Finite presentations of infinite structures: automata and interpretations

Erich Grädel

graedel@rwth-aachen.de.

Aachen University

(joint work with Achim Blumensath)

Logical definability versus computational complexity

Important issue in many fields:

- finite model theory
- databases
- verification
- complexity theory
- knowledge representation
- . . .

Well-understood on finite structures

Logical definability versus computational complexity

Important issue in many fields:

- finite model theory
- databases
- verification
- complexity theory
- knowledge representation
- . . .

Well-understood on finite structures

Limitation to finite structures is often too restrictive.

Logical definability versus computational complexity

Important issue in many fields:

- finite model theory
- databases
- verification
- complexity theory
- knowledge representation
- . . .

Well-understood on finite structures

Limitation to finite structures is often too restrictive.

Considerable efforts to extend methodology to relevant classes of infinite structures

- infinite databases: spatial databases, constraint databases, . . .
- verification for systems with infinite state spaces
- model theory of finitely presented structures

Computational model theory

extends approach and methods of finite model theory to suitable classes of infinite structures

- finite presentations of infinite structures
- complexity of model checking problems
- capturing complexity classes
- model theoretic constructions
- games

Computational model theory

$\mathcal{D}$: domain of not necessarily finite structures

What conditions should be satisfied by $\mathcal{D}$ so that approach of computational model theory is applicable?

Computational model theory

$\mathcal{D}$: domain of not necessarily finite structures

What conditions should be satisfied by $\mathcal{D}$ so that approach of computational model theory is applicable?

Finite presentations: Each structure $\mathfrak{A} \in \mathcal{K}$ should be representable in a finite way (by an algorithm, by an axiomatisation in some logic, by automata, by an interpretation, . . .).

Computational model theory

$\mathcal{D}$: domain of not necessarily finite structures

What conditions should be satisfied by $\mathcal{D}$ so that approach of computational model theory is applicable?

Finite presentations: Each structure $\mathfrak{A} \in \mathcal{K}$ should be representable in a finite way (by an algorithm, by an axiomatisation in some logic, by automata, by an interpretation, . . .).

Effective semantics (for relevant logic L): Given $\psi \in L$ and a (presentation of) $\mathfrak{A} \in \mathcal{D}$ it should be decidable whether $\mathfrak{A} \models \psi$. That is, model checking of L on $\mathcal{D}$ must be effective.

Other possibly relevant conditions (depending on context):

Closure: For $\mathfrak{A} \in \mathcal{D}$ and $\psi(\bar{x}) \in L$, also the expanded structure $(\mathfrak{A}, \psi^{\mathfrak{A}})$ is in $\mathcal{D}$.

Other possibly relevant conditions (depending on context):

Closure: For $\mathfrak{A} \in \mathcal{D}$ and $\psi(\bar{x}) \in L$, also the expanded structure $(\mathfrak{A}, \psi^{\mathfrak{A}})$ is in $\mathcal{D}$.

Effective query evaluation: Given a presentation of $\mathfrak{A} \in \mathcal{D}$ and a formula $\psi(\bar{x}) \in L$ one can effectively compute a presentation of $(\mathfrak{A}, \psi^{\mathfrak{A}})$.

Note: contrary to finite structures, query evaluation does not necessarily reduce to model checking.

Outline of this talk

- survey on different classes of finitely presented structures
- structures presented by interpretations
- structures presented by automata
- automatic groups
- algorithmic problems for automatic structures
- characterizing automatic structures by interpretations

Finitely presentable structures

- **recursive structures**
- **tree-interpretable structures**
  - context-free graphs
  - HR-equational and VR-equational graphs
  - prefix-recognizable graphs
- **tree constructible structures**
- **automatic structures, automatic groups, ω-automatic structures**
- other classes with finite presentations
  - tree-automatic structures, rational structures
  - ground tree rewriting graphs
  - constraint databases
  - metafinite structures

Recursive structures

Countable structures $\mathfrak{A} = (A, f_1, \dots, f_m, R_1, \dots, R_n)$ with computable functions and decidable relations

Long tradition in model theory since 1960s

Recursive structures

Countable structures $\mathfrak{A} = (A, f_1, \dots, f_m, R_1, \dots, R_n)$ with computable functions and decidable relations

Long tradition in model theory since 1960s

Problem: Only quantifier-free formulae have effective semantics

Recursive structures

Countable structures $\mathfrak{A} = (A, f_1, \dots, f_m, R_1, \dots, R_n)$ with computable functions and decidable relations

Long tradition in model theory since 1960s

Problem: Only quantifier-free formulae have effective semantics

Some work studying finite model theory issues for recursive structures

Recursive structures

Countable structures $\mathfrak{A} = (A, f_1, \ldots, f_m, R_1, \ldots, R_n)$ with computable functions and decidable relations

Long tradition in model theory since 1960s

Problem: Only quantifier-free formulae have effective semantics

Some work studying finite model theory issues for recursive structures

- failure of classical results (compactness, completeness, interpolation, Beth,. . .) on recursive structures. (Stolboushkin, Hirst–Harel)
- descriptive complexity (mostly on non-recursive levels) (Hirst–Harel)
- 0-1 laws (Hirst–Harel, G.–Malmström)

Interpretations

$\mathfrak{A}$ σ-structure, L logic, $\mathfrak{B}$ τ-structure

(k-dimensional) $L[\tau,\sigma]$-interpretation : sequence

$$I = \langle \underbrace{D(\overline{x})}_{\text{domain formula}}, \underbrace{E(\overline{x},\overline{y})}_{\text{equality formula}}, \underbrace{(\varphi_R(\overline{x}_1,\dots,\overline{x}_r))_{R\in\sigma}}_{\text{formulae defining relations}} \rangle$$

of $L[\tau]$-formulae (where $\overline{x}, \overline{u}, \overline{x}_i$ are k-tuples of variables)

Interpretations

$\mathfrak{A}$ σ-structure, L logic, $\mathfrak{B}$ τ-structure

(*k*-dimensional) $L[\tau,\sigma]$-interpretation : sequence

$$I = \langle \quad D(\overline{x}) \quad , \quad E(\overline{x},\overline{y}) \quad , \quad (\,\varphi_R(\overline{x}_1,\dots,\overline{x}_r)\,)_{R\in\sigma} \quad \rangle$$

domain formula — equality formula — formulae defining relations

of $L[\tau]$-formulae (where $\overline{x},\overline{u},\overline{x}_i$ are k-tuples of variables)

I interprets $\mathfrak{A}$ in $\mathfrak{B}$ (in short $I(\mathfrak{B}) = \mathfrak{A}$) if I defines a copy of $\mathfrak{A}$ inside $\mathfrak{B}$.

$$h : \quad I(\mathfrak{B}) := \langle D^{\mathfrak{B}}, (\varphi_R^{\mathfrak{B}})_{R\in\sigma}\rangle / E^{\mathfrak{B}} \quad \xrightarrow{\sim} \quad \mathfrak{A}$$

Interpretations

$\mathfrak{A}$ σ-structure, L logic, $\mathfrak{B}$ τ-structure

(k-dimensional) $L[\tau,\sigma]$-interpretation : sequence

$$I = \langle \underbrace{D(\bar{x})}_{\text{domain formula}}, \underbrace{E(\bar{x},\bar{y})}_{\text{equality formula}}, \underbrace{(\varphi_R(\bar{x}_1,\dots,\bar{x}_r))_{R\in\sigma}}_{\text{formulae defining relations}} \rangle$$

of $L[\tau]$-formulae (where $\bar{x}, \bar{u}, \bar{x}_i$ are k-tuples of variables)

