Automata-based presentations of infinite structures

Vince Bárány[1] and Erich Grädel[2] and Sasha Rubin[3]

[1] Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford OX1 3QD, United Kingdom
`vbarany@logic.rwth-aachen.de`
[2] Mathematical Foundations of Computer Science
RWTH Aachen, D-52056 Aachen, Germany
`graedel@logic.rwth-aachen.de`
[3] Department of Mathematics and Applied Mathematics
University of Cape Town, Private Bag, Rondebosch 7701, South Africa
`srubin@math.cornell.edu`

1

Automata-based presentations of infinite structures

Vince Bárány[1] and Erich Grädel[2] and Sasha Rubin[3]

1.1 Finite presentations of infinite structures

The model theory of finite structures is intimately connected to various fields in computer science, including complexity theory, databases, and verification. In particular, there is a close relationship between complexity classes and the expressive power of logical languages, as witnessed by the fundamental theorems of descriptive complexity theory, such as Fagin's Theorem and the Immerman-Vardi Theorem (see [78, Chapter 3] for a survey).

However, for many applications, the strict limitation to finite structures has turned out to be too restrictive, and there have been considerable efforts to extend the relevant logical and algorithmic methodologies from finite structures to suitable classes of infinite ones. In particular this is the case for databases and verification where infinite structures are of crucial importance [130]. *Algorithmic model theory* aims to extend in a systematic fashion the approach and methods of finite model theory, and its interactions with computer science, from finite structures to finitely-presentable infinite ones.

There are many possibilities to present infinite structures in a finite manner. A classical approach in model theory concerns the class of *computable structures*; these are countable structures, on the domain of nat-

[1] Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford OX1 3QD, United Kingdom
`vbarany@logic.rwth-aachen.de`
[2] Mathematical Foundations of Computer Science
RWTH Aachen, D-52056 Aachen, Germany
`graedel@logic.rwth-aachen.de`
[3] Department of Mathematics and Applied Mathematics
University of Cape Town, Private Bag, Rondebosch 7701, South Africa
`srubin@math.cornell.edu`

ural numbers, say, with a finite collection of computable functions and relations. Such structures can be finitely presented by a collection of algorithms, and they have been intensively studied in model theory since the 1960s. However, from the point of view of algorithmic model theory the class of computable structures is problematic. Indeed, one of the central issues in algorithmic model theory is the effective evaluation of logical formulae, from a suitable logic such as first-order logic (FO), monadic second-order logic (MSO), or a fixed point logic like LFP or the modal μ-calculus. But on computable structures, only the quantifier-free formulae generally admit effective evaluation, and already the existential fragment of first-order logic is undecidable, for instance on the computable structure $(\mathbb{N}, +, \cdot)$.

This leads us to the central requirement that for a suitable logic L (depending on the intended application) the model-checking problem for the class $\mathcal{C}$ of finitely presented structures should be algorithmically solvable. At the very least, this means that the L-theory of individual structures in $\mathcal{C}$ should be decidable. But for most applications somewhat more is required:

Effective semantics: There should be an algorithm that, given a finite presentation of a structure $\mathfrak{A} \in \mathcal{C}$ and a formula $\psi(\bar{x}) \in L$, expands the given presentation to include the relation $\psi^{\mathfrak{A}}$ defined by ψ on $\mathfrak{A}$.

This also implies that the class $\mathcal{C}$ should be closed under some basic operations (such as logical interpretations). Thus we should be careful to restrict the model of computation. Typically, this means using some model of *finite automata* or a very restricted form of rewriting.

In general, the finite means for presenting infinite structures may involve different approaches: logical interpretations; finite axiomatisations; rewriting of terms, trees, or graphs; equational specifications; the use of synchronous or asynchronous automata, etc. The various possibilities can be classified along the following lines:

Internal: a set of finite or infinite words or trees/terms is used to represent the domain of (an isomorphic copy of) the structure. Finite automata/rewriting-rules compute the domain and atomic relations (eg. prefix-recognisable graphs, automatic structures).

Algebraic: a structure is represented as the least solution of a finite set of recursive equations in an appropriately chosen algebra of finite and countable structures (eg. VR-equational structures).

Logical: structures are described by interpreting them, using a finite collection of formulae, in a fixed structure (eg. tree-interpretable structures). A different approach consists in (recursively) axiomatising the isomorphism class of the structure to be represented.

Transformational: structures are defined by sequences of prescribed transformations, such as graph-unraveling, or Muchnik's iterations, applied to certain fixed initial structures (which are already known to have a decidable theory). Transformations can also be transductions, logical interpretations, etc. [23]

The last two approaches overlap somewhat. Also, the algebraic approach can be viewed *generatively*: convert the equational system into an appropriate *deterministic grammar* generating the solution of the original equations [44]. The grammar is thus the finite presentation of the graph. One may also say that internal presentations and generating grammars provide descriptions of the *local structure* from which the whole arises, as opposed to descriptions based on *global symmetries* typical of algebraic specifications.

Prerequisites and notation

We assume rudimentary knowledge of finite automata on finite and infinite words and trees, their languages and their correspondence to monadic second-order logic (MSO) [133, 79]. Undefined notions from logic and algebra (congruence on structures, definability, isomorphism) can be found in any standard textbook. We mainly consider the following logics $\mathcal{L}$: first-order (FO), monadic second order (MSO), and weak monadic second-order (wMSO) which has the same syntax as MSO, but the intended interpretation of the set variables is that they range over *finite* subsets of the domain of the structure under consideration.

We mention the following to fix notation: infinite words are called ω-words and infinite trees are called ω-trees (to distinguish them from finite ones); relations computable by automata will be called *regular*; the domain of a *structure* $\mathfrak{B}$ is usually written B and its relations are written $R^{\mathfrak{B}}$. An MSO-formula $\phi(X_1, \cdots, X_j, x_1, \cdots, x_k)$ interpreted in $\mathfrak{B}$ *defines* the set $\phi^{\mathfrak{B}} := \{(B_1, \cdots, B_j, b_1, \cdots, b_k) \mid B_i \subset B, b_i \in B, \mathfrak{B} \models \phi(B_1, \cdots, B_j, b_1, \cdots, b_k)\}$. A wMSO-formula is similar except that the B_i range over finite subsets of B. The *full binary tree* $\mathfrak{T}_2$ is defined as the structure

$$(\{0,1\}^*, \mathtt{suc}_0, \mathtt{suc}_1)$$

where the successor relation $\mathtt{suc}_i$ consists of all pairs (x, xi). Tree automata operate on Σ-*labelled trees* $T : \{0,1\}^* \to \Sigma$. Such a tree is identified with the structure

$$\left(\{0,1\}^*, \mathtt{suc}_0, \mathtt{suc}_1, \{T^{-1}(\sigma)\}_{\sigma \in \Sigma}\right).$$

Rabin proved the decidability of the MSO-theory of $\mathfrak{T}_2$ and the following fundamental correspondence between MSO and tree automata (see [132] for an overview):

For every monadic second-order formula $\varphi(\overline{X})$ in the signature of $\mathfrak{T}_2$ there is a tree automaton $\mathcal{A}$ (and vice versa) such that

$$L(\mathcal{A}) = \{T_{\overline{X}} \mid \mathfrak{T}_2 \models \varphi(\overline{X})\} \tag{1.1}$$

where $T_{\overline{X}}$ denotes the tree with labels for each X_i.

Similar definitions and results hold for r-ary trees, in which case the domain is $[r]^*$ where $[r] := \{0, \cdots, r-1\}$, and finite trees.

In section 1.2.2 and elsewhere we do not distinguish between a term and its natural representation as a tree. Thus we may speak of infinite terms. We consider countable, vertex- and edge-labelled graphs possibly having distinguished vertices (called sources), and no parallel edges of the same label. A graph is *deterministic* if each of its vertices is the source of at most one edge of each edge label.

Interpretations

Interpretations allow one to define an isomorphic copy of one structure in another. Fix a logic $\mathcal{L}$. A d-dimensional $\mathcal{L}$-*interpretation* $\mathcal{I}$ of structure $\mathfrak{B} = (B; (R_i^{\mathfrak{B}})_i)$ in structure $\mathfrak{A}$, denoted $\mathfrak{B} \leq_{\mathcal{L}}^{\mathcal{I}} \mathfrak{A}$, consists of the following $\mathcal{L}$-formulas in the signature of $\mathfrak{A}$,

- a domain formula $\Delta(\overline{x})$,
- a relation formula $\Phi_{R_i}(\overline{x}_1, \cdots, \overline{x}_{r_i})$ for each relation symbol R_i, and
- an equality formula $\epsilon(\overline{x}_1, \overline{x}_2)$,

where each $\Phi_{R_i}^{\mathfrak{A}}$ is a relation on $\Delta^{\mathfrak{A}}$, each of the tuples $\overline{x}_i, \overline{x}$ contain the same number of variables, d, and $\epsilon^{\mathfrak{A}}$ is a congruence on the structure $(\Delta^{\mathfrak{A}}, (\Phi_{R_i}^{\mathfrak{A}})_i)$, so that $\mathfrak{B}$ is isomorphic to

$$(\Delta^{\mathfrak{A}}, (\Phi_{R_i}^{\mathfrak{A}})_i) / \epsilon^{\mathfrak{A}}.$$

If $\mathcal{L}$ is FO then the free $\overline{x}$ are FO and we speak of a *FO interpretation*. If $\mathcal{L}$ is MSO (wMSO) but the free variables are FO, then we speak of a *(weak) monadic second-order interpretation*.

We associate with $\mathcal{I}$ a transformation of formulas $\psi \mapsto \psi^{\mathcal{I}}$. For illustration we define it in the first-order case: the variable x_i is replaced by the d-tuple $\overline{y}_i$, $(\psi \vee \phi)^{\mathcal{I}}$ by $\psi^{\mathcal{I}} \vee \phi^{\mathcal{I}}$, $(\neg\psi)^{\mathcal{I}}$ by $\neg\psi^{\mathcal{I}}$, $(\exists x_i \psi)^{\mathcal{I}}$ by $\exists \overline{y}_i \Delta(\overline{y}_i) \wedge \psi^{\mathcal{I}}$, and $(x_i = x_j)^{\mathcal{I}}$ is replaced by $\epsilon(\overline{y}_i, \overline{y}_j)$. Thus one can translate $\mathcal{L}$ formulas from the signature of $\mathfrak{B}$ into the signature of $\mathfrak{A}$.

Proposition 1.1.1 *If $\mathfrak{B} \leq_{\mathcal{L}}^{\mathcal{I}} \mathfrak{A}$, say the isomorphism is f, then for every formula $\psi(x_1, \cdots, x_k)$ in the signature of $\mathfrak{B}$ and all k-tuples $\overline{b}$ of elements of $\mathfrak{B}$ it holds that*

$$\mathfrak{B} \models \psi(b_1, \cdots, b_k) \iff \mathfrak{A} \models \psi^{\mathcal{I}}(f(b_1), \cdots, f(b_k))$$

In particular, if $\mathfrak{A}$ has decidable $\mathcal{L}$-theory, then so does $\mathfrak{B}$.

Set interpretations

When $\mathcal{L}$ is MSO (wMSO) and the free variables are MSO (wMSO) the interpretation is called a *(finite) set interpretation*. In this last case, we use the notation $\mathfrak{B} \leq_{\text{set}}^{\mathcal{I}} \mathfrak{A}$ or $\mathfrak{B} \leq_{\text{fset}}^{\mathcal{I}} \mathfrak{A}$. We will only consider (finite) set interpretations of dimension 1.

If finiteness of sets is MSO-definable in some structure $\mathfrak{A}$ (as for linear orders or for finitely branching trees) then every structure $\mathfrak{B}$ having a finite-set interpretation in $\mathfrak{A}$ can also be set interpreted in $\mathfrak{A}$.

Example 1.1.2 An interpretation $(\mathbb{N}, +) \leq_{\text{fset}}^{\mathcal{I}} (\mathbb{N}, 0, \mathtt{suc})$ based on the binary representation is given by $\mathcal{I} = (\varphi(X), \varphi_+(X, Y, Z), \varphi_=(X, Y))$ with $\varphi(X)$ always true, $\varphi_=$ the identity, and $\varphi_+(X, Y, Z)$ is

$$\exists C \, \forall n \, [(Zn \leftrightarrow Xn \oplus Yn \oplus Cn) \wedge (C(\mathtt{suc}n) \leftrightarrow \mu(Xn, Yn, Cn)) \wedge \neg C0]$$

where C stands for carry, $\oplus$ is exclusive or, and $\mu(x_0, x_1, x_2)$ is the majority function, in this case definable as $\bigvee_{i \neq j} x_i \wedge x_j$.

To every (finite) subset interpretation $\mathcal{I}$ we associate, as usual, a transformation of formulas $\psi \mapsto \psi^{\mathcal{I}}$, in this case mapping first-order formulas to (weak) monadic second-order formulas.

Proposition 1.1.3 *Let $\mathfrak{B} \leq_{\text{(f)set}}^{\mathcal{I}} \mathfrak{A}$ be a (finite) subset interpretation with isomorphism f. Then to every first-order formula $\psi(x_1, \cdots, x_k)$ in the signature of $\mathfrak{B}$ one can effectively associate a (weak) monadic second-order formula $\psi^{\mathcal{I}}(X_1, \cdots, X_k)$ in the signature of $\mathfrak{A}$ such that for all k-tuples $\overline{b}$ of elements of $\mathfrak{B}$ it holds that*

$$\mathfrak{B} \models \psi(b_1, \cdots, b_k) \iff \mathfrak{A} \models \psi^{\mathcal{I}}(f(b_1), \cdots, f(b_k)) \quad .$$

Consequently, if the (weak) monadic-second order theory of $\mathfrak{A}$ is decidable then so is the first-order theory of $\mathfrak{B}$.

For more on subset interpretations we refer to [23].

1.2 A hierarchy of finitely presentable structures

This section provides an overview of some of the prominent classes of graphs and their various finite presentations.

