_i}\right)$

Structure Rewriting Games

Game arena is a **directed graph** with:

- vertices partitioned into positions of **Player 0** and **Player 1**
- edges **labelled by rewriting rules**

Two interpretations of $\mathfrak{L} \to \mathfrak{R}$:

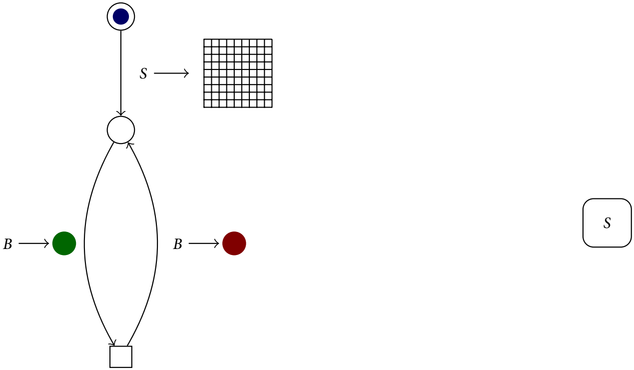
- **Existential**: $\mathfrak{A}_{\text{next}} = \mathfrak{A}[\mathfrak{L} \to \mathfrak{R}/\sigma]$, the player chooses the embedding σ
- **Universal**: $\mathfrak{A}_{\text{next}} = \mathfrak{A}[\mathfrak{L} \to \mathfrak{R}]$, **all** occurrences of $\mathfrak{L}$ are rewritten to $\mathfrak{R}$

Winning condition:

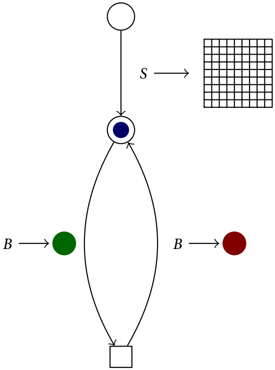
- L_μ (or temporal) formula ψ with **MSO sentences** for predicates, or
- MSO formula φ to be evaluated on the **limit** of the play
  Limit of $\mathfrak{A}_0\mathfrak{A}_1\mathfrak{A}_2 \ldots = \left(\bigcup_{n\in\mathbb{N}} \bigcap_{i\geq n} A_i,\ \bigcup_{n\in\mathbb{N}} \bigcap_{i\geq n} R^{\mathfrak{A}_i}\right)$

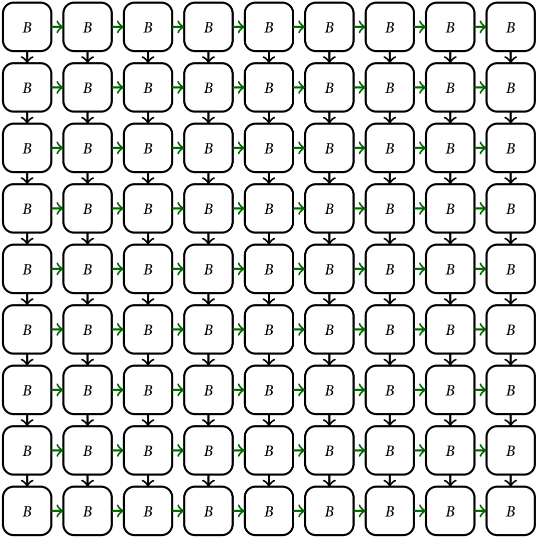
Motivation: many questions are **naturally defined as such games**:
constraint satisfaction, model checking, graph measures, games people play