I interprets $\mathfrak{A}$ in $\mathfrak{B}$ (in short $I(\mathfrak{B}) = \mathfrak{A}$) if I defines a copy of $\mathfrak{A}$ inside $\mathfrak{B}$.

$$h : \quad I(\mathfrak{B}) := \langle D^{\mathfrak{B}}, (\varphi_R^{\mathfrak{B}})_{R\in\sigma}\rangle / E^{\mathfrak{B}} \quad \xrightarrow{\sim} \quad \mathfrak{A}$$

$\mathfrak{A} \leq_L \mathfrak{B}$: there exists L-interpretation of $\mathfrak{A}$ in $\mathfrak{B}$

Interpretation Lemma

$L[\tau, \sigma]$-interpretation $\quad I = \langle D(\overline{x}),\ E(\overline{x}, \overline{y}),\ (\ \varphi_R(\overline{x}_1, \ldots, \overline{x}_r)\ )_{R\in\sigma}\rangle$

- I maps τ-structures $\mathfrak{B}$ to σ-structures $I(\mathfrak{B})$
- in turn, I maps σ-formulae ψ to τ-formulae $I(\psi)$:
  - replace variables $x, y, \ldots$ by k-tuples $\overline{x}, \overline{y}, \ldots$
  - relativize quantifiers to $D(\overline{x})$
  - replace equalities $x = y$ by $E(\overline{x}, \overline{y})$
  - replace atoms $Rx_1 \ldots x_r$ (for $R \in \sigma$) by $\varphi_R(\overline{x}_1, \ldots, \overline{x}_r)$

Interpretation Lemma: $\quad I(\mathfrak{B}) \models \psi \quad \Longleftrightarrow \quad \mathfrak{B} \models I(\psi)$

Structures presented by interpretations

Interpretations provide general and powerful way for defining classes of finitely presentable structures with effective semantics

Structures presented by interpretations

Interpretations provide general and powerful way for defining classes of finitely presentable structures with effective semantics

Take structure $\mathfrak{B}$ with "nice" properties and study closure $\{\mathfrak{A} : \mathfrak{A} \leq_L \mathfrak{B}\}$ under L-interpretations for suitable L.

Finite presentations: by interpretations into $\mathfrak{B}$

Effective semantics: if L is closed under interpretations and L is effective on $\mathfrak{B}$, then L is effective on any $\mathfrak{A} \leq_L \mathfrak{B}$.
(Interpretation Lemma)

Tree interpretable structures

$\mathcal{T}^2 = (\{0,1\}^*, \sigma_0, \sigma_1)$ infinite binary tree

MSO: monadic second-order logic

A structure $\mathfrak{A}$ is tree-interpretable if $\mathfrak{A} \leq_{\text{MSO}} \mathcal{T}^2$:

(one-dimensional) MSO-interpretation of $\mathfrak{A}$ in the infinite binary tree.

Tree interpretable structures

$\mathcal{T}^2 = (\{0,1\}^*, \sigma_0, \sigma_1)$ infinite binary tree

MSO: monadic second-order logic

A structure $\mathfrak{A}$ is tree-interpretable if $\mathfrak{A} \leq_{\text{MSO}} \mathcal{T}^2$:
(one-dimensional) MSO-interpretation of $\mathfrak{A}$ in the infinite binary tree.

Tree-interpretable structures admit effective evaluation of MSO

- Rabin's Theorem: The MSO-theory of $\mathcal{T}^2$ is decidable
- Interpretation Lemma

Tree interpretable graphs

Tree-interpretable graphs generalize various classes of finitely presentable graphs that admit effective evaluation of MSO.

Context-free graphs: (Muller, Schupp)

configuration graphs of pushdown automata

HR-equational and VR-equational graphs: (Courcelle)

defined by graph grammars

Prefix-recognizable graphs: (Caucal)

$G = (V, (E_a)_{a \in A}$ where V regular language, and

$$E_a = \bigcup_{i=1}^{m} X_i(Y_i \times Z_i) = \bigcup_{i=1}^{m} \{(xy, xz) : x \in X_i, y \in Y_i, z \in Z_i\}$$

for regular languages X_i, Y_i, Z_i.

Tree interpretable structures

Theorem (Barthelmann, Blumensath, Caucal, Courcelle, Stirling)

For any graph G, the following are equivalent.

(1) G is tree-interpretable

(2) G is VR-equational

(3) G is prefix-recognizable

(4) G is the restriction to a regular set of the configuration graph of a pushdown automaton with ε-transitions.

Tree interpretable structures

Theorem (Barthelmann, Blumensath, Caucal, Courcelle, Stirling)

For any graph G, the following are equivalent.

(1) G is tree-interpretable

(2) G is VR-equational

(3) G is prefix-recognizable

(4) G is the restriction to a regular set of the configuration graph of a pushdown automaton with ε-transitions.

The classes of context-free graphs and HR-equational graphs are strictly contained in the class of tree-interpretable graphs.

Tree-like structures

More powerful domains than the tree-interpretable structures on which MSO is effective?

Tree constructions:

- **Unfolding** of a labeled graph G from a node v to the tree $\mathcal{T}(G, v)$.
- **Muchnik´s construction:** With relational structure $\mathfrak{A} = (A, R_1, \ldots, R_m)$, associate its iteration

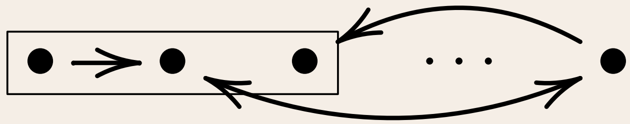
$$\mathfrak{A}^* := (A^*, R_1^*, \ldots, R_m^*, \text{son}, \text{clone})$$