These developments are the product of over two decades of research in diverse fields. We begin our exposition with the seminal work of Muller and Schupp on context-free graphs, we mention prefix-recognisable structures, survey hyperedge-replacement and vertex-replacement grammars and their corresponding algebraic frameworks leading up to equational graphs in algebras with asynchronous or synchronous product operation. These latter structures are better known in the literature by their automatic presentations, and constitute the topic of the rest of this survey.

As a unifying approach we discuss how graphs belonging to individual classes can be characterised as least fixed-point solutions of finite systems of equations in a corresponding algebra of graphs. We illustrate on examples how to go from graph grammars through equational presentations and interpretations to internal presentations and vice versa.

We briefly summarise key results on Caucal's pushdown hierarchy and more recent developments on simply-typed recursion schemes and collapsible pushdown automata.

Figure 1.1 provides a summary of some of the graph classes discussed in this section together with the boundaries of decidability for relevant logics. Rational graphs and automatic graphs featured on this diagram are described in detail in Section 1.3.

1.2.1 From context-free graphs to prefix-recognisable structures

Context-free graphs were introduced in the seminal papers [110, 111, 112] of Muller and Schupp. There are several equivalent definitions. The objects of study are countable directed edge-labelled, finitely branching graphs. An *end* is a maximal connected[4] component of the induced subgraph obtained by removing, for some n, the n-neighbourhood of a fixed

---

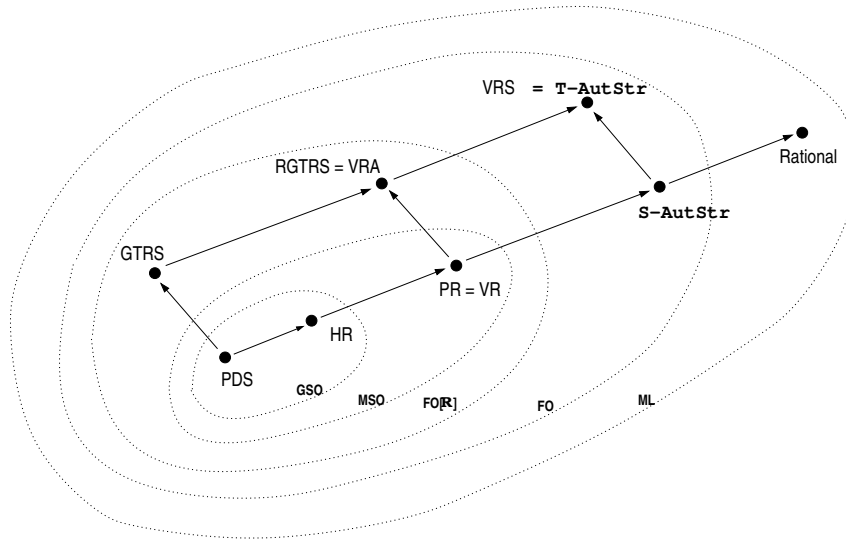[4] connectedness is taken with respect to the underlying undirected graph.

Figure 1.1 Relationship of graph classes and logical decidability boundaries.

vertex v_0. A vertex of an end is on the *boundary* if it is connected to a vertex in the removed neighbourhood. Two ends are end-isomorphic if there is a graph isomorphism (preserving labels as well) between them that is also a bijection of their boundaries. A graph is *context-free* if it is connected and has only *finitely many ends* up to end-isomorphism. This notion is independent of the v_0 chosen.

A graph is context-free if and only if it is isomorphic to the connected component of the configuration graph of a pushdown automaton (without ϵ-transitions) induced by the set of configurations that are reachable from the initial configuration [112].

A *context-free group* is a finitely generated group G such that, for some set S of semigroup generators of G, the set of words $w \in S^*$ representing the identity element of G forms a context-free language. This is independent of the choice of S. Moreover, a group is context-free if and only if its Cayley graph for some (and hence all) sets S of semigroup generators is a context-free graph. Finally, a finitely generated group is context-free if and only if it is *virtually free*, that is, if it has a free subgroup of finite index [111].[5]

---

[5] Originally [111] proved this under the assumption of *accessibility*, a notion related to group decompositions introduced by Wall who conjectured that all

Muller and Schupp have further shown that context-free graphs have a decidable MSO-theory. Indeed, every context-free graph can be MSO-interpreted in the full binary tree.

Example 1.2.1   Consider the group G given by the finite presentation $\langle\, a, b, c \mid ab,\, cc,\, acac,\, bcbc \,\rangle$. The Cayley graph $\Gamma(G, S)$ of G with respect to the set of semigroup generators $S = \{a, b, c\}$ is depicted below.
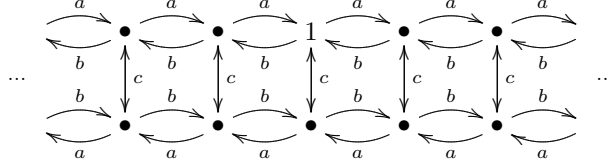
Notice that $\Gamma(G, S)$ has two ends, for any n-neighbourhood of the identity with $n > 1$. These are

   and
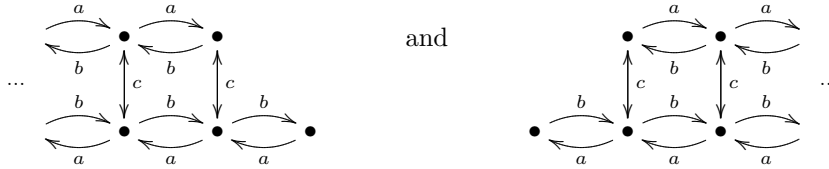
A word $w \in \{a, b, c\}^*$ represents the identity of G if, and only if, w has an even number of c's and the number of a's equals the number of b's. We present a pushdown automaton $\mathcal{A}$ which recognises this set of words and, moreover, has a configuration graph that is isomorphic to $\Gamma(G, S)$. The states of $\mathcal{A}$ are $Q = \{1, c\}$ with $q_0 = 1$ as the initial state, the stack alphabet is $\Gamma = \{a, b\}$, the input alphabet is $\{a, b, c\}$ and $\mathcal{A}$ has the following transitions:

$$
\begin{array}{rlcl}
\text{internal:} & 1\,\theta & \overset{c}{\to} & c\,\theta \\
\text{internal:} & c\,\theta & \overset{c}{\to} & 1\,\theta \\
\text{push:} & q\,\sigma\theta & \overset{\sigma}{\to} & q\,\sigma\sigma\theta \qquad \text{for } q = 1, c \text{ and } \sigma = a, b \\
\text{push:} & q\,\bot & \overset{\sigma}{\to} & q\,\sigma\bot \qquad \text{for } q = 1, c \text{ and } \sigma = a, b \\
\text{pop:} & q\,\sigma\theta & \overset{\bar{\sigma}}{\to} & q\,\theta \qquad \text{for } q = 1, c \text{ and } \{\sigma, \bar{\sigma}\} = \{a, b\}
\end{array}
$$

finitely generated groups would have this property. Muller and Schupp conjectured every context-free group to be accessible, but it was not until Dunwoody [64] proved that all finitely presentable groups are accessible that this auxiliary condition could be dropped from the characterisation of [111]. Unfortunately, many sources forget to note this fact. Later Dunwoody also gave a counterexample refuting Wall's conjecture.

Here θ is the stack content written with its top element on the left and always ending in the special symbol $\bot$ marking the bottom of the stack.

In every deterministic edge-labelled connected graph and for any ordering of the edge labels one obtains a spanning tree by taking the shortest path with the lexicographically least labeling leading to each node from a fixed source. Take such a spanning tree T for the example graph $\Gamma(G,S)$ with root 1_G. Observe that T is regular, having only finitely many subtrees (ends) up to isomorphism. The ordering $a < b < c$ induces the spanning tree depicted below. The Cayley graph $\Gamma(G,S)$ is MSO-interpretable in this regular spanning tree by defining the missing edges using the relators from the presentation of the group.
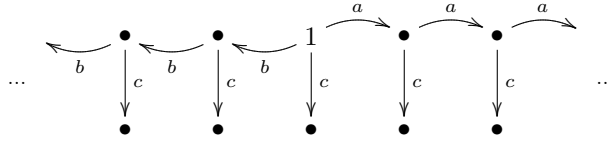
In particular $\Gamma(G,S)$ is MSO-interpretable in the full binary tree, and hence has decidable MSO.

A mild generalisation of pushdown transitions, *prefix-rewriting* rules, take the form $uz \mapsto vz$ where u and v are fixed words and z is a variable ranging over words. As in the previous example, pushdown transitions are naturally perceived as prefix-rewriting rules affecting the state and the top stack symbols. Conversely, Caucal [40] has shown that connected components of configuration graphs of prefix-rewriting systems given by finitely many prefix-rewriting rules are effectively isomorphic to connected components of pushdown graphs. Later, Caucal introduced *prefix-recognisable graphs* as a generalisation of context-free graphs and showed that these are MSO-interpretable in the full binary tree and hence have a decidable MSO-theory [42].

Definition 1.2.2 (Prefix-recognisable relations) Let Σ be a finite alphabet. The set $\mathsf{PR}(\Sigma)$ of prefix-recognisable relations over Σ^* is the smallest set of relations such that

- every regular language $L \subseteq \Sigma^*$ is a prefix-recognisable unary relation;
- if $R, S \in \mathsf{PR}$ (arities r and s) and L is regular then $L \cdot (R \times S) = \{(uv_1, \ldots, uv_r, uw_1, \ldots, uw_s) \mid u \in L, \overline{v} \in R, \overline{w} \in S\} \in \mathsf{PR}$;
- if $R \in \mathsf{PR}$ of arity $m > 1$ and $\{i_1, \ldots, i_m\} = \{1, \ldots, m\}$, then $R^{(\overline{i})} = \{(u_{i_1}, \ldots, u_{i_m}) \mid (u_1, \ldots, u_m) \in R\} \in \mathsf{PR}$;
- if $R, S \in \mathsf{PR}$ are of the same arity, then $R \cup S \in \mathsf{PR}$.

Example 1.2.3   Consider the lexicographic ordering $<_{\mathrm{lex}}$ on an ordered alphabet Σ. It is prefix-recognisable being the union of

$$\Sigma^* \cdot (\{\varepsilon\} \times \Sigma^+) \quad \text{and} \quad \Sigma^* \cdot (a\Sigma^* \times b\Sigma^*) \quad \text{for all } a < b \in \Sigma \ .$$

Following [22] we say that a structure $\mathfrak{A} = (A, \{R_i\}_i)$ is *prefix-recognizable* if A is a regular set of words over some finite alphabet Σ and each of the relations R_i is in $\mathsf{PR}(\Sigma)$. Prefix-recognisable structures can be characterized in terms of interpretations. On the basis of tree automata, it is relatively straightforward to show that the prefix-recognisable structures coincide with the structures that are MSO-interpretable in the binary tree $\mathfrak{T}_2$ [97, 42, 22]. This result has been strengthened by Colcombet [51] to first-order interpretability in the expanded structure $(\mathfrak{T}_2, \prec)$ (note that the prefix relation $\prec$ is MSO-definable but not FO definable in $\mathfrak{T}_2$). Colcombet proved that MSO-interpretations and FO-interpretations in $(\mathfrak{T}_2, \prec)$ have the same power, which gives a new characterisation of prefix-recognisable structures. We summarize these results as follows.

Theorem 1.2.4   *For every structure $\mathfrak{A}$, the following are equivalent.*

(1) $\mathfrak{A}$ is isomorphic to a prefix-recognisable structure;
(2) $\mathfrak{A}$ is MSO-interpretable in the full binary tree $\mathfrak{T}_2$;
(3) $\mathfrak{A}$ is FO-interpretable in $(\mathfrak{T}_2, \prec)$.

In particular, every prefix-recognisable structure has a decidable MSO-theory.

Below we discuss further characterisations of prefix-recognisable structures in terms of vertex-replacement grammars, or as least solutions of VR-equational systems.

1.2.2 Graph grammars and graph algebras

In this section we consider vertex- and edge-labelled graphs. In formal language theory grammars generate sets of finite words. Similarly, context-free graph grammars produce sets of finite graphs - start from an initial nonterminal and rewrite nonterminal vertices and edges according to the derivation rules. Just as for languages, the set of valid derivation trees, or parse trees, forms a regular set of trees labelled by derivation rules of the graph grammar. Conversely, consider a collection Θ of graph operations — such as disjoint union, recolourings, etc. — as primitives. Every closed Θ-term t evaluates to a finite graph $[\![t]\!]$, and similarly every

Θ-term $t(\overline{x})$ evaluates to a finite graph $\llbracket t(\overline{x}) \rrbracket$ with non-terminal (hyper)-edges and/or vertices. Formally, evaluation is the unique homomorphism from the initial algebra of Θ-terms to the Θ-algebra of finite graphs with non-terminals. Each regular tree language L of closed terms thus represents a family of finite graphs $\{\llbracket t \rrbracket \mid t \in L\}$. For a concise treatment of graph grammars and finite graphs we refer to the surveys [69, 59] and the book [53].

Our focus here is on individual countable graphs generated by *deterministic* grammars via 'complete rewriting'. A suitable framework for formalising complete rewriting, in the context of term rewriting, is convergence in complete partial orders (cpo's). Since no classical order- or metric-theoretic notion of limit seems to exist for graphs, we use the more general categorical notion of colimit [11]. We outline this framework in which an infinite term (over the graph operations Θ) yields a countable graph; details may be found in [55, 11, 53].

In the category $\mathbb{G}$ of graphs and their homomorphisms every diagram of the form

$$G_0 \xrightarrow{f_0} G_1 \xrightarrow{f_1} G_2 \xrightarrow{f_2} \cdots \xrightarrow{f_{n-1}} G_n \xrightarrow{f_n} G_{n+1} \xrightarrow{f_{n+1}} \cdots$$

has a colimit G, i.e. a kind of least common extension G of the G_ns with homomorphisms $g_n : G_n \to G$ such that $g_n = g_{n+1}f_n$ for all n.[6] We assume that the graph operations in Θ determine endofunctors of $\mathbb{G}$ that are cocontinuous i.e. colimit preserving.