Example Game: Connect–5

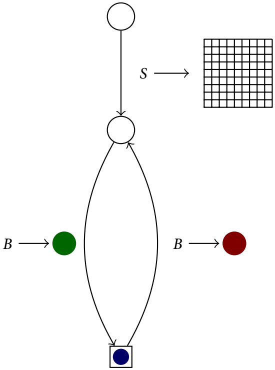


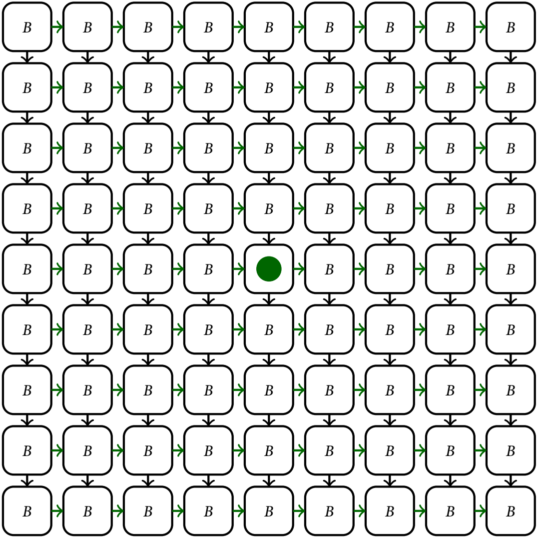
Example Game: Connect–5



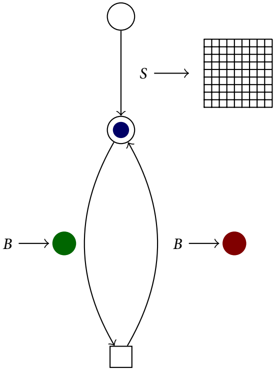


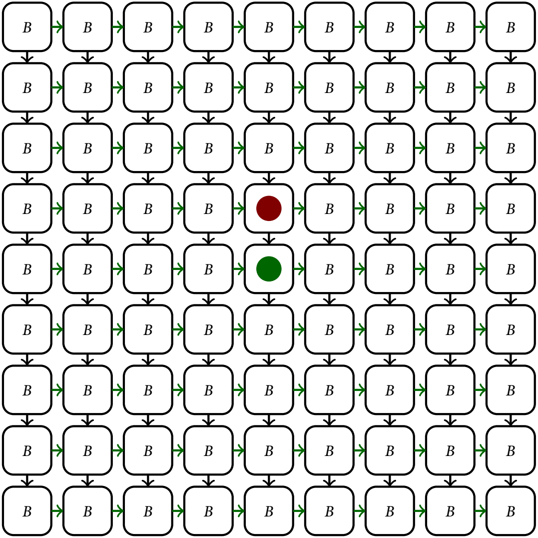
Example Game: Connect–5



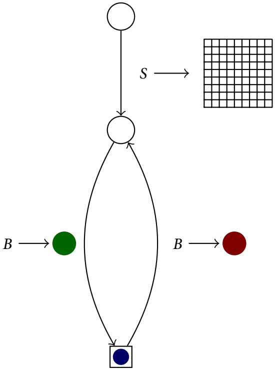


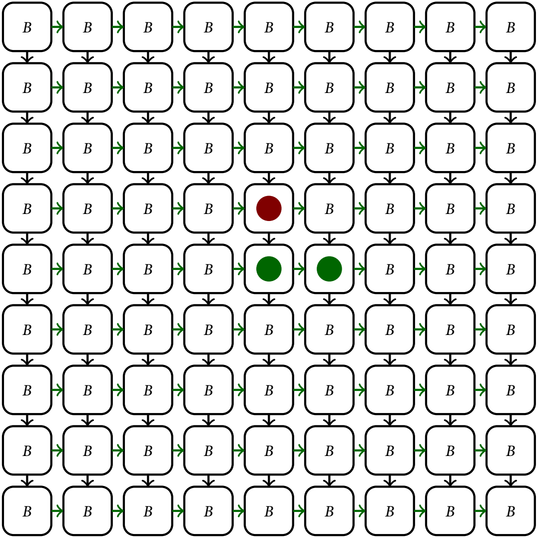
Example Game: Connect–5



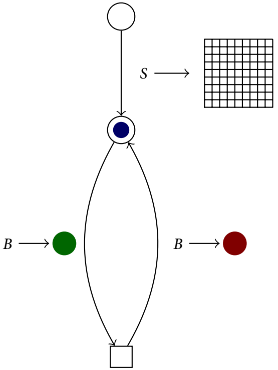


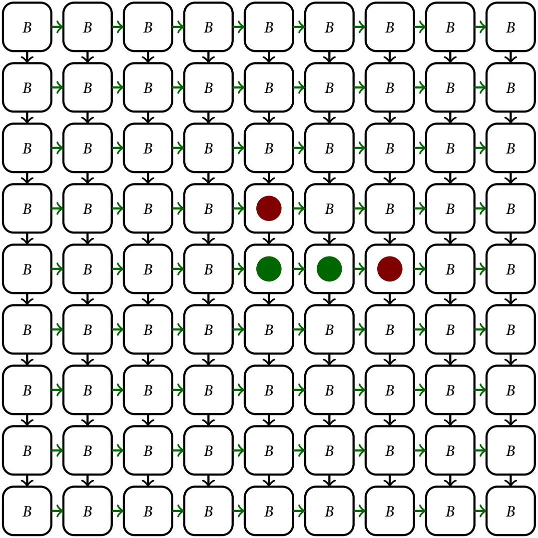
Example Game: Connect–5





Example Game: Connect–5



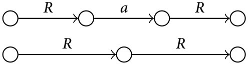


$$\exists x_1 \dots x_5 \Big(\bigwedge_{1\le i\le 5} G(x_i) \wedge \Big(\bigwedge_{1\le i\le 5} R(x_i, x_{i+1}) \vee \bigwedge_{1\le i\le 5} C(x_i, x_{i+1}) \vee \bigwedge_{1\le i\le 5} \exists y \big(R(x_i, y) \wedge C(y, x_{i+1})\big) \vee \bigwedge_{1\le i\le 5} \exists y \big(R(x_i, y) \wedge C(x_{i+1}, y)\big)\Big)\Big)$$

Simple Structure Rewriting

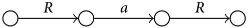
Separated Structures: no element is in two non-terminal relations
(Courcelle, Engelfriet, Rozenberg, 1991)

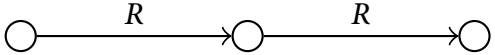
Separated:
Not Separated:



Simple Structure Rewriting

Separated Structures: no element is in two non-terminal relations
(Courcelle, Engelfriet, Rozenberg, 1991)

Separated: 

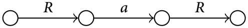
Not Separated: 