with relations

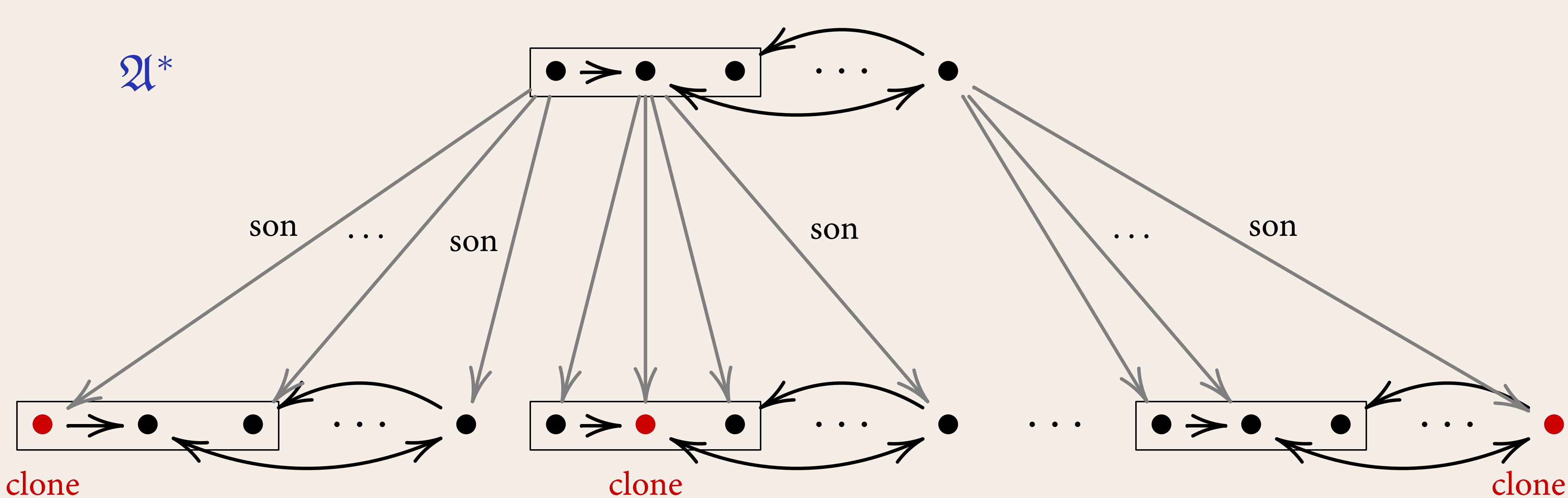
$$R_i^* := \{(wa_1, \ldots, wa_r) : w \in A^*, (a_1, \ldots, a_r) \in R_i\}$$
$$\text{son} := \{(w, wa) : w \in A^*, a \in A\}$$
$$\text{clone} := \{waa : w \in A^*, a \in A\}$$

Muchnik's construction

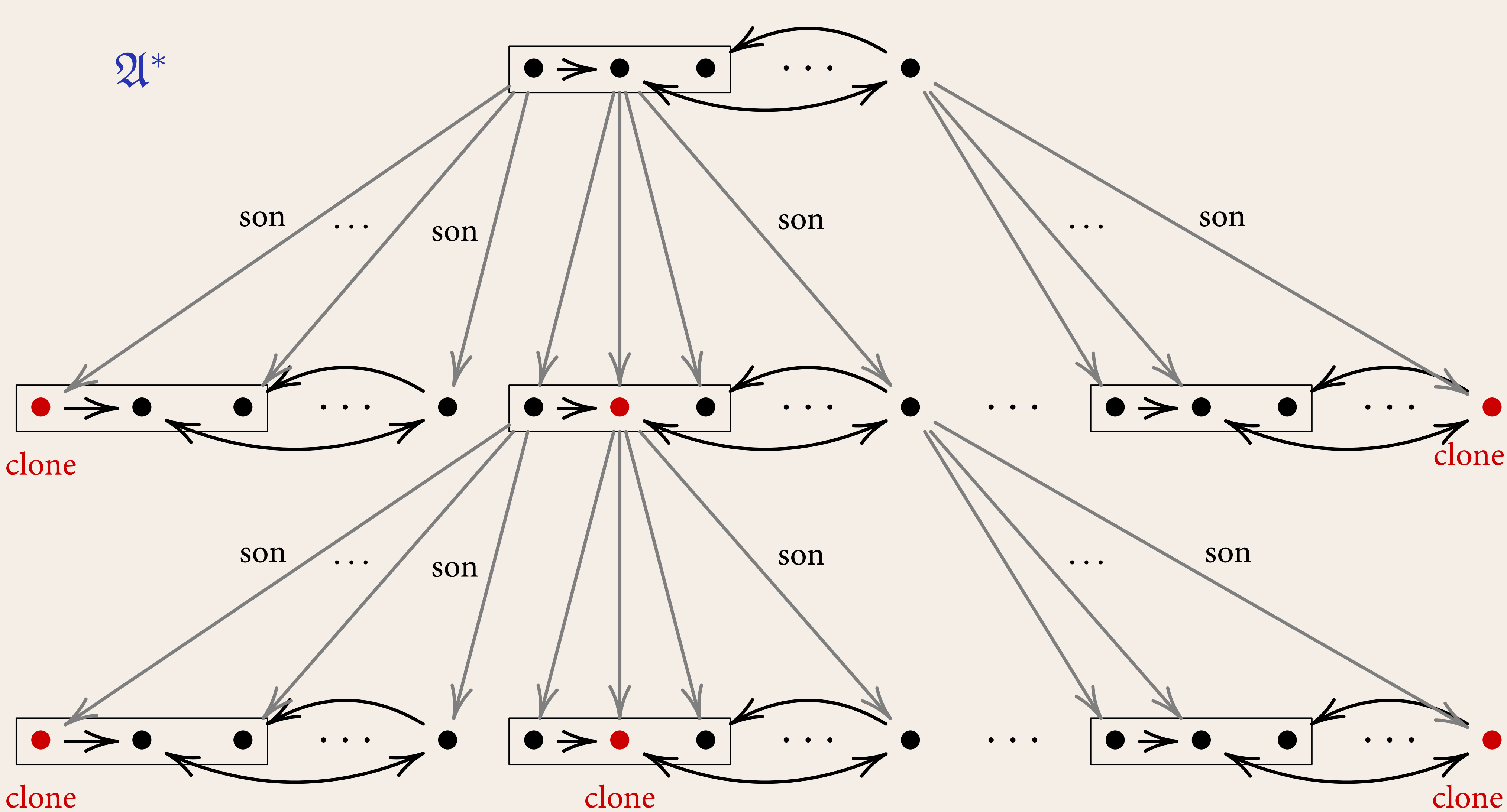
$\mathfrak{A}$



Muchnik's construction



Muchnik's construction



Tree constructible structures

Decidability

- If the MSO-theory of (G, v) is decidable, then so is the MSO-theory of its unfolding $\mathcal{T}(G, v)$ (Courcelle, Walukiewicz).
- If the MSO-theory of $\mathfrak{A}$ is decidable, then so is the MSO-theory of its iteration $\mathfrak{A}^*$ (Muchnik, Walukiewicz, Berwanger-Blumensath)

Tree constructible structures

Decidability

- If the MSO-theory of (G, v) is decidable, then so is the MSO-theory of its unfolding $\mathcal{T}(G, v)$ (Courcelle, Walukiewicz).
- If the MSO-theory of $\mathfrak{A}$ is decidable, then so is the MSO-theory of its iteration $\mathfrak{A}^*$ (Muchnik, Walukiewicz, Berwanger-Blumensath)

Unfoldings are interpretable in iterations: $\mathcal{T}(G, v) \leq_{\mathrm{MSO}} (G^*, v)$

Tree constructible structures

Decidability

- If the MSO-theory of (G, v) is decidable, then so is the MSO-theory of its unfolding $\mathcal{T}(G, v)$ (Courcelle, Walukiewicz).
- If the MSO-theory of $\mathfrak{A}$ is decidable, then so is the MSO-theory of its iteration $\mathfrak{A}^*$ (Muchnik, Walukiewicz, Berwanger-Blumensath)

Unfoldings are interpretable in iterations: $\mathcal{T}(G, v) \leq_{\mathrm{MSO}} (G^*, v)$

Tree constructible structures: Closure of finite structures under MSO-interpretations and Muchnik's construction.