On the other side, take the cpo of finite and infinite terms over the signature $\Theta \cup \{\bot\}$, with the empty term $\bot$ and the extension ordering $s \sqsubseteq t$. We may turn it into a category $\mathbb{T}_{\Theta}$ with each relation $s \sqsubseteq t$ inducing a unique arrow $s \to t$. Moreover, in this category, colimits (of diagrams as above) exist and an infinite term t is the colimit of approximations $t_0 \to t_1 \to \cdots$ (think that t_i is the restriction of t to the first i levels). The evaluation mapping $\llbracket \cdot \rrbracket$ has a unique cocontinuous extension, also denoted $\llbracket \cdot \rrbracket$, mapping infinite terms to colimits of graphs.

This completes the basic description. Now consider a grammar $\mathcal{G}$ whose derivation rules $\langle X_i \mapsto t_i(\overline{X}) \rangle$ can be expressed by Θ-terms. These terms determine cocontinuous endofunctors in the category of terms $\mathbb{T}_{\Theta}$. By the Knaster-Tarski theorem the functors have a least fixed-point $\overline{G}$, which by Kleene's Theorem is attained as the colimit of the chain

---

[6] There are examples of ascending chains $G_0 \xrightarrow{f_0} G_1 \xrightarrow{f_1} \cdots$ and $G_0 \xrightarrow{g_0} G_1 \xrightarrow{g_1} \cdots$ with identical graphs but different embeddings yielding different colimits, whence there is no apparent canonical way of defining a limit knowing only that each G_n is embeddable into G_{n+1}.

$\langle \gamma^n(\overline{\emptyset}) \rangle_n$ with the natural homomorphisms. The graph *generated* by the grammar from the corresponding non-terminal X_i is defined to be the component G_i of the colimit $\overline{G}$.

Equivalently, given the system of equations $\mathcal{E}_{\mathcal{G}} = \langle X_i = t_i(\overline{X}) \rangle$ one can construct a syntactic (uninterpreted) solution of $\mathcal{E}_{\mathcal{G}}$ by 'unraveling' these equations from the initial non-terminal X_0 of the grammar. This results in a possibly infinite regular term $t_{\mathcal{G}}$, which is precisely the least fixed-point solution for X_0 in $\mathbb{T}_{\Theta}$. By cocontinuity of the evaluation mapping $[\![t_{\mathcal{G}}]\!]$ is isomorphic to the least fixed-point solution of $\mathcal{E}_{\mathcal{G}}$ in $\mathbb{G}$, that is to the graph generated by $\mathcal{G}$.

In what follows we focus on different sets of graph operations Θ (namely, HR, VR and some extensions). It has been observed that for suitable choices of operations, most notably avoiding products, the evaluation mapping can be realised as a monadic second-order interpretation or transduction [11, 60]. Consequently every interpretation $[\![t]\!] \leq^{\mathcal{I}}_{\mathrm{MSO}} t$ naturally translates to an internal presentation of $[\![t]\!]$ using tree automata. Moreover, for a regular term t the MSO-theory of $[\![t]\!]$ is decidable by Rabin's Theorem.

Finally we mention that all this smoothly extends to solutions of infinite sets of equations [33]. Although unraveling might not result in a regular solution term, as long as it has a decidable MSO-theory so does the solution graph.

Equational graphs and hyperedge-replacement grammars

Hyperedge-replacement (HR) grammars are a very natural generalisation of context-free grammars from formal language theory. Every HR-grammar defines a 'language' of finite graphs just as context-free grammars define languages of finite words. The class of graph languages defined by HR-grammars possesses many structural properties akin to those well-known for context-free languages. The interested reader is referred to the monograph [80].

An HR-grammar is given as a finite collection of rules that allow the replacement of any hyperedge of a hypergraph bearing a non-terminal label by the right hand side of a matching rule, which is a given finite hypergraph with a number of distinguished vertices equal to the arity of the hyperedge to be replaced. A copy of the right-hand side of a matching rule is then glued to the original hypergraph precisely at these distinguished vertices and corresponding end vertices of the hyperedge being replaced. Derivation begins with a distinguished non-terminal.

As outlined at the start of section 1.2.2, each *deterministic* HR-grammar

determines a unique countable graph constructed from the initial graph by complete rewriting in the course of which every non-terminal hyperedge is eventually replaced by the right-hand side of the unique matching rule. A countable graph is HR-*equational*, or simply *equational*, if it is generated by a deterministic HR grammar [55]. The class of equational graphs will be denoted by HR. Equational graphs constitute a proper extension of the class of context-free graphs [41].

Proposition 1.2.5 *A connected graph is context-free if, and only if, it is equational and of finite degree.*

Example 1.2.6 To generate the context-free graph of Example 1.2.1 with a deterministic HR grammar we take as our initial graph the 1-neighbourhood of the root node (labelled with 1 above) and attach to it non-terminal hyperedges labelled with X and with Y, respectively, whose vertices enumerate the boundaries of either ends. Similarly, the 1-neighbourhood of the boundary of each end, that is the vertices of the corresponding non-terminal hyperedge, constitutes the right-hand side of the matching rule. Again, non-terminal hyperedges are attached to mark the new boundary. The initial graph and the rule for the non-terminal X obtained this way are pictured below.
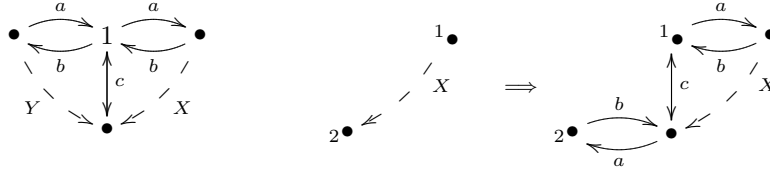
Notice how the linearity of the generated graph is reflected in the linearity of the replacement rules each having only a single non-terminal hyperedge on the right. In the next example a non-linear rule is used to generate a tree, which is not context-free.

Example 1.2.7 The complete bipartite graph $K_{1,\omega}$ and the full ω-branching tree $\mathfrak{T}_\omega$ (in the signature of graphs) are not context-free, but can be generated by the following rules from the initial graph $\bullet \overset{X}{\dashrightarrow} \bullet$.

The HR-algebra of finite and countable graphs corresponding to hyperedge-replacement grammars is a many-sorted algebra defined as follows.

For each n there is a separate sort $\mathbb{G}_n$ of graphs with n *sources*. These are distinguished vertices, though not necessarily distinct, named $v_1, \ldots, v_n$. There are constants of each sort $\mathbb{G}_n$: these are hypergraphs having at most one hyperedge, exactly n vertices, each vertex a distinct source. The HR-algebra is built on the following operations: *disjoint union* $\oplus$, *renaming of sources* $\mathtt{rename}_{c \mapsto c'}$, and *fusion of sources* $\mathtt{fuse}_{\approx}$ according to an equivalence $\approx$ on source names. By convention $\oplus$ is understood to automatically shift the source names of its second argument by the maximum of the source names used in the first to avoid naming conflict. Also $\mathtt{fuse}$ assigns the least source name of a class to each fused node while dropping the others.

It is intuitively clear how a hyperedge-replacement step can be expressed using disjoint union with the right-hand side of the rule followed by a fusion and renaming of sources. Formally, one transforms an HR-grammar $\mathcal{G}$ into a system of finitely many equations $X_i = t_i(\overline{X})$ where variables play the role of non-terminals of the grammar and the terms t_i are chosen such that, when variables are interpreted as individual hyperedges, $[\![t_i(\overline{X})]\!]$ is the right hand side of the matching rule for a hyperedge labelled X_i.

Example 1.2.8   The equation corresponding to the single rule of the HR grammar of Example 1.2.7 generating $\mathfrak{T}_\omega$ is

$$X = \mathtt{rename}_{0 \mapsto 0, 1 \mapsto 1}\big(\, \mathtt{fuse}_{\{0,2\},\{1,4\}}(\; \overset{0}{\bullet} \to \overset{1}{\bullet} \,\oplus X \oplus X \;)\,\big)\,.$$

Note that the source names of the first and second occurrences of X are shifted by 2 and by 4, respectively, while forming their disjoint union. Thus, after fusion we obtain precisely the right hand side of the HR-rule generating $\mathfrak{T}_\omega$, however, with additional source names. The renaming operation in this term has the effect of forgetting the source names 2 and above. So the least solution of this equation is indeed $\mathfrak{T}_\omega$ with its root labelled 0 and one of its children with 1.

The generating power of HR-grammars is limited by the fact that edges can only be 'created' via fusion of sources (after having taken the disjoint union of two graphs). Because there are only a fixed number of source names available in a finite HR-equational system there is a bound on the size of complete bipartite subgraphs $K_{n,n}$ that can be created [12], cf. Theorem 1.2.12. The infinite bipartite graph $K_{\omega,\omega}$ is thus an example of a prefix-recognisable graph which is not HR-equational.

It is a key observation that in case of HR-terms the evaluation mapping

$t \mapsto [\![t]\!]$ is expressible as an MSO-interpretation. In fact, since edges cannot be created by any of the HR operations, the vertex-edge-adjacency graph of $[\![t]\!]$ is MSO-interpretable in the tree representation of t, whether t is finite or infinite.

Theorem 1.2.9 *For a countable graph G the following are equivalent.*

(1) G is generated by a deterministic HR grammar;

(2) G is HR-equational, i.e. the evaluation of a regular HR-term, i.e. the least solution of a finite system of HR-equations;

(3) The two-sorted incidence graph $\hat{G}$ of G is monadic second-order interpretable in the full binary tree, i.e. $\hat{G} \leq_{\mathrm{MSO}} \mathfrak{T}_2$.

For a detailed presentation of these and other algebraic frameworks and their connections to the generative approach based on graph grammars we advise consulting [55, 12, 21]. In [54] Courcelle considered an extension of monadic second-order logic, denoted CMSO_2, in which one can quantify over sets of edges as well as over sets of vertices and, additionally, make use of modulo counting quantifiers. Notice that the last item of the previous theorem implies that the CMSO_2-theory of equational graphs is interpretable in S2S and is thus decidable. Further, Courcelle proved that CMSO_2 is able to axiomatise each and every equational graph up to isomorphism.

Theorem 1.2.10 *Each HR-equational graph is axiomatisable in CMSO_2. Consequently the isomorphism problem of equational graphs is decidable.*

Sénizergues considered HR-equational graphs of finite out-degree and proved that they are, up to isomorphism, identical with the ε-closures of configuration graphs of normalised[7] pushdown automata restricted to the set of reachable configurations. Further, he proved that bisimulation equivalence of HR-equational graphs of finite out-degree is decidable [128]. This last result is an improvement on the decidability of bisimulation equivalence for deterministic context-free processes, which is a consequence of the celebrated result of Sénizergues establishing decidability of the DPDA language equivalence problem.

---

[7] Here a PDA is said to be normalised, if in addition to being in a familiar normal-form its ε-transitions may not push anything on the stack. Hence the finiteness bound on the out-degree of configurations. For precise definitions see [128].

Vertex-replacement grammars

Vertex replacement systems are a finite collection of graph rewriting rules that allow one to substitute given finite graphs in place of single vertices while keeping all the connections. This form of graph rewriting emerged as the most robust and manageable from among a host of different notions within a very general framework [55, 69, 59, 58]. The corresponding VR-algebra of graphs is built on the following operations: constant graphs of a single c-coloured vertex $\overset{a}{\bullet}$, *disjoint union* $\oplus$, *recolouring of vertices* $\texttt{recol}_{c \mapsto c'}$ and *introduction of a-coloured edges* $\texttt{edge}_{c \overset{a}{\to} d}$ from every c-coloured vertex to every d-coloured vertex.

The evaluation of VR-terms, whether finite or infinite, is realisable as a monadic second-order interpretation. More precisely, as VR-equational graphs are interpretations of regular terms obtained by unfolding a finite system of VR equations, they can be MSO-interpreted in a regular tree, hence also in the full binary tree $\mathfrak{T}_2$, and thus are prefix-recognisable. These and other characterisations, together with our previous discussion of prefix-recognisable structures are summarised in the next theorem.

Theorem 1.2.11   *For a countable graph G the following are equivalent.*

(1) G is isomorphic to a prefix-recognisable structure;

(2) G is generated by a deterministic VR grammar;

(3) G is VR-equational, i.e. the evaluation of a regular VR-term, i.e. the least solution of a finite system of equations of the form $X_i = t_i(\overline{X})$ with finite VR-terms $t_i(\overline{X})$;

(4) $G \leq_{\mathrm{MSO}} \mathfrak{T}_2$;

(5) $G = h^{-1}(\mathfrak{T}_2)|_C$, i.e. the vertices of G are obtained by restricting the nodes of $\mathfrak{T}_2$ to a regular set C, and its edges are obtained by taking the inverse of a rational substitution h to $\mathfrak{T}_2$;

(6) G is isomorphic to the ϵ-closure of the configuration graph of a pushdown automaton.

Further, the HR-equational graphs can be characterised as the class of VR-equational graphs of finite tree width [11].

Theorem 1.2.12   VR-*equational graphs of finite tree width are* HR-*equational.*

Example 1.2.13   The complete bipartite graph $K_{\omega,\omega}$ is a prominent example of a VR-equational graph that is not HR-equational. A VR grammar and the corresponding system of VR equations generating $K_{\omega,\omega}$ are

given below.

$$
\begin{aligned}
\overset{X}{\bullet} \;&\Rightarrow\; \overset{A}{\bullet}\longleftrightarrow\overset{A}{\bullet} & X \;&=\; \texttt{edge}_{a\leftrightarrow b}(\; A \oplus \texttt{recol}_{a\mapsto b}(A)\;) \\
\overset{A}{\bullet} \;&\Rightarrow\; \bullet \;\; \overset{A}{\bullet} & A \;&=\; \overset{a}{\bullet} \oplus A
\end{aligned}
$$

The expressive power of this formalism (for describing families of finite graphs) is not increased by extending the VR operations by graph transformations that are definable using quantifier-free formulas (of which $\texttt{recol}_{c\mapsto c'}$ and $\texttt{edge}_{c\overset{a}{\to}d}$ are particular examples), nor by the *fusion* operations $\texttt{fuse}_c$ identifying all nodes bearing a certain colour c [60]. Care has to be taken when defining countable graphs as evaluations of infinite terms, for it is unclear how to deal with infinite terms built with non-monotonic operations. Nonetheless, infinite terms built with operations definable by *positive* quantifier-free formulas can be evaluated unambiguously [11].