Simple Rule $\mathfrak{L} \to \mathfrak{R}$: $\mathfrak{R}$ is **separated** and $\mathfrak{L}$ is a **single tuple in relation**

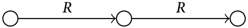
Simple Structure Rewriting

Separated Structures: no element is in two non-terminal relations **(Courcelle, Engelfriet, Rozenberg, 1991)**

Separated:

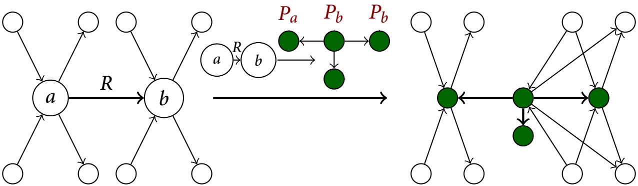


Not Separated:



Simple Rule $\mathfrak{L} \to \mathfrak{R}$: $\mathfrak{R}$ is **separated** and $\mathfrak{L}$ is a **single tuple in relation**

Example



Main Result

Logics

- $\mathrm{L}_\mu[\mathrm{MSO}]$: Temporal properties expressed in L_μ (subsumes LTL) with properties of structures (states) expressed in MSO
- lim MSO: Property of the limit structure expressed in MSO

Theorem

- *Let R be a* ***finite*** *set of* ***(universal) simple structure rewriting rules,***
- *and φ be an $\mathrm{L}_\mu[\mathrm{MSO}]$ or lim MSO formula.*

Then the set $\{\pi \in R^\omega : (\lim) S(\pi) \vDash \varphi\}$ is ***ω-regular****.*

Corollary

Establishing the winner of (universal) finite simple rewriting games is decidable.

Why Universal Rewriting?

Simple Rewriting: Universal vs. Existential

- **Universal** is arguably **less natural** than the **Existential**
- **Graph grammars** (one player) are defined in the **Existential** way
- **Generated structures** have bounded clique-width **in both cases**

Why Universal Rewriting?