- MSO is effective on tree constructible structures
- There exist tree constructible structures that are not tree interpretable (Courcelle)

Automatic structures

$\mathfrak{A} = (A, R_1, \dots, R_s)$ is automatic if there exist a regular language $L_\delta \subseteq \Sigma^*$ and a surjective function $h : L_\delta \to A$ such that the relations

$$L_= := \{(u, v) : h(u) = h(v)\} \subseteq L_\delta \times L_\delta$$

$$L_{R_i} := \{(u_1, \dots, u_r) : \mathfrak{A} \models R_i h(u_1) \dots h(u_r)\} \subseteq L_\delta \times \dots \times L_\delta$$

are regular (i.e. recognizable by synchronous automata)

Automatic structures

$\mathfrak{A} = (A, R_1, \dots, R_s)$ is automatic if there exist a regular language $L_\delta \subseteq \Sigma^*$ and a surjective function $h : L_\delta \to A$ such that the relations

$$L_= := \{(u, v) : h(u) = h(v)\} \subseteq L_\delta \times L_\delta$$
$$L_{R_i} := \{(u_1, \dots, u_r) : \mathfrak{A} \models R_i h(u_1) \dots h(u_r)\} \subseteq L_\delta \times \dots \times L_\delta$$

are regular (i.e. recognizable by synchronous automata)

Automatic presentation of $\mathfrak{A}$: list of automata

$$\langle M_\delta, M_=, M_{R_1}, \dots, M_{R_s} \rangle$$

recognizing $L_\delta, L_=, L_{R_1}, \dots, L_{R_s}$.

(Khoussainov-Nerode, Blumensath, Blumensath-G.)

Synchronous automata

Automaton M, recognizing a relation $R \subseteq \underbrace{\Sigma^* \times \cdots \times \Sigma^*}_{r}$:

works on alphabet $\Gamma := (\Sigma \cup \{\Box\})^r - \{\Box\}^r$

$$(u_1, \ldots, u_r) \in R \qquad \Longleftrightarrow$$

$$M \text{ accepts} \quad \underbrace{\begin{matrix} u_{11}u_{12} & \ldots & u_{1j}\Box\Box\Box\ldots\Box \\ u_{21}u_{22}\ldots & \ldots & u_{2k}\Box\ldots\Box \\ \vdots & \vdots & \vdots \\ u_{i1}u_{i_2} & \ldots & u_{i\ell} \\ \vdots & \vdots & \vdots \\ u_{r1}u_{r2} & \ldots & u_{rj}\Box\Box\Box\ldots\Box \end{matrix}}_{\ell=\max\{|u_i|:i=1,\ldots,r\}} \in \Gamma^*$$

Examples of automatic structures

- $(\mathbb{N}, +)$ is automatic

  – $L_\delta = \{0,1\}^*1 \cup \{0\}$

  – $h(w_0 \dots w_{n-1}) = \sum_{i<n} w_i 2^i$ (h injective)

  – L_+ recognised by automaton M_+

  scans
  $$\begin{array}{l} u_0 u_1 \dots \quad \dots u_m \square \dots\dots \square \\ v_0 v_1 \dots \qquad\qquad \dots v_{n-1} \square \\ w_0 w_1 \dots \qquad\qquad\qquad \dots w_m \end{array}$$

  remembering carry bit c_i for $u_0 \dots u_{i-1} + v_0 \dots v_{i-1}$

  checks whether $w_i = u_i + v_i + c_i \pmod 2$

- every finite structure is automatic

- the configuration graphs of Turing machines are automatic

Examples of automatic structures

- $(\mathbb{N}, +, |_m)$ is automatic

$$x \,|_m\, y \qquad :\Longleftrightarrow \qquad x \text{ is a power of } m \text{ dividing } y$$

use m-ary representation of numbers

$$L_{|_m} = \left\{ \begin{matrix} u \\ v \end{matrix} : \begin{matrix} u \\ v \end{matrix} = \begin{matrix} 0 \dots & 0\,1\,\square \dots & \dots \square \\ 0 \dots & 0 v_r v_{r+1} & \dots v_n \end{matrix} \right\}$$

- $\text{Tree}(m) = (\{0, \dots, m-1\}^*, \sigma_0, \dots, \sigma_{m-1}, \leq, \text{el})$ is automatic

  - $\sigma_i : \quad u \mapsto ui$
  - $u \leq v : \quad \exists w \quad uw = v$
  - $\text{el}(u, v) : \quad |u| = |v|$

Automatic groups

$(G, \cdot)$ countable group with set S of semigroup generators

$\Longrightarrow$ canonical surjective map $h : S^* \to G$

Automatic groups

$(G, \cdot)$ countable group with set S of semigroup generators

$\Longrightarrow$ canonical surjective map $h : S^* \to G$

Cayley graph: $\Gamma(G, S) = (G, (\xrightarrow{s})_{s \in S})$

vertices: $g \in G$, edges: $g \xrightarrow{s} h$ iff $g \cdot s = h$

Automatic groups

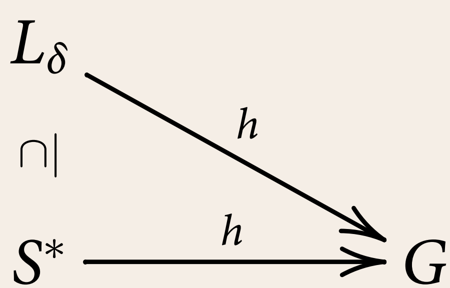
$(G, \cdot)$ countable group with set S of semigroup generators

$\Longrightarrow$ canonical surjective map $h : S^* \to G$

Cayley graph: $\Gamma(G, S) = (G, (\xrightarrow{s})_{s \in S})$

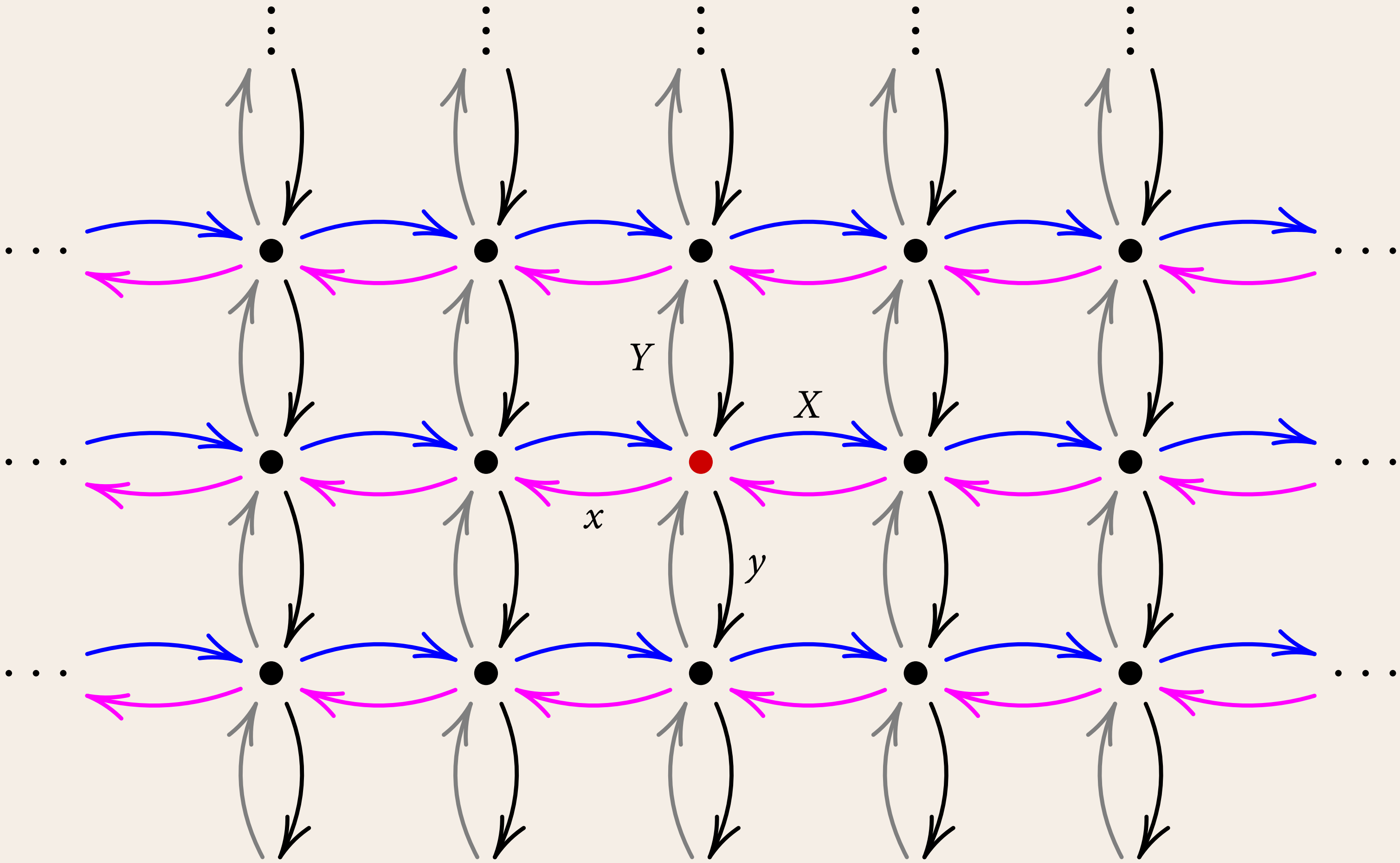
vertices: $g \in G$, edges: $g \xrightarrow{s} h$ iff $g \cdot s = h$