In this setting Theorem 1.2.11 can be generalised to infinite systems of equations (whose unfoldings are typically non-regular terms) using infinite deterministic automata [33], leading us to the following families of transition graphs.

1.2.3 Higher-order data structures

Tree-constructible graphs and Caucal's pushdown hierarchy
Courcelle introduced MSO-*compatible transductions* in the investigation of structures with decidable monadic theories. Let $\mathcal{C}$ and $\mathcal{C}'$ be classes of structures on signatures σ and σ', respectively. Following [57] we say that a functional transduction $T : \mathcal{C} \to \mathcal{C}'$ is MSO-compatible if there is an algorithm mapping each monadic formula φ of signature σ' to a monadic formula φ^T in the signature σ such that

$$
\mathfrak{A} \models \varphi^T \quad \Longleftrightarrow \quad T(\mathfrak{A}) \models \varphi \; .
$$

MSO-interpretations are the most natural examples of MSO-compatible transductions. Slightly more generally, the MSO-*definable transductions* of Courcelle are MSO-compatible. Recall that these are given by a k-copying operation (for some k) followed by an MSO-interpretation and in particular the resulting structure may have k times the cardinality of the original one.

The more difficult result that the *unfolding* operation, mapping graphs $(\mathfrak{G}, v)$ to trees $\mathfrak{T}_{(\mathfrak{G}, v)}$, is also MSO-compatible appeared in [61] (see also [57] for an exposition and a treatment of the simpler case of deterministic

graphs). We note that this result also follows from Muchnik's Theorem [126, 138, 17] and that it generalises Rabin's theorem.

A rich class of graphs, each with decidable monadic theory, can now be constructed. Caucal [43] proposed the hierarchies of graphs and trees obtained by alternately applying unfoldings and MSO-interpretations starting with finite graphs:

Definition 1.2.14

$$
\begin{aligned}
\mathsf{Graphs}_0 &= \{\text{finite edge- and vertex-labelled graphs}\} \\
\mathsf{Trees}_{n+1} &= \{\mathfrak{T}_{\mathfrak{G},v} \mid (\mathfrak{G}, v) \in \mathsf{Graphs}_n\} \\
\mathsf{Graphs}_{n+1} &= \{\mathcal{I}(\mathfrak{T}) \mid \mathfrak{T} \in \mathsf{Trees}_{n+1}, \mathcal{I} \text{ is an MSO interpretation}\}
\end{aligned}
$$

By the results above, we have

Theorem 1.2.15 *For every $n \in \mathbb{N}$ every graph G from Graphs_n has a decidable MSO-theory.*

Fratani [72, 73] provided an alternative proof of the above theorem, among a host of other results on higher-order pushdown graphs, using a different kind of MSO-compatible operation. Indeed, she established that if a homomorphism of words maps the branches of a tree T to those of T' surjectively while also preserving the node-labeling then definability and decidability results for MSO over T' can be transferred to T.

The Caucal hierarchy is very robust. Various weakenings and strengthenings of the definition yield exactly the same classes [37]. In fact, in place of MSO-interpretations, Caucal originally used inverse rational mappings in the style of item (5) of Theorem 1.2.11. Recently Colcombet [51] proved that every graph of Graphs_{n+1} can in fact be obtained via a first-order interpretation in some tree belonging to Trees_{n+1}. The next theorem provides internal presentations of graphs of each level as a generalisation of Theorem 1.2.11 item (6) thereby justifying the name pushdown hierarchy.

Theorem 1.2.16 ([37]) *For every n a graph G is in Graphs_n if, and only if, it is isomorphic to the ϵ-closure of the configuration graph of a higher-order pushdown automaton at level n.*

The strictness of the hierarchy was also shown in [37]. The level-zero graphs are the finite graphs, trees at level one are the regular trees, and as we have seen in Theorem 1.2.11 the level-one graphs are the prefix-recognisable ones. The deterministic level-two trees are known as

algebraic trees. From the second level onwards we have no clear structural understanding of the kind of graphs that inhabit the individual levels. We recommend [134] for an exposition.

Term-trees defined by recursion schemes

Caucal also gave a kind of algebraic characterisation of term-trees at level n as fixed points of *safe* higher-order recursion schemes.

Theorem 1.2.17 ([43]) *For every n, the class of term-trees* Trees$_n$ *coincides with that of term-trees generated by safe higher-order recursion schemes of level at most n.*

The notion of higher-order schemes is a classical one [62, 56]. Safety is a technical restriction (implicit in [62]) ensuring that no renaming of variables (α-conversion) is needed during the generative substitutive reduction (β-reduction) process constructing the solution-term [1, 117]. Safe schemes are intimately related to the pushdown hierarchy. This connection is well explained in [1] showing that while on the one hand order-n schemes can define the behaviour and hence (the unfolding of) the configuration graphs of level-n deterministic pushdown automata, on the other hand, deterministic pushdown automata of level n can evaluate safe order-n schemes. Safety is hereto essential.

In order to evaluate arbitrary schemes [81] introduced *higher-order collapsible pushdown automata* (CPDA), a kind of generalisation of panic automata [92], and gave in essence the following characterisation in the spirit of Theorem 1.2.16.

Theorem 1.2.18 *The term-trees defined by order-n recursion schemes are up to isomorphism identical with the unfoldings of ϵ-closures of configuration graphs of level-n collapsible higher-order pushdown automata.*

As shown in [117, 81], it is not necessary to assume safety for establishing decidability of the MSO-theories of term-trees that are solutions of higher-order schemes.

Theorem 1.2.19 *The* MSO-*theory of a term-tree defined by an arbitrary higher-order recursion scheme is decidable.*

Consequently, configuration graphs of higher-order collapsible pushdown automata can be model-checked against modal μ-calculus formulas. However, there is a second-order CPDA whose configuration graph interprets the infinite grid and whose MSO-theory is thus undecidable

[81]. This shows that higher-order CPDA configuration graphs constitute a proper extension of Caucal's pushdown hierarchy.

1.2.4 Introducing products

There is a connection between the internal presentations of graphs seen so far and the graph operations used in the corresponding equational framework. Pushdown stacks are naturally represented as strings. The set of strings over some alphabet can in turn be modelled as an algebra of terms built with unary functions, one for each letter of the alphabet. Strings thus correspond to terms and letters to unary functions. In functional programming terminology the abstract data type of, say, binary strings has the recursive type definition

$$\mathcal{T} \;=\; \bot \;\oplus\; 0(\mathcal{T}) \;\oplus\; 1(\mathcal{T}) \tag{1.2}$$

Here the letters 0 and 1 are seen as type constructors and the empty string $\bot$ is a constant type constructor. The set of finite strings is the least fixed-point solution of this equation.

Automata operating on terms of type $\mathcal{T}$ can be viewed as functions mapping terms to states. Moreover these functions are defined according to structural recursion. Analogously, recursion schemes (fix-point equations) in an algebra of graph operations transform automata-based internal presentations of a graph into equational specifications. We can use the recursion scheme associated to the type definition (1.2) to define any PR-graph by a VR equation extending the type definition. For instance, the graph of the lexicographic order from Example 1.2.3 satisfies the following equation

$$L = \mathtt{edge}_{0 \to 1, \varepsilon \to 0, \varepsilon \to 1}(\bullet^\varepsilon \oplus \mathtt{recol}_{0,1,\varepsilon \mapsto 0}(L) \oplus \mathtt{recol}_{0,1,\varepsilon \mapsto 1}(L)).$$

We briefly explain how to go from automata presenting a PR-graph to a VR-equation. For a language $V \subset \{0,1\}^*$ recognised by an automaton with transition table $\Delta \subset Q \times \Sigma \times Q$ and final states F the following VR-equation colours each word $w \in \{0,1\}^*$ by those states q such that the automaton starting from q accepts w. (N.B. in accordance with (1.2) the simulation proceeds right-to-left.)

$$X = \bullet^F \oplus \mathtt{recol}_{\{q' \mapsto q : \Delta(q,0,q')\}}(X) \oplus \mathtt{recol}_{\{q' \mapsto q : \Delta(q,1,q')\}}(X)$$

In general, every PR-graph $\bigcup_i U_i \cdot (V_i \times W_i)$ is the recolouring of a graph satisfying a VR-equation of the form

$$X = \vartheta(\vartheta_\varepsilon(\bullet) \oplus \vartheta_0(X) \oplus \vartheta_1(X)) \;. \tag{1.3}$$

Here, the states of the automata recognising V_i or W_i are encoded as vertex colours (just as above) and ϑ_ε colours $\bullet$ by the final states of the V_i's and W_i's. Edge colours are used to represent states of automata for each U_i. For every $v \in V_i$ and $w \in W_i$, and z accepted by the automaton for U_i from state q there is a q-coloured edge (zv, zw). To this end, ϑ_0 and ϑ_1 recolour the vertices and edges, and ϑ adds an edge between all $x \in V_i$ and $y \in W_i$ coloured by the final states of U_i.

In passing we mention that higher-order stacks can also be represented as strings: either as well-bracketed sequences of stack symbols, or as strings of stack operations yielding the particular stack configuration. The former comes at the cost of losing regularity of the domain and has no apparent algebraic counterpart. The latter gives rise to a unary algebra of higher-order stacks that is not, except for level 1 pushdown stacks, freely generated by the stack operations. Thus there is no unique term representing a general stack. The work of Fratani, Carayol and others [72, 73, 33, 32] has shown that both of these deficiencies can be turned into features.

We now turn to graphs internally presented by finite trees. A type definition for $\{0, 1\}$-labelled binary branching trees is

$$\mathcal{T} \;=\; \bot \;\oplus\; 0(\mathcal{T} \otimes \mathcal{T}) \;\oplus\; 1(\mathcal{T} \otimes \mathcal{T}) \tag{1.4}$$

where $\otimes$ denotes direct product. Later we will compare this with another type definition (1.6). Colcombet observed that this schema can be used to define graphs with internal presentations involving tree automata operating on finite trees. He proposed extensions of the VR-algebraic framework by the *asynchronous product* $\otimes_A$ [48] and by the *synchronous product* $\otimes_S$ [50, 49] which we shall denote here by VRA and VRS, respectively.

Definition 1.2.20 (Synchronous and asynchronous product) The products are defined for vertex and edge-coloured graphs $\mathcal{G}$ and $\mathcal{H}$ as follows. In the synchronous product there is a d-coloured edge from (g, h) to (g', h') if, and only if, both (g, g') and (h, h') are connected by a d-edge in $\mathcal{G}$ and $\mathcal{H}$, respectively. The edge relation E_d of the asynchronous product $\mathcal{G} \otimes_A \mathcal{H}$ is defined as the union of $\{((g, h), (g', h)) \mid E_d^{\mathcal{G}}(g, g'), h \in H\}$ and $\{((g, h), (g, h')) \mid E_d^{\mathcal{H}}(h, h'), g \in G\}$. The definition of vertex colours requires a little care. In both cases a vertex (g, h) of the product has colour $\delta(c, c')$ whenever g has colour c and h has colour c'. Here the function $\delta : C^2 \to C$ is a parameter of the product operation. However,

it is really only relevant that δ acts as a pairing function on some sufficiently large subsets of the colours. For instance, Colcombet identifies C with $\{0, 1, \ldots, N-1\}$ and defines δ as addition modulo N [48].

As before, VRA-equational and VRS-equational graphs are defined as least fixed-point solutions of a finite system of equations in the respective algebra. Both product operations are cocontinuous with respect to graph embeddings. Therefore the evaluation mapping of both VRA and VRS terms uniquely extends from finite terms to infinite terms. Hence, just as for HR- and VR-equational graphs, the solution of a system of VRA or VRS equations is the evaluation of the regular term obtained by unraveling the system of equations.

Example 1.2.21   The infinite two-dimensional grid $(\mathbb{N} \times \mathbb{N}, \texttt{Up}, \texttt{Right})$ is easily constructed as the asynchronous product of the VR-equational, even context-free, graphs $(\mathbb{N}, \texttt{Up})$ and $(\mathbb{N}, \texttt{Right})$:

$$
\begin{aligned}
G &= \otimes_A(N_u, N_r) \\
N_u &= \texttt{edge}_{a \overset{\texttt{Up}}{\to} b}\left(\overset{a}{\bullet} \oplus \texttt{recol}_{a \mapsto b, b \mapsto c}(N_u)\right) \\
N_r &= \texttt{edge}_{a \overset{\texttt{Right}}{\to} b}\left(\overset{a}{\bullet} \oplus \texttt{recol}_{a \mapsto b, b \mapsto c}(N_r)\right)
\end{aligned}
$$

The unfolding of this system of equations is, schematically, an infinite term consisting of two periodic branches joined at the root. Elements of the grid G, by definition of asynchronous product, are represented as pairs of nodes of this term-tree with one node on either branch, corresponding to the respective co-ordinates. The example of the grid, whose MSO theory is undecidable, shows that the evaluation mapping of VRA terms (also of VRS terms) can not be realised by an MSO-interpretation.

For any VRA or VRS-term t, vertices of $[\![t]\!]$ can be identified with maximal *subsets* of nodes of t belonging to sub-terms joined by a product operator. It is thus easily expressible in MSO whether a set X of nodes (finite or infinite[8]) is actually well-formed in this sense, i.e. whether it represents an element of $[\![t]\!]$.

VR with asynchronous product and ground term rewriting

Ground term rewrite systems (GTRSs) are a natural generalisation of prefix-rewriting to trees. They are term rewrite systems given by rewrit-

---

[8]   In least fixed-point semantics only finite sets are considered, whereas in greatest fixed-point semantics both finite and infinite sets can represent elements of the solution, provided that there is an infinite nesting of product operators in t.

ing rules in which no variables occur. Tree automata are a special case of GTRSs (see [52]).