Simple Rewriting: Universal vs. Existential

- **Universal** is arguably **less natural** than the **Existential**
- **Graph grammars** (one player) are defined in the **Existential** way
- **Generated structures** have bounded clique-width **in both cases**

Establishing the winner in existential games is undecidable:
Simulate **active context-free games** (thanks to **Anca Muscholl**)

Active Context-Free Games

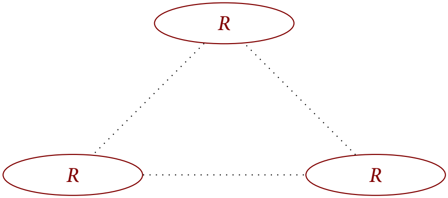
- Played on **a word** (letters $\rightsquigarrow$ predicates)
- **Juliet** selects **a position** in the word
- **Romeo** selects **a CFG rule** to apply
- **Winner:** Juliet wins if a word in **a regular language** L is reached

Proof: Intuition behind Simple Rewriting

Simple Rewriting ignoring Terminal Relations

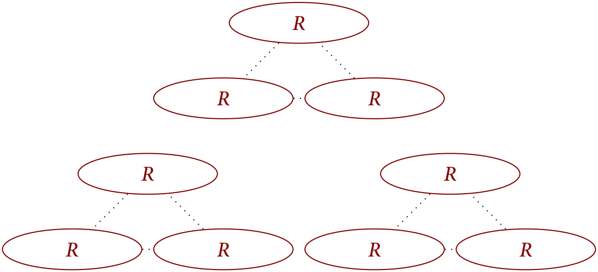
Proof: Intuition behind Simple Rewriting

Simple Rewriting ignoring Terminal Relations



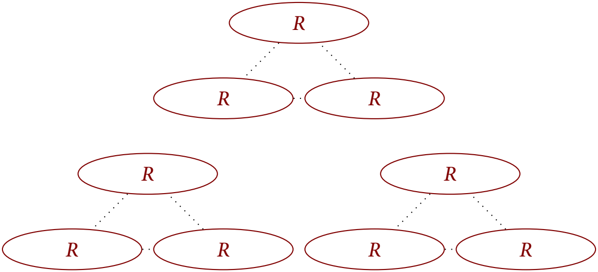
Proof: Intuition behind Simple Rewriting

Simple Rewriting ignoring Terminal Relations



Proof: Intuition behind Simple Rewriting

Simple Rewriting ignoring Terminal Relations



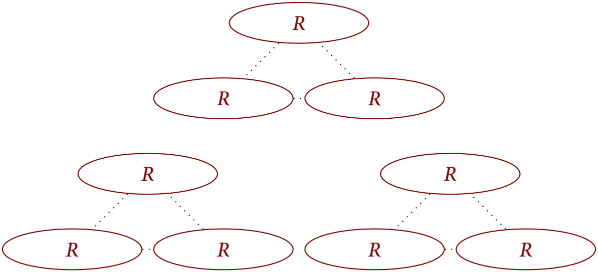
MSO is compositional:

$$\mathrm{Th}^k(\mathfrak{A} \oplus^{\text{connect}} \mathfrak{B}) = \mathrm{Th}^k(\mathfrak{A}) \oplus^{\text{connect}} \mathrm{Th}^k(\mathfrak{B})$$

Not compositional: e.g. $|P| = |Q|$ (in SO, MSO_2 using Hamilton cycle)

Proof: Intuition behind Simple Rewriting

Simple Rewriting ignoring Terminal Relations



MSO is compositional:

$$\mathrm{Th}^k(\mathfrak{A} \oplus^{\text{connect}} \mathfrak{B}) = \mathrm{Th}^k(\mathfrak{A}) \oplus^{\text{connect}} \mathrm{Th}^k(\mathfrak{B})$$

Not compositional: e.g. $|P| = |Q|$ (in SO, MSO_2 using Hamilton cycle)

Proof formally: through MSO interpretation in the binary tree

Proof: Interpreting a Structure in a Tree

Description of how to build $\mathfrak{A}$ is a tree $\mathcal{T}(\mathfrak{A})$ with:

- **Leafs** of **different colours** $1 \ldots k$
- $\oplus$ representing **disjoint sum**
- $i \leftarrow j$ to **change colour** of **all** nodes from i to j
- $e(i, j)$ to **add all pairs** of (i, j)-coloured nodes to e