$(G, \cdot)$ is an **automatic group** if there is a finite set $S \subseteq G$ of semigroup generators, such that $\Gamma(G, S)$ is an **automatic structure** with presentation $\langle L_\delta, h, \dots \rangle$ where $L_\delta \subseteq S^*$ and



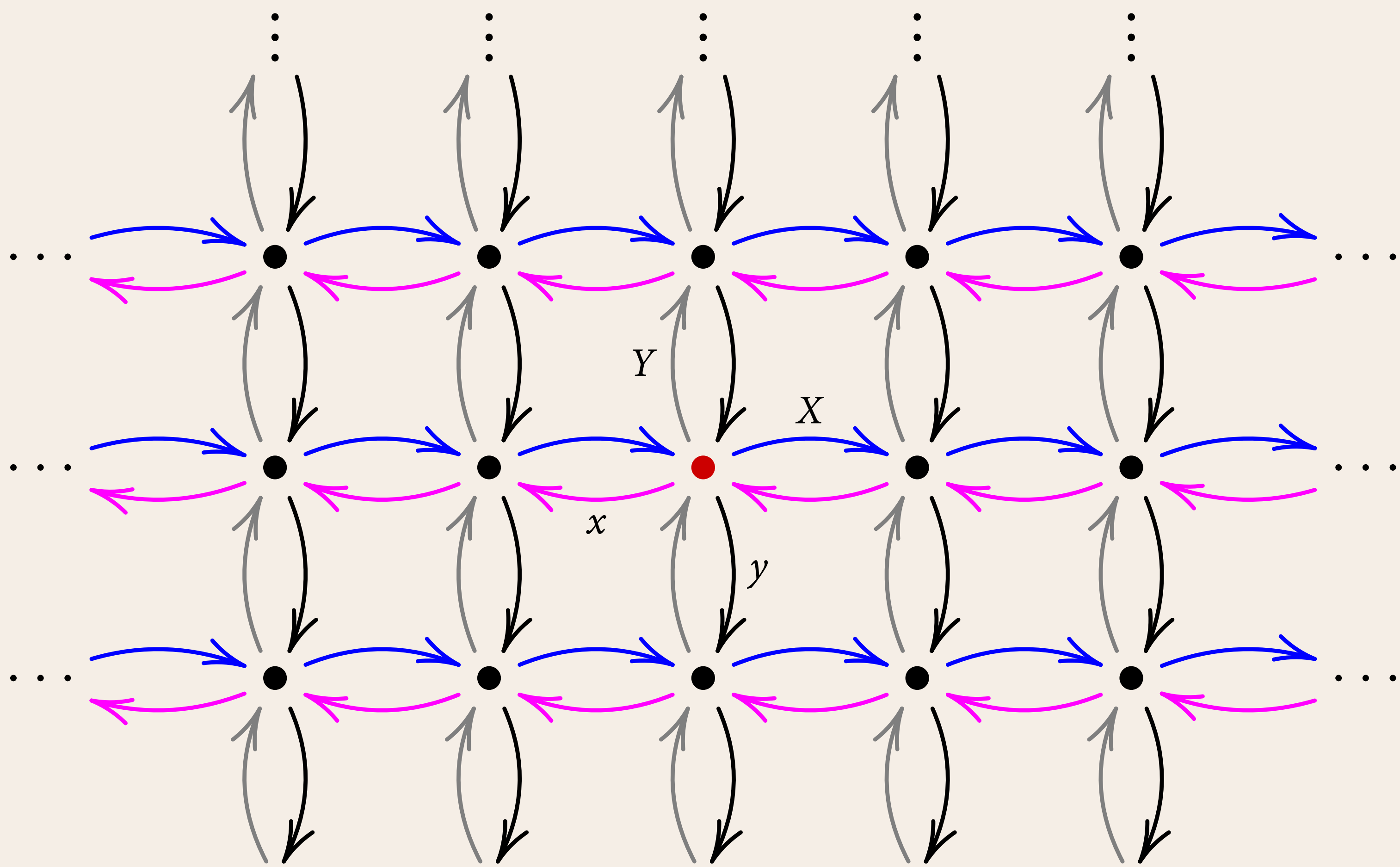
Automatic groups: Example

$(\mathbb{Z} \times \mathbb{Z}, +)$, with $S = \{X, Y, x, y\}$, $x = X^{-1}, y = Y^{-1}$