Example 1.2.22 The rewrite rule $a \to f(a)$ confined to terms of the form $d(f^n(a), f^m(a))$ is a GTRS whose configuration graph is isomorphic to the infinite square grid.

We have noted that prefix-recognisable graphs are identical to ε-closures of pushdown graphs. This correspondence is achieved by generalising the simple prefix-rewriting rules of pushdown systems of the form $v \to w$ where v and w are strings to replacement rules $V \to W$ for given regular languages V, W. The latter rule allows one to rewrite any prefix $v \in V$ of a given string by any word from W. Regular Ground Term Rewrite Systems (RGTRS) generalise GTRS in the exact same manner: simple ground rewrite rules $s \to t$ with ground terms s, t are replaced by 'rule schemes' $S \to T$ with regular sets of terms on both left and right-hand side.

Löding [99, 100] and Colcombet [48] studied transition graphs of GTRSs and RGTRSs from a model-checking point of view. In Löding's work vertices of the transition graph are those terms reachable from an initial term, whereas Colcombet considers all terms of a given type as vertices.

The VR-equations defining PR graphs (1.3) easily generalise to VRA-equations defining graphs of RGTRSs using the recursion scheme (1.4):

$$X = \vartheta(\vartheta_\varepsilon(\bullet) \oplus \vartheta_0(X \otimes_A X) \oplus \vartheta_1(X \otimes_A X)) \tag{1.5}$$

For each rule $S_i \to T_i$ of the RGTRS we simulate (frontier to root) tree automata recognising S_i and T_i. Vertices of X represent terms, so we call these vertex-terms. A vertex-term is coloured by those states q occurring at the root of the term after being processed by the automata. The simulation is initialised as follows: ϑ_ε labels $\bullet$ by initial states, and ϑ adds edges between all vertex-terms coloured by accepting states of automata for S_i and T_i. Updates occur in ϑ_js according to the transition rules, similarly to (1.3). To this end assume that two vertex-terms v', v'' are coloured by states q' and q'' respectively. After taking the product the paired vertex-term $j(v', v'')$ is initialised with colour (q', q'') (cf. Def. 1.2.20). This pair is then recoloured to q by ϑ_j whenever (q, j, q', q'') is a transition.

Notice how naturally the asynchronous product captures closure of RGTR rewriting under contexts: if there was an edge between v and v' then there is an edge between $j(v, v'')$ and $j(v', v'')$, and, symmetrically,

between $j(v'', v)$ and $j(v'', v')$. One obtains along these lines the following generalisations of Theorem 1.2.11 (cf. examples 1.2.22 and 1.2.21).

Theorem 1.2.23 (Colcombet [48])

(i) *A countable graph is* VRA-*equational if, and only if, it is (after removal of certain colours) isomorphic to an RGTRS graph*[9].

(ii) *Each* VRA-*equational graph is finite-subset interpretable in a regular term-tree, hence also in the full binary tree.*

Theorem 1.2.12 also extends to VRA-equational graphs [48, 100].

Theorem 1.2.24   VRA-*equational graphs of finite tree-width are HR-equational.*

An immediate consequence of Theorem 1.2.23 is that the FO-theory of every VRA-equational structure is decidable via interpretation in S2S. In fact, for any VRA-equational graph $G = (V, \{E_a\}_a)$ the subset interpretation, hence also first-order decidability, extends to G with additional reachability predicates $R_C = \{\, (v, w) \mid w$ can be reached from v using edges of colours from $C \,\}$ for arbitrary subsets C of edge colours [48].

Theorem 1.2.25   VRA-*equational graphs have a decidable first-order theory with reachability.*

This result cannot be improved much further. Examples of [139] show that 'regular reachability', i.e. the problem whether there exists a path in a given VRA-equational graph between two given nodes and such that the labeling of the path belongs to a given regular language over the set of colours, is undecidable. In [100] Löding identified a maximal fragment of CTL that is decidable on every GTRS graph (with vertices restricted to terms reachable from an initial one) that can express, besides reachability, recurring reachability.

VR with synchronous product and tree-automatic structures
We have remarked that in the subset interpretation of VRA terms the subsets are used in a special form. Indeed, in the evaluating interpretation they merely serve the purpose of outlining the shape of a finite term. General finite-subset interpretations are more powerful and are capable of expressing the evaluation of VRS terms. In fact, these two formalism are equally expressive.

---

[9] Here RGTRS graphs are taken in the sense of [48] as being restricted to the set of terms of a given type.

This is best explained by *tree-automatic presentations*. These are internal presentations of VRS-structures which will be formally introduced in the next section. For now it suffices to use the characterisation (Theorem 1.3.18) that tree-automatic graphs are those that are wMSO-interpretable in a regular tree (reflected in the equivalence of (1) and (2) below).

Theorem 1.2.26 (Colcombet [50])
For every countable graph G the following are equivalent

(1) G is isomorphic to a tree-automatic graph.
(2) G is interpretable in a regular tree (wlog. the full binary tree) via a finite-subset interpretation.
(3) G is the restriction of a VRS-equational vertex-labelled graph G' to its set of vertices of a given colour;

We have noted that the evaluation mapping of VRS-terms can be naturally defined as a finite subset interpretation - this justifies (3) $\rightarrow$ (2). Continuing our discussion of translations from automata-based internal presentations into equational specifications using graph products we illustrate the remaining translation (2) $\rightarrow$ (3) from finite-tree automatic to VRS-equational presentations on graphs as we did for PR and RGTRS. That is, we build the terms of the presentation from the bottom up while also simulating the automata constituting the tree-automatic presentation by VRS-operations.

Start with a graph (V, E) that is definable via finite-subset interpretation in the full binary tree. By the fundamental correspondence that wMSO-definable relations in a regular tree are exactly those that are recognised by tree automata operating on finite trees, we see that V may be taken to be a regular set of finite Σ-labelled binary trees, and E is recognised by an automaton $\mathcal{A}$ accepting pairs of such trees.

The tree automaton $\mathcal{A}$ has transition rules (here we read them from left-to-right, i.e in top-down fashion, but that is a matter of choice and the simulation will actually proceed from bottom up) of the form

$$r : (q, \langle a, b \rangle, q_0, q_1) \qquad \text{with } a, b \in \{0, 1, \square\}$$

where the symbol $\square$ is necessary for padding either components of a pair of trees so that they have the same shape. It indicates the fact that no node is defined in the current position, i.e. that the automaton finds itself below a leaf of the respective tree (while still reading the other).

We may assume that the transition rules enforce a proper usage of the padding symbols.

We introduce edge relations E_q and E_r for each state q and each rule r of the automaton. The simulation of transitions of the synchronous automaton on *pairs of labelled trees* necessitates a more sophisticated recursion scheme associated to the following type definition of $\{0, 1\}$-labelled binary branching trees.

$$\mathcal{T} \;=\; \perp \;\oplus\; (\,\{0,1\} \otimes \mathcal{T} \otimes \mathcal{T}\,) \tag{1.6}$$

There is a natural identification of terms of this type and of those of the more natural type definition (1.4). As far as unary predicates are concerned the current type definition does not provide any advantage. However, compared with (1.4) the current type definition has a more powerful associated recursion scheme allowing for defining non-trivial *binary* relations between terms with different root labels. This will allow us to specify tree-automatic graphs via VRS-equations of the following form analogous to (1.6)

$$X = \vartheta \left(\bullet^\perp \oplus (\vartheta_0 \otimes_S \vartheta_1(X) \otimes_S \vartheta_2(X)) \right) \tag{1.7}$$

Here too, as in (1.3) and in (1.5) the ϑ's are VR-expressions facilitating the simulation of the automaton. The expression ϑ_0 specifies the graph with vertex set $\{0, 1\}$ and having an r-labelled edge from a to b for each rule r such that $r = (\cdot, \langle a, b \rangle, \cdot, \cdot)$ and with VR operations (here equivalently expressed as positive quantifier-free definable operations) responsible for updating the edge relations to simulate the transitions of $\mathcal{A}$. This is done in two phases.

- First, in preparation, state-labelled edges are used to 'enable' compatible rule-labelled edges in either copy of the graph: for each rule $r = (\cdot, \langle \cdot, \cdot \rangle, q_1, q_2)$ and $i \in \{1, 2\}$ the expression ϑ_i adds an E_r-edge from x to y for every E_{q_i}-edge from x to y in the graph.
- Then, after the synchronous product of rule-labelled edges has been taken, edges labelled by rules are renamed to their resulting states: ϑ adds for each state q an E_q-edge from x to y for every E_r-edge from x to y such that $r = (q, \langle \cdot, \cdot \rangle, \cdot, \cdot)$. In addition, ϑ deals with the case when either x or y is the singleton tree $\perp$. For this we may assume that all necessary information is coded in vertex labels implemented as reflexive edges and maintained along with the rest of the edge labels as explained here.

Finally, to obtain the graph G' as required in item (3) of Theorem 1.2.26

we also use vertex colours to keep track of the states of the tree automaton recognising V. The generalisation of this construction to arbitrary relational structures is straightforward.

1.3 Automatic Structures

1.3.1 Fundamentals

This section concerns structures with internal presentations consisting of automata operating synchronously on their inputs. The starting point of this investigation is the robust nature of finite automata. In particular, synchronous automata are effectively closed under certain operations that can be viewed in logical terms, i.e. Boolean operations, projection, cylindrification and permutation of arguments. Thus a structure whose domain and atomic operations are computable by such automata has decidable first-order theory (Definition 1.3.2 and Theorem 1.3.4).

Example 1.3.1 (i) The domain and relations of the following structure are regular.

$$\mathcal{S}_\Sigma = (\Sigma^*, \{\texttt{suc}_a\}_{a \in \Sigma}, \prec_{\texttt{prefix}}, \texttt{el})$$

where Σ^* is the set of finite words over alphabet Σ, the binary relation $\texttt{suc}_a$ is the successor relation (x, xa) for $x \in \Sigma^*$, the binary relation $\prec_{\texttt{prefix}}$ is the prefix relation and the binary relation $\texttt{el}$ is the equal-length relation.

(ii) The following structure can be coded (eg. in base k least significant digit first) so that the domain and atomic operations are regular.

$$\mathcal{N}_k = (\mathbb{N}, +, |_k)$$

where $+$ is the usual addition on natural numbers and $x \,|_k\, y$ holds precisely when x is a power of k and x divides y.

Actually the link between synchronous automata and logic goes both ways. It was first expressed in terms of weak monadic second-order logic: a set of tuples $(A_1, \cdots, A_n)$ of finite sets of natural numbers is weak monadic second-order definable in $(\mathbb{N}, \mathrm{S})$ if and only if the corresponding n-ary relation of characteristic strings (a subset of $(\{0, 1\}^*)^n$) is synchronous rational. This was proved by [27] and [68], and is implicit in [135].

A first-order characterisation was provided by [65]: a relation $R \subset (\Sigma^*)^n$ is synchronous rational if and only if R is first-order definable

in $\mathcal{S}_\Sigma$ for $|\Sigma| \geq 2$. Similarly, the Büchi-Bruyère Theorem states that a relation $R \subset \mathbb{N}^n$ (coded in base $k \geq 2$ least significant digit first) is synchronous rational if and only if it is first-order definable in $\mathcal{N}_k$ (proofs of which can be found in [104] and [137]).

These results were generalised to full MSO on the line $(\mathbb{N}, S)$ and weak MSO and full MSO on the tree $(\{0, 1\}^*, \mathtt{suc}_0, \mathtt{suc}_1)$ and form the basis of the logical characterisation of automatic structures (Section 1.3.4). However, we start with the more common internal definition.

Recall that the four basic types of automata operate on finite or infinite words or trees. So, let $\square$ be one of $\mathtt{word,}$ $\omega\mathtt{-word,}$ $\mathtt{tree,}$ $\omega\mathtt{-tree}$.

We consider a structure $\mathfrak{B} = (B, \{R_i\})$ comprising relations R_i over the domain $\mathrm{dom}(\mathfrak{B}) = B$. Thus constants and operations are implicitly replaced by their graphs.

Definition 1.3.2 (Automatic presentation)   A $\square$-automatic presentation of $\mathfrak{B}$ consists of a tuple $\mathfrak{d} = (\mathcal{A}, \mathcal{A}_\approx, \{\mathcal{A}_i\})$ of finite synchronous $\square$-automata and a *naming function* $f : \mathcal{L}(\mathcal{A}) \to B$ such that

- Each $\mathcal{L}(\mathcal{A}_i)$ is a relation on the set $\mathcal{L}(\mathcal{A})$.
- $\mathcal{L}(\mathcal{A}_\approx)$ is a congruence relation on the structure $(\mathcal{L}(\mathcal{A}), \{\mathcal{L}(\mathcal{A}_i)\}_i)$.
- The quotient structure is isomorphic to $\mathfrak{B}$ via f.

Moreover, the quotient structure is called an *automatic copy* of $\mathfrak{B}$. We say that the presentation is *injective* whenever f is, in which case $\mathcal{A}_\approx$ can be omitted.

Definition 1.3.3 (Automatic structure[10])   A structure $\mathfrak{B}$ is $\square$-*automatic* if it has an $\square$-automatic presentation. If $\mathfrak{B}$ is $\square$-automatic for some $\square$ then $\mathfrak{B}$ is simply called *automatic*. The classes of automatic structures are respectively denoted by $\mathtt{S\text{-}AutStr}$, $\omega\mathtt{S\text{-}AutStr}$, $\mathtt{T\text{-}AutStr}$ and $\omega\mathtt{T\text{-}AutStr}$.

The following theorem motivates the study of automatic structures and so may be called the *Fundamental Theorem* of automatic structures/presentations.

Theorem 1.3.4 (Definability)   *There is an algorithm that given a $\square$-automatic presentation $(\mathfrak{d}, f)$ of a structure $\mathfrak{A}$ and a FO-formula $\varphi(\overline{x})$ in the signature of $\mathfrak{A}$ defining a k-ary relation R over $\mathfrak{A}$, effectively constructs a synchronous $\square$-automaton recognising $f^{-1}(R)$.*

Immediate corollaries are

---

[10] Some authors write *automatically presentable*.