Proof: Interpreting a Structure in a Tree

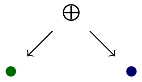
Description of how to build $\mathfrak{A}$ is a tree $\mathcal{T}(\mathfrak{A})$ with:

- **Leafs** of **different colours** $1 \dots k$
- $\oplus$ representing **disjoint sum**
- $i \leftarrow j$ to **change colour** of **all** nodes from i to j
- $e(i, j)$ to **add all pairs** of (i, j)-coloured nodes to e

Proof: Interpreting a Structure in a Tree

Description of how to build $\mathfrak{A}$ is a tree $\mathcal{T}(\mathfrak{A})$ with:

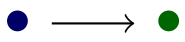
- **Leafs** of **different colours** $1 \ldots k$
- $\oplus$ representing **disjoint sum**
- $i \leftarrow j$ to **change colour** of **all** nodes from i to j
- $e(i, j)$ to **add all pairs** of (i, j)-coloured nodes to e

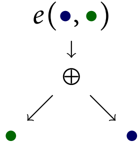


Proof: Interpreting a Structure in a Tree

Description of how to build $\mathfrak{A}$ is a tree $\mathcal{T}(\mathfrak{A})$ with:

- **Leafs** of **different colours** $1 \dots k$
- $\oplus$ representing **disjoint sum**
- $i \leftarrow j$ to **change colour** of **all** nodes from i to j
- $e(i, j)$ to **add all pairs** of (i, j)-coloured nodes to e

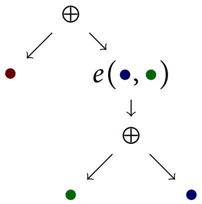




Proof: Interpreting a Structure in a Tree

Description of how to build $\mathfrak{A}$ is a tree $\mathcal{T}(\mathfrak{A})$ with:

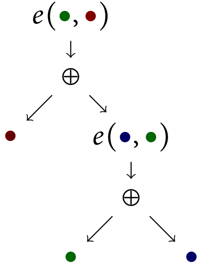
- **Leafs** of **different colours** $1 \dots k$
- $\oplus$ representing **disjoint sum**
- $i \leftarrow j$ to **change colour** of **all** nodes from i to j
- $e(i, j)$ to **add all pairs** of (i, j)-coloured nodes to e



Proof: Interpreting a Structure in a Tree

Description of how to build $\mathfrak{A}$ is a tree $\mathcal{T}(\mathfrak{A})$ with:

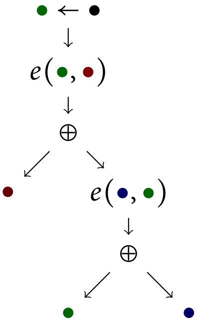
- **Leafs** of **different colours** $1 \ldots k$
- $\oplus$ representing **disjoint sum**
- $i \leftarrow j$ to **change colour** of **all** nodes from i to j
- $e(i, j)$ to **add all pairs** of (i, j)-coloured nodes to e



Proof: Interpreting a Structure in a Tree

Description of how to build $\mathfrak{A}$ is a tree $\mathcal{T}(\mathfrak{A})$ with:

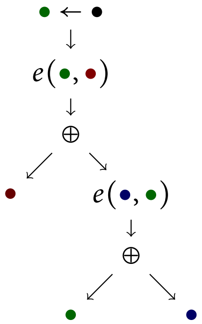
- **Leafs** of **different colours** $1 \dots k$
- $\oplus$ representing **disjoint sum**
- $i \leftarrow j$ to **change colour** of **all** nodes from i to j
- $e(i, j)$ to **add all pairs** of (i, j)-coloured nodes to e



Proof: Interpreting a Structure in a Tree

Description of how to build $\mathfrak{A}$ is a tree $\mathcal{T}(\mathfrak{A})$ with:

- **Leafs** of **different colours** $1 \dots k$
- $\oplus$ representing **disjoint sum**
- $i \leftarrow j$ to **change colour** of **all** nodes from i to j
- $e(i, j)$ to **add all pairs** of (i, j)-coloured nodes to e



Theorem:
For every k there is an MSO-to-MSO **interpretation** $\mathcal{I}$ such that for all structures $\mathfrak{A}$ of **clique-width** $\leq k$ holds $\mathcal{I}(\mathcal{T}(\mathfrak{A})) \cong \mathfrak{A}$.