Automatic groups: Example

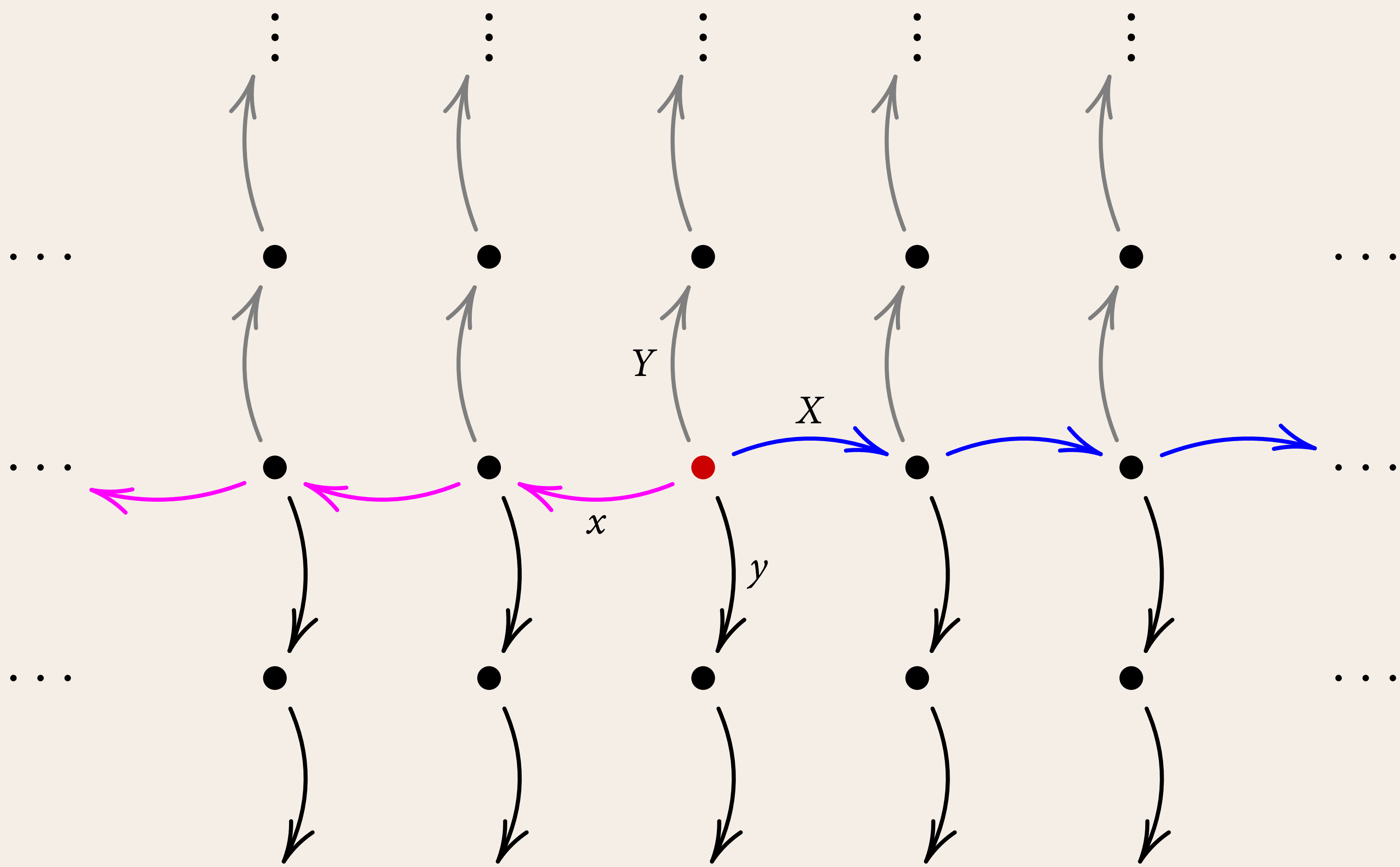
$(\mathbb{Z} \times \mathbb{Z}, +)$, with $S = \{X, Y, x, y\}$, $x = X^{-1}, y = Y^{-1}$



$$L_\delta = (X^* \cup x^*)(Y^* \cup y^*)$$

Automatic groups: Example

$(\mathbb{Z} \times \mathbb{Z}, +)$, with $S = \{X, Y, x, y\}$, $x = X^{-1}, y = Y^{-1}$

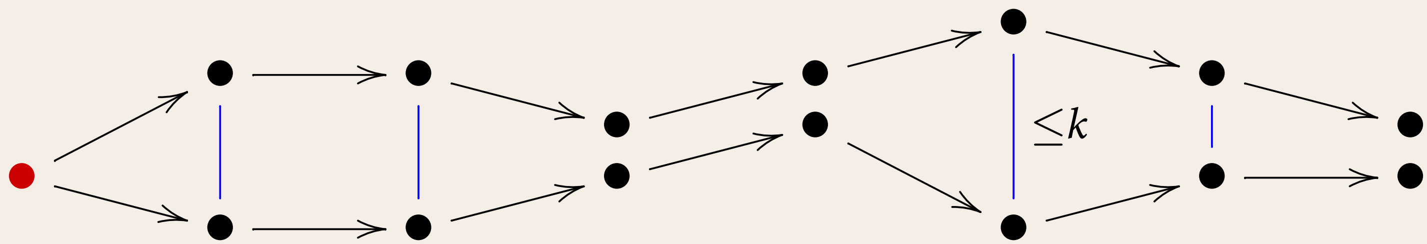


$$L_\delta = (X^* \cup x^*)(Y^* \cup y^*)$$

The k-fellow traveller property

$(G, \cdot)$ automatic group, with automatic presentation $h : L_\delta \to G$.

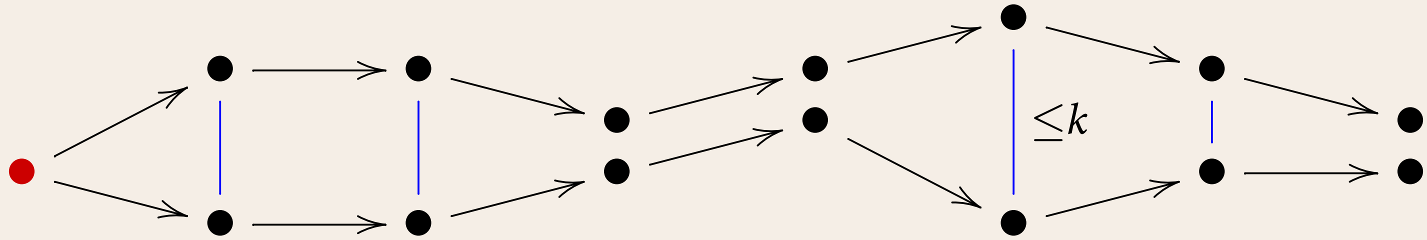
For all $u, v \in L_\delta$, if $\mathrm{dist}(u, v) \leq 1$ in $\Gamma(G, S)$, then
$\mathrm{dist}(u_1 \dots u_i, v_1 \dots v_i) \leq k$ for all $i \leq \max(|u|, |v|)$.



The k-fellow traveller property

$(G, \cdot)$ automatic group, with automatic presentation $h : L_\delta \to G$.

For all $u, v \in L_\delta$, if $\mathrm{dist}(u, v) \leq 1$ in $\Gamma(G, S)$, then
$\mathrm{dist}(u_1 \dots u_i, v_1 \dots v_i) \leq k$ for all $i \leq \max(|u|, |v|)$.



Proposition. $(G, \cdot)$ is automatic $\iff$ for some S and k, there is a regular language $L_\delta \subseteq S^*$ such that the canonical map $h : L_\delta \to G$ is surjective and satisfies the k-fellow traveller property.

Automatic groups versus automatic Caley graphs

By definition, if $(G, \cdot)$ is an automatic group, then for some S, the Cayley graph $\Gamma(G, S)$ is an automatic graph.

The converse is not true!

Counterexample: (Senizergues) The Heisenberg group H is the group of affine transformations of $\mathbb{Z}^3$ generated by $S = \{\alpha, \beta, \gamma\}$.

$$\alpha : (x, y, z) \mapsto (x + 1, y, z + y)$$

$$\beta : (x, y, z) \mapsto (x + 1, y + 1, z)$$

$$\gamma : (x, y, z) \mapsto (x, y, z + 1)$$

- Obviously, $\Gamma(H, S)$ is first-order interpretable in $(\mathbb{N}, +)$. Hence, $\Gamma(H, S)$ is an automatic graph.
- But H is not an automatic group (Epstein et al.).