(i) *Decidability*: The FO-theory of every automatic structure is decidable.

(ii) *Interpretations*: The class of □-automatic structures is closed under FO-interpretations.

We point out that the Fundamental Theorem implies that every relation first-order definable from □-regular relations is itself □-regular.

Remark 1.3.5 One may allow finitely many parameters $\varphi(\overline{a}, \overline{x})$ under the following conditions. For finite-word and finite-tree presentations any parameters can be used. However, for ω-tree (and ω-word) presentations a parameter a can be used if $f^{-1}(a)$ contains a regular ω-tree (ultimately periodic ω-word).

Consequently □-automatic structures (on a given signature) are closed with respect to operations such as disjoint union, ordered sum and direct product – each a special case of generalised products treated in [20, 23]. However AutStr and ωS-AutStr are not closed under weak direct-power. For instance, $(\mathbb{N}, +)$ is in S-AutStr but its weak direct-power is isomorphic to $(\mathbb{N}, \times)$, which is not in S-AutStr (see [20]). On the other hand, it is straightforward to see that T-AutStr and ωT-AutStr are closed under weak direct-power.

1.3.2 Examples

Obviously every finite structure is automatic. Here are a some examples of structures with automatic presentations.

Example 1.3.6 (Ordinals) (i) $(\omega, <) \in$ S-AutStr: The simplest automatic copy is the unary one: $(0^*, \{(0^k, 0^l) \mid k < l\})$.

(ii) Every ordinal below ω^ω is in S-AutStr: An automatic copy of ω^k is $((0^*1)^k, <_{\text{lex}})$ where $<_{\text{lex}}$ denotes the lexicographic order[11] which is clearly regular. In this presentation the naming function is

$$0^{n_{k-1}}1 \dots 0^{n_0}1 \mapsto n_{k-1}\omega^{k-1} + \dots + n_1\omega^1 + n_0 \quad .$$

(iii) Every ordinal below ω^{ω^ω} is in T-AutStr: recall that the ordinal ω^α has a representation as the set of functions $f : \alpha \to \omega$ with f equal to 0 in all but finitely many places. These functions are ordered as follows: $f < g$ if the largest β with $f(\beta) \neq g(\beta)$ has that $f(\beta) < g(\beta)$. Then for fixed k, a function $f : \omega^k \to \omega$ is coded by the tree T_f with

---

[11] Given an ordering on the symbols of the alphabet a word u is lexicographically smaller than w if either u is a proper prefix of w or if in the first position where u and w differ there is a smaller symbol in u than in w.

domain a finite subset of $0^*1^*2^* \cdots k^*$ so that for every β, expressed in Cantor-normal-form as $\omega^{k-1}c_0 + \omega^{k-2}c_1 \cdots + \omega^0 c_{k-1}$, $0 \leq c_i < \omega$, we have $T_f(0^{c_0}1^{c_1} \cdots (k-1)^{c_{k-1}}k^{f(\beta)}) = 1$.

Example 1.3.7 (Orderings) (i) $(\mathbb{Q}, <) \in$ S-AutStr: The countable linear order $(\{0,1\}^*1, <_{\mathrm{lex}})$ is dense without endpoints.

(ii) $(\mathbb{R}, <) \in \omega$S-AutStr.

Example 1.3.8 (Groups) (i) Every finitely-generated group with an Abelian group of finite index is in S-AutStr. And these are the only finitely generated word-automatic groups [116].

(ii) The direct sum of countably many copies of $\mathbb{Z}/m\mathbb{Z}$ is in S-AutStr.

(iii) The subgroup $\mathbb{Z}[1/k]$ of rationals of the form $\{zk^{-i} \mid z \in \mathbb{Z}, i \in \mathbb{N}\}$ for fixed $k \in \mathbb{N}$ is in S-AutStr.

(iv) The Prüfer p-group $\mathbb{Z}(p^\infty) = \mathbb{Z}[1/p]/\mathbb{Z}$ (prime p) is in S-AutStr [114].

(v) Real addition $(\mathbb{R}, +)$ is in ωS-AutStr.

However, the additive group of the rationals $(\mathbb{Q}, +)$ is not automatic [136]. In fact, Tsankov shows that no torsion free Abelian group that is p-divisible for infinitely many primes p is automatic.

Example 1.3.9 (Arithmetics) (i) $(\mathbb{N}, +)$ is in S-AutStr: For every natural $k > 1$, the base k least-significant-digit-first presentation of naturals (with or without leading zeros) constitutes a naming function of an automatic presentation. A finite automaton can perform the schoolbook addition method while keeping track of the carry in its state. Such a presentation is injective when leading zeros are suppressed.

(ii) $(\mathbb{N}, \cdot)$ is in T-AutStr: The presentation is based on the unique factorisation of every natural number n into prime powers $2^{n_2}3^{n_3} \cdots p^{n_p}$. Each n_k is written, say in binary notation, on a single branch of a tree with domain 0^*1^*. Multiplication is reduced to the addition of corresponding exponents. This construction can naturally be generalised to give tree-automatic presentations of weak direct powers of word-automatic structures [20, 25].

Example 1.3.10 (Equivalence relations)   The following have finite-word automatic presentations.

(i) There is one class of size n for every $n \in \mathbb{N}$.

(ii) There are $d(n)$ classes of size $n \in \mathbb{N}$ where $d(n)$ is the number of divisors of n. (This is the direct product of the previous equivalence relation with itself).

Example 1.3.11 (Free algebras) (i) The free algebra with n unary operations and at most ω many constants is in S-AutStr.

(ii) The free monoid generated by a single constant is in S-AutStr. However, no non-unary free or even free-associative algebra on two or more constants is in S-AutStr.

(iii) The free algebra generated by countably many constants and any finite number of operations is in T-AutStr.[12] For instance suppose there is one binary operation F. The domain of the presentation consists of all $\{F, c, \perp\}$-labelled binary trees. The operation (representing F) takes trees S and T as input and returns the tree with domain the prefix-closure of $(\mathrm{dom}(S) \cup \mathrm{dom}(T))\{0, 1\}$ and taking the following values: the root position is labelled F; position $\alpha 0$ is labelled by the label of S at position α; position $\alpha 1$ by the label of T at position α (if either of these latter positions does not exist, the label is $\perp$). It is not known whether finitely generated (non-unary) term algebras are in T-AutStr.

Example 1.3.12 (Boolean Algebras)   The signature we work in consists of the symbols for boolean operations $\cap, \cup, \cdot^c$ and constants $\perp, \top$.

(i) Every finite power of the algebra of finite and co-finite subsets of $\mathbb{N}$ is in S-AutStr.

(ii) The countable atomless Boolean algebra is in T-AutStr: It is isomorphic to the algebra of sets consisting of the clopen sets in Cantor space. Each clopen set has a natural representation as a finite tree.

(iii) The algebra of all subsets of $\mathbb{N}$ is in ωS-AutStr.

(iv) The algebra of all subsets of $\mathbb{N}$ factored by the congruence of having finite symmetric difference is in ωS-AutStr. It is unknown whether this structure can be injectively presented in ωS-AutStr.

(v) The interval algebra of the real interval $[0, 1)$ is in ωT-AutStr.

(vi) The algebra of all subsets of $\{0, 1\}^*$ with a distinguished set $\mathcal{F}$ consisting of those $X \subset \{0, 1\}^*$ such that for every path $\pi \in \{0, 1\}^\omega$ only finitely many prefixes of π are in X.

Example 1.3.13 (Graphs) (i) The infinite upright grid is in S-AutStr: Here the structure is $(\mathbb{N} \times \mathbb{N}, \mathtt{Up}, \mathtt{Right})$ with the functions $\mathtt{Right} : (n, m) \mapsto (n+1, m)$ and $\mathtt{Up} : (n, m) \mapsto (n, m+1)$. It can be automatically presented on the domain $a^* b^*$ with relations

$$R = \begin{pmatrix} a \\ a \end{pmatrix}^* \begin{pmatrix} b \\ a \end{pmatrix} \begin{pmatrix} b \\ b \end{pmatrix}^* \begin{pmatrix} \square \\ b \end{pmatrix}$$

---

[12] Communicated by Damian Niwinski.

and U defined by a similar regular expression.

(ii) The transition graphs of pushdown automata are in `S-AutStr`:[13] Given a pushdown automaton $\mathcal{A}$ with states Q, stack alphabet Γ, input alphabet Σ and transition relation Δ we can construct an automatic presentation of the transition graph of its configurations as follows. We take $Q\Gamma^*$ to be the domain of the presentation in which $q\gamma$ represents the configuration of state q and stack $\gamma \in \Gamma^*$. For each $a \in \Sigma$ there is an a-transition from $q\gamma$ to $q'\gamma'$ if, and only if, $\gamma = z\alpha$, $\gamma' = w\alpha$ and $(q, z, q', w) \in \Delta$ for some $z \in \Gamma$ and $w \in \Gamma^*$. Since Δ is finite, this relation is obviously regular for each a. Notice that in these presentations the transition relations are not only regular but in fact defined by prefix-rewriting rules (cf. Section 1.2.1 on context-free graphs).

(iii) The transition graphs of Turing machines are in `S-AutStr` [87]. We can give an automatic presentation of each TM $\mathcal{M}$ similar to those of pushdown automata. Configurations are encoded as strings $\alpha q\beta \in \Gamma^*Q\Gamma^*$ where α and β are the tape contents to the left, respectively, to the right of the head of $\mathcal{M}$, and q is the current state. Observe that, as opposed to presentations of pushdown graphs, the state is now positioned not at the left of the string but at the location of the head. Consequently, rewriting is not confined to prefixes, but rather occurs around the state symbol: transitions are of the form $\alpha uqw\beta \mapsto \alpha u'q'w'\beta$ for adequate u, w, u', w' and q, q' as determined by the transition function of $\mathcal{M}$. The fact that TM graphs are presentable using *infix rewriting* has the profound consequence that reachability questions in infix-rewriting systems are generally undecidable, as opposed to graphs of *prefix-rewriting* systems, whose monadic second-order theory is decidable (cf. Theorem 1.2.4).

Example 1.3.14 (Automata-theoretic structures)   The following structures turn out to be universal for their respective classes (see Theorem 1.3.17).

(i) Let

$$\mathcal{S}_\Sigma = (\Sigma^*, \{\mathtt{suc}_a\}_{a \in \Sigma}, \prec_{\mathtt{prefix}}, \mathtt{el})$$

and

$$\mathcal{S}_\Sigma^\omega = (\Sigma^{\leq\omega}, \{\mathtt{suc}_a\}_{a \in \Sigma}, \prec_{\mathtt{prefix}}, \mathtt{el})$$

---

[13] For *visibly pushdown automata* the same representation of configurations also allows for the trace equivalence relation to be recognised by a finite automaton. In [10] this presentation was utilised to obtain a decidability result.

be the structures defined on finite, respectively on finite and ω-words, comprising the successor relations $\mathtt{suc}_a = \{(w, wa) \mid w \in \Sigma^*\}$; the prefix relation $u \prec_{\mathtt{prefix}} w$ (where u is finite and w is finite or infinite); and the equal-length relation: $u \mathbin{\mathtt{el}} w$ if, and only if, $|u| = |w|$. Clearly $\mathcal{S}_\Sigma \in \mathtt{S\text{-}AutStr}$ and $\mathcal{S}_\Sigma^\omega \in \omega\mathtt{S\text{-}AutStr}$. Note that if Σ is unary, then $\mathcal{S}_\Sigma$ reduces to $(\mathbb{N}, +1, <, =)$.

(ii) The structure $\mathcal{T}_\Sigma \in \mathtt{T\text{-}AutStr}$ has domain consisting of all finite binary Σ-labelled trees and has operations

$$(\preceq_{\mathtt{ext}}, \equiv_{\mathrm{dom}}, (\mathtt{suc}_a^d)_{d \in \{l,r\}, a \in \Sigma}, (\epsilon_a)_{a \in \Sigma})$$

where $T \preceq_{\mathtt{ext}} S$ if $\mathrm{dom}(T) \subset \mathrm{dom}(S)$ and $S(\alpha) = T(\alpha)$ for $\alpha \in \mathrm{dom}(T)$; $T \equiv_{\mathrm{dom}} S$ if $\mathrm{dom}(T) = \mathrm{dom}(S)$; $\mathtt{suc}_a^d(T) = S$ if S is formed from T by extending its leaves in direction d and labeling each new such node by a; and ϵ_a is the tree with a single node labelled a.

Similarly the structure $\mathcal{T}_\Sigma^\omega \in \omega\mathtt{T\text{-}AutStr}$ has domain consisting of all finite and infinite trees and operations

$$(\preceq_{\mathtt{ext}}, \equiv_{\mathrm{dom}}, (\mathtt{suc}_a^d)_{d \in \{l,r\}, a \in \Sigma}, (\epsilon_a)_{a \in \Sigma}).$$

that are restricted to finite trees, except that $T \preceq_{\mathtt{ext}} S$ is defined as above but allows S to be an infinite tree.

1.3.3 Injectivity

Recall that an automatic presentation is injective if the naming function is injective. The problem of injectivity is this:

Does every □-automatic structure have an injective □-automatic presentation?

An injective presentation has the advantage that it is easier to express certain cardinality-properties of sets of elements (Theorem 1.4.6). We consider the four cases.

Finite words

From a finite-word automatic presentation of $\mathfrak{A}$ one defines an injective presentation of $\mathfrak{A}$ by restricting to a regular set D of unique representatives. These can be chosen using a regular well-ordering of the set of all finite words. For instance, define $D \subset \mathcal{L}(\mathcal{A})$ to be the length-lexicographically least words from each $\mathcal{L}(\mathcal{A}_\approx)$ equivalence class.