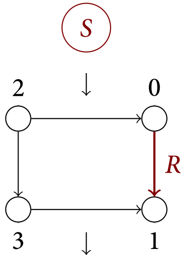
Proof: Simple Rewriting in the Tree

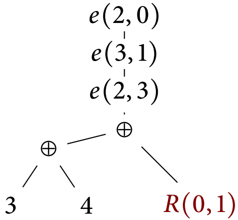
S

$\downarrow$

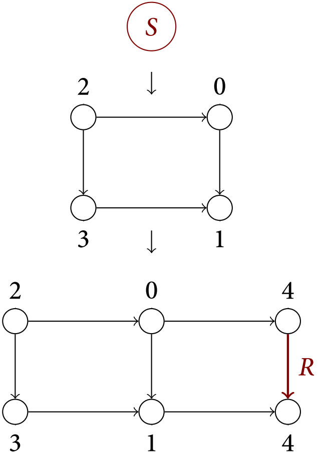
S

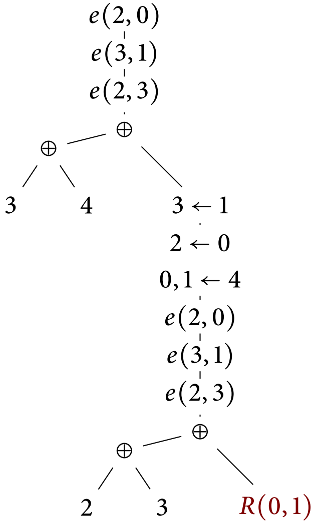
Proof: Simple Rewriting in the Tree



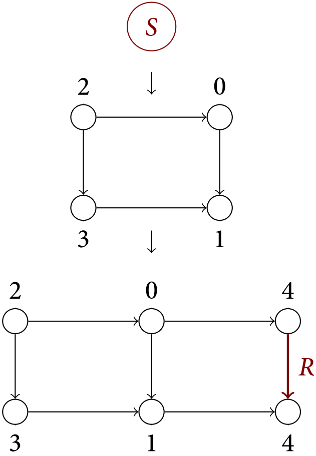


Proof: Simple Rewriting in the Tree

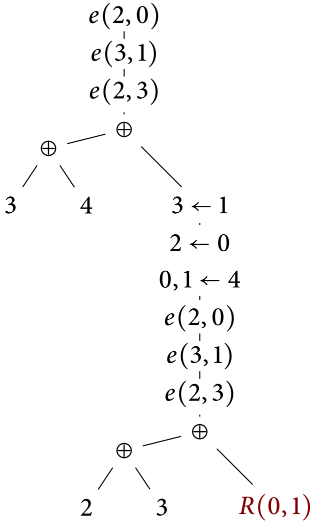




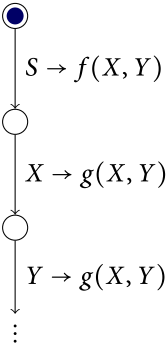
Proof: Simple Rewriting in the Tree



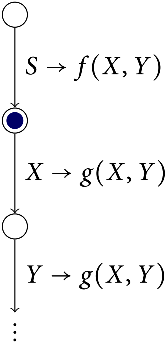
MSO-to-MSO **interpretation:** $\varphi \to \psi$

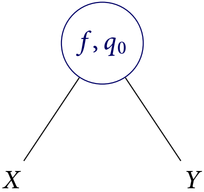


Proof: From Tree to Alternating Word Automata

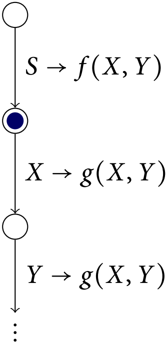


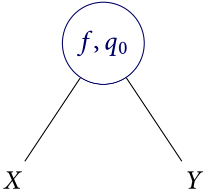
Proof: From Tree to Alternating Word Automata





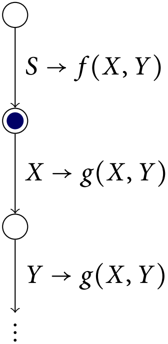
Proof: From Tree to Alternating Word Automata