ω-automatic structures

$\mathfrak{A} = (A, R_1, \dots, R_s)$ is ω-automatic if there exist a ω-regular language $L_\delta \subseteq \Sigma^\omega$ and a surjective function $h : L_\delta \to A$ such that the relations

$$L_= := \{(u, v) : h(u) = h(v)\} \subseteq L_\delta \times L_\delta$$

$$L_{R_i} := \{(u_1, \dots, u_r) : \mathfrak{A} \models R_i h(u_1) \dots h(u_r)\} \subseteq L_\delta \times \dots \times L_\delta$$

are ω-regular, i.e. recognizable by synchronous Büchi automata.

ω-automatic structures

$\mathfrak{A} = (A, R_1, \ldots, R_s)$ is ω-automatic if there exist a ω-regular language $L_\delta \subseteq \Sigma^\omega$ and a surjective function $h : L_\delta \to A$ such that the relations

$$L_= := \{(u, v) : h(u) = h(v)\} \subseteq L_\delta \times L_\delta$$

$$L_{R_i} := \{(u_1, \ldots, u_r) : \mathfrak{A} \models R_i h(u_1) \ldots h(u_r)\} \subseteq L_\delta \times \cdots \times L_\delta$$

are ω-regular, i.e. recognizable by synchronous Büchi automata.

- every automatic structure is ω-automatic
- $(\mathbb{R}, +)$ and $(\mathbb{R}, +, \leq, |_m, 1)$ are ω-automatic

$$x \mid_m y \quad :\Longleftrightarrow \quad \exists k, r \in \mathbb{Z} : x = m^k, \quad y = r \cdot x$$

- ω-Tree$(m) = (\{0, \ldots, m-1\}^{\leq\omega}, \sigma_0, \ldots, \sigma_{m-1}, \leq, \mathrm{el})$ is ω-automatic

First-order logic on ω-automatic structures

$\mathrm{FO}(\exists^\omega)$: FO + "$\exists$ infinitely many x such that …"

Theorem. Given $\varphi(\bar{x}) \in \mathrm{FO}(\exists^\omega)$ and an ω-automatic structure $\mathfrak{A}$, one can effectively compute an automatic presentation of $(\mathfrak{A}, \varphi^{\mathfrak{A}})$.

First-order logic on ω-automatic structures

$\mathrm{FO}(\exists^\omega)$: FO + "$\exists$ infinitely many x such that ... "

Theorem. Given $\varphi(\overline{x}) \in \mathrm{FO}(\exists^\omega)$ and an ω-automatic structure $\mathfrak{A}$, one can effectively compute an automatic presentation of $(\mathfrak{A}, \varphi^{\mathfrak{A}})$.

Regular and ω-regular relations are closed under

- first-order operations: classical automata theory
- the quantifier $\exists^\omega$:
  for regular relations, this follows from the Pumping Lemma
  for ω-regular relations, more complicated arguments needed

Corollary.
The $\mathrm{FO}(\exists^\omega)$-theory of every ω-automatic structure is decidable.

Query evaluation and model checking

Query evaluation: (for a logic L on (ω)-automatic structures)

Given: $\varphi(\overline{x}) \in L$, and an automatic presentation of $\mathfrak{A}$

Compute: a presentation of $\varphi^{\mathfrak{A}} := \{\overline{a} : \mathfrak{A} \models \varphi(\overline{a})\}$

(compatible with presentation of $\mathfrak{A}$)

Model checking:

Given: $\varphi(\overline{x}) \in L$, a presentation of $\mathfrak{A}$ and $\overline{a}$

Decide: $\mathfrak{A} \models \varphi(\overline{a})$?

Query evaluation and model checking

Query evaluation: (for a logic L on (ω)-automatic structures)

Given: $\varphi(\overline{x}) \in L$, and an automatic presentation of $\mathfrak{A}$

Compute: a presentation of $\varphi^{\mathfrak{A}} := \{\overline{a} : \mathfrak{A} \models \varphi(\overline{a})\}$
(compatible with presentation of $\mathfrak{A}$)

Model checking:

Given: $\varphi(\overline{x}) \in L$, a presentation of $\mathfrak{A}$ and $\overline{a}$

Decide: $\mathfrak{A} \models \varphi(\overline{a})$?

structure complexity: For fixed φ, determine complexity of $\mathfrak{A} \mapsto \varphi^{\mathfrak{A}}$
in terms of size of deterministic automata representing $\mathfrak{A}$.

expression complexity: For fixed $\mathfrak{A}$, determine complexity of
$\mathfrak{A} \mapsto \varphi^{\mathfrak{A}}$ or $\mathrm{Th}_L(\mathfrak{A})$ in terms of length of φ

combined complexity: both inputs variable

More powerful logics than FO

Query evaluation and model checking are undecidable for

- transitive closure logics
- fixed point logics (μ-calculus, LFP, . . .)
- FO + counting
- monadic second-order logic

on certain fixed automatic structures.

More powerful logics than FO

Query evaluation and model checking are undecidable for

- transitive closure logics
- fixed point logics (μ-calculus, LFP, ...)
- FO + counting
- monadic second-order logic

on certain fixed automatic structures.

- define multiplication in $(\mathbb{N}, +)$
- configuration graphs of Turing machines are automatic.

Any logic that is strong enough for REACHABILITY can express the halting problem.

Complexity

There are automatic structures with non-elementary FO-theories.

Examples: $(\mathbb{N}, +, |_m)$, $\mathrm{Tree}(m)$

Complexity

There are automatic structures with non-elementary FO-theories.

Examples: $(\mathbb{N}, +, |_m)$, $\mathrm{Tree}(m)$

Simple fragments of (relational) FO:

	structure complexity	expression complexity
Model Checking		
quantifier-free	LOGSPACE	ALOGTIME
existential	NP	PSPACE
Query-Evaluation		
quantifier-free	LOGSPACE	PSPACE
existential	PSPACE	EXPSPACE

Automatic structures with functions

Model checking complexity of quantifier-free formulae with functions:

structure complexity: NLOGSPACE-complete

expression complexity: PTIME-complete and solvable in time $O(|\varphi|^2)$

Automatic structures with functions

Model checking complexity of quantifier-free formulae with functions:

structure complexity: NLOGSPACE-complete
expression complexity: PTIME-complete and solvable in time $O(|\varphi|^2)$

Corollary. The word problem of any automatic group can be solved in quadratic time.

$(G, \cdot)$ automatic group, generated by $\{s_1, \ldots, s_m\}$

$\Rightarrow \quad G' := (G, e, g \mapsto gs_1, \ldots, g \mapsto gs_m)$ is automatic structure

Word problem for $(G, \cdot)$ described by term equations over G'.

The isomorphism problem

Theorem.