Finite trees

Except in the finite word case, there is no regular well ordering of the set of all finite trees [39]. However one can still convert a finite-tree automatic presentation into an injective one [47]. The idea is to associate with each tree t a new tree $\hat{t}$ of the following form: the domain is the intersection of the prefix-closures of the domains of all trees that are $\mathcal{L}(\mathcal{A}_{\approx})$-equivalent to t; a node is labelled σ if t had label σ in that position; a leaf x is additionally labelled by those states q from which the automaton $\mathcal{A}_{\approx}$ accepts the pair consisting of the subtree of t rooted at x and the tree with empty domain.[14] Using transitivity and symmetry of $\mathcal{L}(\mathcal{A}_{\approx})$, if $\hat{t} = \hat{s}$ then t is $\mathcal{L}(\mathcal{A}_{\approx})$-equivalent to s. Moreover each equivalence class is associated with finitely many new trees, and so a representative may be chosen using any fixed regular linear ordering of the set of all finite trees.

ω-words

There is a structure in ωS-AutStr that does not have an injective ω-word automatic presentation [82]. The proof actually shows that the structure has no injective presentation in which the domain and atomic relations are Borel.

However, every *countable* structure in ωS-AutStr does have an injective ω-word automatic presentation [85] (and consequently is also in S-AutStr). This follows from the more general result that every ω-word regular equivalence relation with countable index has a regular set of representatives [85].

ω-trees

It has not yet been settled whether injective presentations suffice, even for the countable structures.

1.3.4 Alternative characterisations

Automatic structures were defined internally. We now present equivalent characterisations: logical (FO and MSO) and equational.

First-order characterisations

In order to capture regularity in the binary representation of $\mathbb{N}$ using first-order logic Büchi suggested the expansion $(\mathbb{N}, +, \{2^n \mid n \in \mathbb{N}\})$ of

---

[14] The construction given in [47] is slightly more general and allows one to effectively factor finite-subset interpretations in any tree.

Presburger arithmetic, which is, however, insufficient (see [26]). Boffa and Bruyére considered expressively complete expansions of $(\mathbb{N}, +)$ by relations of the form $x \mid_k y$ (defined to hold precisely when x is a power of k and x divides y).

Theorem 1.3.15 (Büchi-Bruyére, cf. [26])   *A relation $R \subseteq \mathbb{N}^r$ is regular in the least-significant-digit-first base k presentation of $\mathbb{N}$ if, and only if, R is first-order definable in the structure $\mathcal{N}_k = (\mathbb{N}, +, \mid_k)$.*

Closer to automata, the structures $\mathcal{S}_\Sigma$ on words (see example 1.3.14) allow one to define every regular relation on alphabet Σ.

Theorem 1.3.16 ([65])   *Let Σ be a finite, non-unary alphabet. A relation over Σ^* is regular if, and only if, it is first-order definable in $\mathcal{S}_\Sigma$.*

The proofs of these theorems are by now standard. From left to write one writes a formula $\phi_{\mathcal{A}}(x)$ that expresses the existence of a successful run in automaton $\mathcal{A}$ on input x. For the other direction the atomic operations of the structures are regular forms the base case for structural induction on the formula. Both theorems transfer to automatic structures by replacing definability with interpretability [24, 25].

Theorem 1.3.17 (First-order characterisation of `S-AutStr`)   *The following conditions are equivalent.*

- $\mathfrak{A} \in$ `S-AutStr`*.*
- $\mathfrak{A}$ *is first-order interpretable in $\mathcal{S}_\Sigma$ (for some/all Σ with $|\Sigma| \geq 2$).*
- $\mathfrak{A}$ *is first-order interpretable in $\mathcal{N}_k$ (for some/all $k \geq 2$).*

These structures have been called *universal* or *complete* (with respect to FO-interpretations) for the class of finite-word automatic structures. There are similar universal structures for the other classes of automatic structures. These are the structures $\mathcal{S}_\Sigma^\omega$, $\mathcal{T}_\Sigma$ and $\mathcal{T}_\Sigma^\omega$ from Example 1.3.14 [20, 14].

Finite set interpretations

The four notions of automatic presentation have straightforward reformulations in terms of subset interpretations either in the line $\Delta_1 = (\mathbb{N}, \text{suc})$ or in the tree $\Delta_2 = (\{0, 1\}^*, \text{suc}_0, \text{suc}_1)$.

Theorem 1.3.18 (Automatic presentations as subset interpretations) *There are effective transformations establishing the following equivalences.*

(i) $\mathfrak{A} \in$ S-AutStr *if, and only if,* $\mathfrak{A} \leq_{\mathrm{fset}} \Delta_1$

(ii) $\mathfrak{A} \in \omega$S-AutStr *if, and only if,* $\mathfrak{A} \leq_{\mathrm{set}} \Delta_1$

(iii) $\mathfrak{A} \in$ T-AutStr *if, and only if,* $\mathfrak{A} \leq_{\mathrm{fset}} \Delta_2$

(iv) $\mathfrak{A} \in \omega$T-AutStr *if, and only if,* $\mathfrak{A} \leq_{\mathrm{set}} \Delta_2$

Equivalently, one may formulate universality with respect to FO interpretations. Following [47] we define the *(finite) subset envelope* $\mathcal{P}_{(f)}(\mathfrak{A})$ of a structure $\mathfrak{A}$ by adjoining to $\mathfrak{A}$ its (finite) subsets as new elements ordered by set inclusion.

Definition 1.3.19 Given $\mathfrak{A} = (A, \{R_i\})$ write $P(A)$ for the set of all subsets of A. The *subset envelope* $\mathcal{P}(\mathfrak{A})$ is the structure with domain $P(A)$ and relations $R'_i := \{(\{a_1\}, \ldots, \{a_n\}) \mid (a_1, \ldots, a_n) \in R_i\}$ and the subset relation $\subseteq$ defined on $P(A)$. The *finite-subset envelope* $\mathcal{P}_f(\mathfrak{A})$ is the substructure of $\mathcal{P}(\mathfrak{A})$ whose domain is the set of finite subsets of A.

It is immediate that

$$\mathfrak{B} \leq_{\mathrm{(f)set}} \mathfrak{A} \iff \mathfrak{B} \leq_{\mathrm{FO}} \mathcal{P}_{(f)}(\mathfrak{A})$$

In particular, this yields natural universal structures, with respect to FO-interpretations, for each of the four classes of automatic structures.

Corollary 1.3.20 *(i)* $\mathcal{P}_f(\Delta_1)$ *is universal for* S-AutStr.

(ii) $\mathcal{P}(\Delta_1)$ *is universal for* ωS-AutStr.

(iii) $\mathcal{P}_f(\Delta_2)$ *is universal for* T-AutStr.

(iv) $\mathcal{P}(\Delta_2)$ *is universal for* ωT-AutStr.

VRS-Equational structures

Recall that the VRS-algebra of graphs extends the VR-algebra with the synchronous product operation and that VRS-equational systems define exactly the finite-tree automatic graphs (see Section 1.2.4 and Theorem 1.2.26).

A finite VRS-equational system whose unfolding is a linear VRS-term specifies a structure in S-AutStr. This happens if in the defining equations one of the arguments of each occurrence of $\oplus$ and of $\otimes_S$ is a finite graph (and so these act like unary operations). Conversely, for word-automatic presentations Equation (1.7) reduces to the following form:

$$X = \vartheta \left(\bullet^{\perp} \oplus \left(\vartheta_0 \otimes \vartheta_1(X) \right) \right) \tag{1.8}$$

This scheme matches the following type definition obtained by restricting (1.6) to words:

$$\mathcal{T} = \perp \oplus \left(\{0, 1\} \otimes \mathcal{T} \right) \tag{1.9}$$

This recursive definition of the set of words has the same advantage over (1.2) as (1.6) has over (1.4) when it comes to defining binary relations over words via structural induction, e.g. via finite automata. Over words we have the following special case of Theorem 1.2.26.

Theorem 1.3.21 (Colcombet [50])
For every countable structure $\mathfrak{A}$ the following are equivalent

(1) *$\mathfrak{A}$ is isomorphic to a word-automatic graph.*
(2) *$\mathfrak{A}$ is the restriction of some $\mathfrak{B}$ to its elements of a certain colour, where $\mathfrak{B}$ can be specified by a VR-equation $Z = \pi(X)$, where π simply forgets some of the structure of X, together with a VRS-equation for X of the form (1.8);*
(3) *$\mathfrak{A}$ is finite-subset interpretable in $(\mathbb{N}, \mathtt{suc})$.*

The equivalence of the first and the third item is a direct consequence of the classical correspondence of automata on words and monadic second-order logic of one successor and was already stated in Theorem 1.3.18. Nonetheless, this can also be inferred from the fact that the solution term obtained by unfolding (1.8) is (essentially) a periodic linear VRS-term that evaluates, via a finite-subset interpretation, to the word-automatic structure specified by equation (1.8).

More generally, let VRS^- denote the extension of VR with unary operations $X \mapsto G_0 \otimes_S X$ where G_0 is any finite graph. Moreover let us call a *chain interpretation* a subset interpretation in a tree where each of the subsets representing an element is linearly ordered by the ancestor relation of the tree. It is not hard to see that solutions of finite systems of VRS^--equations are finite-chain interpretable in a regular tree and that these in turn are word automatic [50].

1.3.5 Rational graphs

If we allow the more general *asynchronous automata* in the definition of an automatic presentation of a graph we get the notion of a rational graph. Thus vertices are labelled with finite words of a rational language over some finite alphabet Σ, and the edge relations are required to be rational subsets of $\Sigma^* \times \Sigma^*$.

With no aim for completeness we list below some results on rational graphs (asynchronous) in comparison with automatic graphs (synchronous). For a comprehensive treatment the reader is referred to [105].

The class of rational graphs strictly includes that of finite-word automatic graphs. In their seminal paper [87] Khoussainov and Nerode also introduced asynchronous automatic structures. As an example they gave an asynchronous automatic presentation of ω^ω, which is not in `S-AutStr` (see Theorem 1.4.12). Asynchronous automatic presentations of Cayley-graphs of finitely generated groups have also been considered as generalisations of 'automatic groups' [31].

The price of increasing expressiveness is a loss of tractability: in general, rational graphs do not have a decidable first-order theory. This renders rational graphs useless for representing data, let alone programs. However, in the context of formal language theory rational graphs seem to fill a gap. Considering rational graphs as infinite automata, i.e. as acceptors of languages, Morvan and Stirling have shown that they trace exactly the context-sensitive languages [108, 107] (see also [34] for a simplified approach). Rispal and others [123, 107, 34] have subsequently observed that this holds true for automatic graphs as well.

Although first-order queries on rational graphs are in general intractable there are some interesting decidable subclasses.

Morvan observed that by a result of Eilenberg and Schützenberger, graphs defined by rational relations over a *commutative* monoid have a decidable first-order theory. In particular, over the unary alphabet the monoid structure is isomorphic to $(\mathbb{N}, +)$ whence the unary rational graphs are those first-order definable in $(\mathbb{N}, +)$ [105]. Similarly, rational graphs over $(\mathbb{N}, +)^d$ are those having a d-dimensional first-order interpretation in $(\mathbb{N}, +)$.

Carayol and Morvan showed that on rational graphs that also happen to be trees (this is an undecidable property) first-order logic is decidable [36, 106]. The decision method is based on locality of FO as formulated by Gaifman and uses a compositional technique. The authors also exhibit a rational graph that is a finitely branching tree but is not finite-word automatic.

1.3.6 Generalisations

Automata with oracles

Consider an expansion Δ_i^O of $\Delta_i := ([i]^*, \mathtt{suc}_0, \cdots, \mathtt{suc}_{i-1})$ by a unary predicate $O \subset [i]^*$. Every MSO formula (with free MSO variables) of the expanded structure corresponds to a tree automaton with *oracle O*. An automaton with oracle is one that, while in position $u \in [i]^*$, can decide on its next state using the additional information of whether or

not $u \in O$. Thus for automata working on infinite words/trees the oracle O is simply read as part of the input. In the case of automata working on finite words/trees, the entire oracle is scanned, and so the acceptance condition should be taken appropriately (eg. Muller/Rabin).

Call a set O *decidable* if $\mathrm{MSO}(\Delta_i^O)$ is decidable, and *weakly decidable* if $\mathrm{wMSO}(\Delta_i^O)$ is decidable. Early work on decidable oracles used the contraction method to show that certain oracles on the line, such as $\{n! \mid n \in \mathbb{N}\}$, are decidable [67]. This was extended to the profinitely ultimately periodic words [38], which it turns out capture all the decidable unary predicates on the line [119, 120]. Nonetheless, it is still of interest to produce explicit examples of decidable oracles, see for instance [38, 74, 75, 7].

Definition 1.3.22 If in the definition of automatic presentation (1.3.2) we replace □-automata with □-automata with oracle O, we get a notion of □-*automatic presentation with oracle* O. A structure is called *automatic with oracle* if it has a □-automatic presentation with some oracle.

Example 1.3.23 The group of rationals $(\mathbb{Q}, +)$ has recently been shown to have no word-automatic presentation [136]. However it is finite-word automatic with oracle $\#2\#3\#4\cdots$. This is based on the idea, independently found by Frank Stephan and Joe Miller and reported in [114], that there is a presentation of $([0,1) \cap \mathbb{Q}, +)$ by finite words in which $+$ is regular, but the domain is not: every rational in $[0,1)$ can be expressed as $\sum_{i=2}^{n} \frac{a_i}{i!}$ for a unique sequence of natural numbers a_i satisfying $0 \le a_i < i$. The presentation codes this rational as $\#a_2\#a_3\#a_4\cdots$ where a_i is written in decimal notation (and hence has length less than the length of i written in decimal notation). Addition is performed with the least significant digit first, based on the fact that

$$\frac{a_i + b_i + c}{i!} = \frac{1}{(i-1)!} + \frac{a_i + b_i + c - i}{i!}$$

where $c \in \{0,1\}$ is the carry in.

We immediately have that a structure is (finite-)word/tree automatic with oracle O if and only if it is (finite) set interpretable in Δ_1^O/Δ_2^O. Hence we have the following generalisation of the Fundamental Theorem and its corollaries (1.3.4).