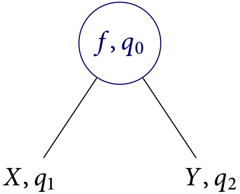




existential: pick transition

Proof: From Tree to Alternating Word Automata

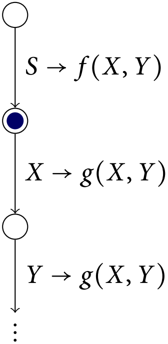


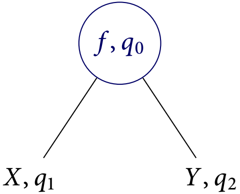


existential: pick transition

$f, q_0 \to (q_1, q_2)$

Proof: From Tree to Alternating Word Automata



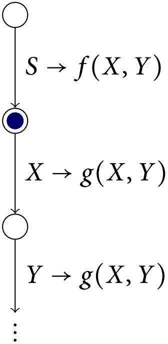


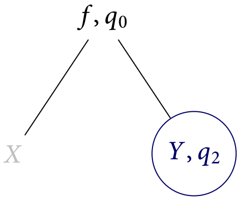
existential: pick transition

universal: left or right

$$f, q_0 \to (q_1, q_2)$$

Proof: From Tree to Alternating Word Automata



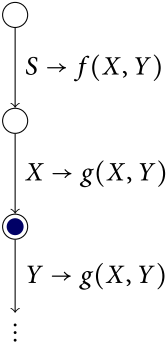


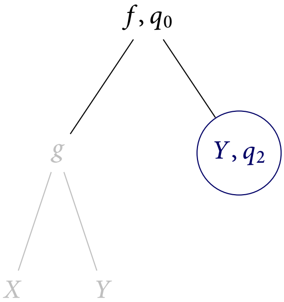
existential: pick transition

$f, q_0 \to (q_1, q_2)$

universal: left or right

Proof: From Tree to Alternating Word Automata





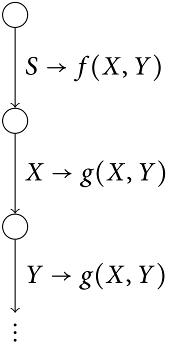
existential: pick transition

universal: left or right

$f, q_0 \to (q_1, q_2)$

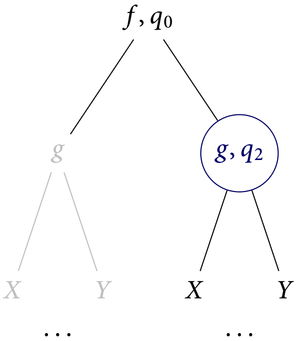
ignore

Proof: From Tree to Alternating Word Automata



existential: pick transition

universal: left or right



$f, q_0 \to (q_1, q_2)$

ignore

Outlook

Many questions are naturally defined as structure rewriting games.

Outlook

Many questions are naturally defined as structure rewriting games.

Checking MSO on structures of bounded clique-width
Important in practice, e.g. for software verification (separation logic)

- **MONA**: **fails for 3 colours** (reachability)
- **QBF Solvers**: work for **bounded model checking**
- **Composition Method**: **not tested yet** (works for reachability)

Outlook

Many questions are naturally defined as structure rewriting games.

Checking MSO on structures of bounded clique-width
Important in practice, e.g. for software verification (separation logic)

- **MONA**: **fails for 3 colours** (reachability)
- **QBF Solvers**: work for **bounded model checking**
- **Composition Method**: **not tested yet** (works for reachability)

Questions

- **Decidable fragments** with **existential** rules?
- Other **logics and graph measures**, e.g. for FO, FO[Reach]?
- Can we solve such games **in practice**?

Outlook

Many questions are naturally defined as structure rewriting games.

Checking MSO on structures of bounded clique-width
Important in practice, e.g. for software verification (separation logic)

- **MONA**: **fails for 3 colours** (reachability)
- **QBF Solvers**: work for **bounded model checking**
- **Composition Method**: **not tested yet** (works for reachability)

Questions

- **Decidable fragments** with **existential** rules?
- Other **logics and graph measures**, e.g. for FO, FO[Reach]?
- Can we solve such games **in practice**?

Thank You