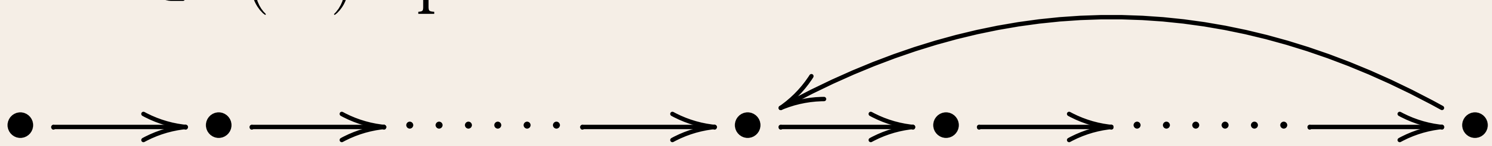
The isomorphism problem for automatic structures is undecidable.

The isomorphism problem

Theorem.
The isomorphism problem for automatic structures is undecidable.

Proof. Every deterministic TM M can be effectively translated into TM M' whose configuration graph $C(M')$ contains

- ω many copies of $(\mathbb{N}, \mathrm{succ})$
- for each $x \in L(M)$ a path



Hence, $$L(M) = \varnothing \iff \underbrace{C(M')}_{\text{automatic}} \cong \underbrace{\omega \cdot (\mathbb{N}, \mathrm{succ})}_{\text{automatic}}$$

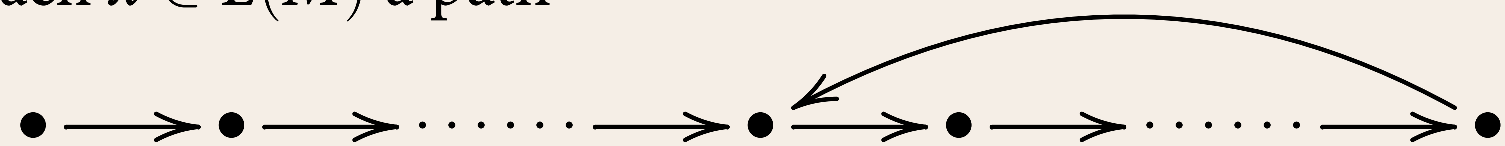
The isomorphism problem

Theorem.

The isomorphism problem for automatic structures is undecidable.

Proof. Every deterministic TM M can be effectively translated into TM M' whose configuration graph $C(M')$ contains

- ω many copies of $(\mathbb{N}, \mathrm{succ})$
- for each $x \in L(M)$ a path



Hence, $$L(M) = \varnothing \iff \underbrace{C(M')}_{\text{automatic}} \cong \underbrace{\omega \cdot (\mathbb{N}, \mathrm{succ})}_{\text{automatic}}$$

Theorem.

The connectivity problem for automatic graphs is undecidable.

Automatic structures and interpretations

$\mathfrak{A} \leq_{\mathrm{FO}} \mathfrak{B}$: $\mathfrak{A}$ is first-order interpretable in $\mathfrak{B}$

Automatic structures and ω-automatic structures are closed under FO-interpretations:

$$\mathfrak{B} \text{ is } (\omega)\text{-automatic}, \quad \mathfrak{A} \leq_{\mathrm{FO}} \mathfrak{B} \quad \Longrightarrow \quad \mathfrak{A} \text{ is } (\omega)\text{-automatic}$$

Automatic structures and interpretations

$\mathfrak{A} \leq_{\mathrm{FO}} \mathfrak{B}$: $\mathfrak{A}$ is first-order interpretable in $\mathfrak{B}$

Automatic structures and ω-automatic structures are closed under FO-interpretations:

$$\mathfrak{B} \text{ is } (\omega)\text{-automatic}, \quad \mathfrak{A} \leq_{\mathrm{FO}} \mathfrak{B} \quad \Longrightarrow \quad \mathfrak{A} \text{ is } (\omega)\text{-automatic}$$

In particular, the (ω)-automatic structures are closed under

- expansion by definable relations
- factorisation by definable congruences
- substructures with definable universe
- finite powers

Note: They are not closed under taking arbitrary substructures

Model theoretic characterisation of automatic structures

Theorem. The following are equivalent:

(1) $\mathfrak{A}$ is automatic

(2) $\mathfrak{A} \leq_{\mathrm{FO}} (\mathbb{N}, +, |_m)$ for some (and hence all) $m \geq 2$

(3) $\mathfrak{A} \leq_{\mathrm{FO}} \mathrm{Tree}(m)$ for some (and hence all) $m \geq 2$

Model theoretic characterisation of automatic structures

Theorem. The following are equivalent:

(1) $\mathfrak{A}$ is automatic

(2) $\mathfrak{A} \leq_{\mathrm{FO}} (\mathbb{N}, +, |_m)$ for some (and hence all) $m \geq 2$

(3) $\mathfrak{A} \leq_{\mathrm{FO}} \mathrm{Tree}(m)$ for some (and hence all) $m \geq 2$

Theorem. The following are equivalent:

(1) $\mathfrak{A}$ is ω-automatic

(2) $\mathfrak{A} \leq_{\mathrm{FO}} (\mathbb{R}, +, \leq, |_m, 1)$ for some (and hence all) $m \geq 2$

(3) $\mathfrak{A} \leq_{\mathrm{FO}} \omega\text{-}\mathrm{Tree}(m)$ for some (and hence all) $m \geq 2$

Characterising automatic groups

Theorem. $(G, \cdot)$ is an automatic group

$$\Longleftrightarrow$$

there is finite set $S \subseteq G$ of semigroup generators such that Caley graph $\Gamma(G, S)$ is FO-definable in $\mathrm{Tree}(S)$.

Characterising automatic groups

Theorem. $(G, \cdot)$ is an automatic group

$$\Longleftrightarrow$$

there is finite set $S \subseteq G$ of semigroup generators such that Caley graph $\Gamma(G, S)$ is FO-definable in $\mathrm{Tree}(S)$.

There exist first-order formulae $D(x), E(x, y), \varphi_1(x, y), \ldots, \varphi_m(x, y)$ of vocabulary $\{s_1, \ldots, s_m, \leq, \mathrm{el}\}$ such that

$$\langle D^{\mathrm{Tree}(S)}, \varphi_1^{\mathrm{Tree}(S)}, \ldots, \varphi_m^{\mathrm{Tree}(S)} \rangle / E^{\mathrm{Tree}(S)} \quad = \quad \Gamma(G, S)$$

(equality rather than isomorphism)

Structures that are not automatic

How to prove that a structure $\mathfrak{A}$ is not automatic ?

1) $\mathfrak{A}$ not countable: $(\mathbb{R}, +, , \cdot)$

2) $\mathrm{Th}(\mathfrak{A})$ undecidable: $(\mathbb{N}, +, , \cdot)$

3) Growth rates

Take automatic presentation of $\mathfrak{A}$ with bijective $h : L_\delta \longrightarrow A$

For any set $a_1, a_2, \dots$ of definable elements in $\mathfrak{A}$ (ordered by $|h^{-1}(a_i)|$), and any finite set F of definable functions on $\mathfrak{A}$, let

$$G_1 := \{a_1\}, \qquad G_{n+1} := G_n \cup \{a_{n+1}\} \cup \bigcup_{f \in F} f(G_n \times \cdots \times G_n)$$

Theorem. $|G_n| = 2^{O(n)}$ for all n

Elements of G_n are represented by words of length $O(n)$

Application

Corollary. $(\mathbb{N}, \cdot)$ is not automatic.

For $a_1, a_2, \ldots$ enumeration of primes, $f(x, y) = x \cdot y$

$$|G_n| = 2^{\Omega(n^2)}$$

But: $(\mathbb{N}. \cdot)$ is tree automatic