Theorem 1.3.24 (i) Definability*: Say $(\mathfrak{d}, f)$ is a □-automatic presentation with oracle O of a structure $\mathfrak{A}$ and $\varphi(\overline{x})$ is a FO-formula in the*

signature of $\mathfrak{A}$ defining a k-ary relation R over $\mathfrak{A}$. Then the relation $f^{-1}(R)$ is recognised by an $\square$-automaton with oracle O.

(ii) Interpretations: *The class of $\square$-automatic structures with oracle O is closed under FO-interpretations.*

(iii) Decidability: *The previous statements can be made effective under the following conditions.*

    1 *For $\square \in \{\mathtt{word}, \ \mathtt{tree}\}$ we require that* $\mathrm{wMSO}(\Delta_i^O)$ *be decidable.*

    2 *For $\square \in \{\omega\text{-}\mathtt{word}, \ \omega\text{-}\mathtt{tree}\}$ we require that* $MSO(\Delta_i^O)$ *be decidable.*

    In particular, under these conditions, every $\mathfrak{A}$ that is $\square$-automatic with oracle O has decidable FO-*theory.*

Of course Δ_i^O can be viewed as a coloured tree. As in Corollary 1.3.20 we have universal structures with respect to FO-definability. For instance $\mathcal{P}(\Delta_2^O)$ is universal for $\omega\mathtt{T\text{-}AutStr}$ with oracle O. The following result concerns finite-set interpretations in arbitrary trees.

Theorem 1.3.25 ([47])   *To every finite set interpretation $\mathcal{I}$ one can effectively associate a* wMSO *interpretation $\mathcal{J}$ such that for every tree t and structure $\mathfrak{A}$ if $\mathcal{P}_f(\mathfrak{A}) \cong \mathcal{I}(t)$ then $\mathfrak{A} \cong \mathcal{J}(t)$.*

This can be used to show that certain structures, such as the random graph, are not finite-tree automatic in the presence of any oracle [47].

1.3.7 Subclasses

In this section we restrict the complexity of the regular domains in automatic presentations to yield some of the more robust subclasses of `S-AutStr` and `T-AutStr`.

Polynomial domain

The most natural restriction is to consider presentations where the words and trees take labels from a unary alphabet $|\Sigma| = 1$. Word-automatic presentations over a unary alphabet were introduced and studied by Blumensath [20] and Rubin [89, 124].

The *density* of a language $L \subset \Sigma^*$ is the function $n \mapsto |L \cap \Sigma^n|$.

Definition 1.3.26   A structure is *unary automatic* if it has an injective word-automatic presentation in which the domain consists of words from a unary alphabet. A structure is *p-automatic* if it has an injective word-automatic presentation in which the domain has polynomial density. Let `1-AutStr` and `P-AutStr` denote these respective classes of structures.

Regular sets of polynomial density were characterised by Szilard et al. [131] as being a finite union of the form

$$D = \bigcup_{i < N} u_{i,1} v_{i,1}^* u_{i,2} \ldots u_{i,n_i} v_{i,n_i}^* u_{i,n_i+1} \tag{1.10}$$

where the degree of the polynomial of the density function is equal to the maximum of the n_i's. In [6] it was demonstrated that every finite-word-automatic presentation over a domain as in (1.10) can be transformed into an equivalent one (cf. Section 1.4.4) over a domain that is a regular subset of

$$a_1^* a_2^* \ldots a_n^*$$

where n is equal to the maximum of the n_i's. In particular, word-automatic presentations over a domain of linear density are unary automatic. This transformation yields a kind of normal-form of word-automatic presentations over a polynomially growing domain.

Theorem 1.3.27 ([6])   *A structure $\mathfrak{A}$ has an automatic presentation over a domain of density $\mathcal{O}(n^d)$ if, and only if, it has a d-dimensional interpretation in $\mathfrak{M} := (\mathbb{N}, <, \{\equiv_{(\text{mod } m)}\}_{m>1})$ if, and only if, it is finite-subset interpretable in $\Delta_1 := (\mathbb{N}, \text{suc})$ with subsets of size at most d.*

Corollary 1.3.28 ([113],[20])   *A structure $\mathfrak{A}$ is unary automatic if, and only if, it is first-order definable in $\mathfrak{M}$ if, and only if, it is MSO-interpretable in Δ_1.*

Unary automatic structures form a very restricted subclass of VR-equational structures and have a decidable MSO-theory. Using pumping arguments one can show that Presburger arithmetic $(\mathbb{N}, +)$ has no p-automatic presentation [20, 121]. On the other hand, the infinite grid is p-automatic but not unary automatic. Thus we have

$$\texttt{1-AutStr} \subsetneq \texttt{P-AutStr} \subsetneq \texttt{S-AutStr} .$$

The expansion of $\mathfrak{M}$ with the successor function suc and a constant for 0 admits quantifier elimination. Hence, every p-automatic structure can be interpreted in $(\mathbb{N}, 0, \text{suc}, <, \{\equiv_{(\text{mod } m)}\}_{m>1})$ using quantifier-free formulas.

Every p-automatic structure inherits the PSPACE upper-bound on the complexity of its first-order theory from $\mathfrak{M}$. This is as low as possible since FO model-checking is PSPACE-hard for any structure with at least two elements. Adding even the simplest form of iteration to

FO leads to undecidability. For every k-counter machine it is straightforward to construct a p-automatic presentation of its configuration graph where each configuration $(q, n_1, \ldots, n_k)$ is represented by the word $qc_1^{n_1} \cdots c_k^{n_k}$. It follows that the first-order theory with reachability FO[**R**] of a p-automatic structure is undecidable in general. In comparison, while unary automatic structures have a decidable MSO-theory, the FO(DTC) theory of $(\mathbb{N}, \mathrm{succ})$ interprets full first-order arithmetic and is therefore highly undecidable [20].

Observe, that graphs having rational presentation over a finitely generated commutative monoid (cf. Section 1.3.5) can be seen as analogues of p-automatic graphs. Indeed, every monoid element is represented by some word $g_1^{r_1} g_2^{r_2} \ldots g_n^{r_n}$ over the generators.

Finite-rank tree-automatic presentations

The analogue of *p*-automatic to tree-automatic structures is restricting to presentations involving trees of bounded rank. Intuitively the rank of a tree corresponds to its branching degree (which can be measured in terms of the Cantor-Bendixson rank).

Recall a Σ-labelled n-ary tree T is a function from a prefix-closed subset of $[n]^*$ to Σ. We say that T has *rank k* if its domain has polynomial density of degree at most k.

A finite-tree automatic presentation is called of *rank k* if for some regular language D of polynomial density of degree at most k the domain of every tree in the presentation is a subset of D. Collectively we speak of *bounded-rank tree-automatic presentations*. The class of structures with rank k presentations is denoted `k-T-AutStr`.

Example 1.3.29 The ordinal ω^{ω^k} has a rank $k + 1$ tree-automatic presentation.

Let $\mathcal{T}_k$ denote the structure corresponding to the unlabelled k-ary tree with domain $0^*1^* \cdots (k-1)^*$. Note that $\mathcal{T}_k$ is wMSO-interpretable in the ordinal ω^k (in the signature of order), and vice-versa.

Proposition 1.3.30 *The following are equivalent.*

- $\mathfrak{A}$ *is in* `k-T-AutStr`,
- $\mathfrak{A}$ *is finite-set interpretable in* $\mathcal{T}_k$ *(or equivalently in the ordinal* ω^k*),*
- $\mathfrak{A}$ *is the solution of a finite system of VRS-equations whose unfolding is a term-tree of rank k.*

The hierarchy is strict:

$$\texttt{S-AutStr} = \texttt{1-T-AutStr} \subsetneq \texttt{2-T-AutStr} \subsetneq \cdots \subsetneq \texttt{T-AutStr}.$$

Indeed, if $\texttt{k+1-T-AutStr} = \texttt{k-T-AutStr}$ for some k then the finite-subset envelope $\mathcal{P}_f(\omega^{k+1})$ would be finite-set interpretable in ω^k. But by Theorem 1.3.25 then ω^{k+1} is wMSO interpretable in ω^k, which is known not to be possible [98, Lemma 4.5].[15]

1.3.8 Comparison of classes

Since words are special cases of trees, and finite ones special cases of infinite ones, one immediately sees the inclusions indicated by the arrows in the figure. All the arrows except for the dotted one are known to be strict inclusions. We now discuss the separating examples as well as the double lines indicating equality of the classes when restricted to countable structures. Since $\omega\texttt{S-AutStr}$ and $\omega\texttt{T-AutStr}$ contain uncountable structures while $\texttt{S-AutStr}$ and $\texttt{T-AutStr}$ do not, we split our discussion along these lines.
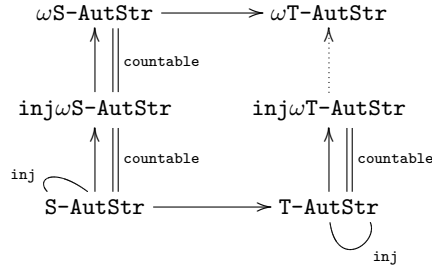

Figure 1.2 Relationship of classes of automatic structures

Countable structures

The structure $(\mathbb{N}, \times)$ separates $\texttt{T-AutStr}$ from $\texttt{S-AutStr}$ (see [20], or [88] for an alternative proof).

Every injective $\omega\texttt{S-AutStr}$ presentation of a countable structure can be effectively transformed into a $\texttt{S-AutStr}$ presentation. This is because a countable ω-regular set $X \subseteq \{0,1\}^\omega$ only contains ultimately periodic words, and moreover there is a bound on the size of the periods

---

[15] We thank Łukasz Kaiser for discussions on the notions of this section and Alex Rabinovich for providing the latter reference.

(which can be computed from an automaton for X). Similar facts hold for countable regular sets of infinite trees [115].

    The next theorem generalises this in the word case:

Theorem 1.3.31 ([85]) (i) *The countable structures in $\omega\texttt{S-AutStr}$ are precisely those in $\texttt{S-AutStr}$.*

(ii) *Given a (not necessarily injective) automatic presentation of some $\mathfrak{A} \in \omega\texttt{S-AutStr}$ it is decidable whether $\mathfrak{A}$ is countable or not, and if it is, an automatic presentation of $\mathfrak{A}$ over finite words can be constructed.*

    On the other hand, we do not know whether every countable structure in $\omega\texttt{T-AutStr}$ is in $\texttt{T-AutStr}$.

Uncountable structures

The only known non-trivial methods dealing with uncountable structures appear in [82]:

(i) The algebra $(\mathcal{P}(\{0,1\}^*), \cap, \cup, \cdot^c, \mathcal{F})$ from example 1.3.12(6) is an uncountable structure separating $\omega\texttt{T-AutStr}$ from $\omega\texttt{S-AutStr}$.

(ii) Recall Example 1.3.12(4) consisting of the algebra of subsets of $\mathbb{N}$ (call it $\mathcal{A}$) quotiented by having finite symmetric difference (call it $\approx$). Construct a variant structure as the disjoint union of $\mathcal{A}$ and $\mathcal{A}/_{\approx}$, with a unary predicate U identifying the elements of $\mathcal{A}$ and a binary relation R relating $a \in A$ to its representative in $\mathcal{A}/_{\approx}$. This uncountable structure separates $\omega\texttt{S-AutStr}$ from $\texttt{inj}\omega\texttt{S-AutStr}$.

1.4 More on word-automatic presentations

1.4.1 Beyond first-order logic

The Fundamental Theorem can be strengthened to include order-invariant definable formulas as well as certain additional quantifiers.

Generalised quantifiers

We briefly recall the definition of generalised quantifiers as introduced by Lindström.

Definition 1.4.1 Fix a finite signature $\tau = (R_i)_{i \leq k}$, where R_i has associated arity r_i. A *quantifier Q* is a class of τ-structures closed under isomorphism. Let σ be another signature. Given σ-formulas $\Psi_i(\overline{x}_i, \overline{z})$

with $|\overline{x}_i| = r_i$ ($i \leq k$), the syntax $Q\overline{x}_1, \cdots, \overline{x}_n(\Psi_1, \cdots, \Psi_k)$ has the following meaning on a σ-structure $\mathcal{A}$:

$$(\mathcal{A}, \overline{a}) \models Q\overline{x}_1, \cdots, \overline{x}_k(\Psi_1, \cdots, \Psi_k) \quad \text{iff} \quad (A; \Psi_1^{\mathcal{A}}(\cdot, \overline{a}), \cdots, \Psi_k^{\mathcal{A}}(\cdot, \overline{a})) \in Q,$$

where $\Psi^{\mathcal{A}}(\cdot, \overline{a})$ is the relation defined in $\mathcal{A}$ by Ψ with parameters $\overline{a}$. The *arity* of a quantifier is the maximum of the r_is. A quantifier is *n-ary* if its arity is at most n.

The extension of first-order logic by a collection **Q** of generalised quantifiers will be denoted FO[**Q**].

Examples 1.4.2 (i) The unary quantifier $\{(A; X) \mid \emptyset \neq X \subset A\}$ is 'there exists'.
(ii) The unary quantifier 'there exists infinitely many', written $\exists^{\infty}$, is the class of structures $(A; X)$ where X is an infinite subset of A.
(iii) The unary *modulo quantifier* 'there are k modulo m many' (here $0 \leq k < m$), written $\exists^{(k,m)}$, is the class of structures $(A; X)$ where X contains k modulo m many elements. Write $\exists^{\mathsf{mod}}$ for the collection of modulo quantifiers.
(iv) The unary *Härtig quantifier* is the class of structures $(A; P, Q)$ where $P, Q \subset A$ and $|P| = |Q|$.
(v) Every set $C \subset (\mathbb{N} \cup \{\infty\})^n$ induces the unary *cardinality quantifier* $Q_C = \{(A; P_1, \cdots, P_n) \mid (|P_1|, \cdots, |P_n|) \in C\}$. In fact, a given unary quantifier over signature $(R_i)_{i \leq k}$ is identical to some cardinality quantifier with $n = 2^k$.
(vi) The binary *reachability quantifier* is the class of structures of the form $(A; E, \{c_s\}, \{c_f\})$ where $E \subset A^2$, $c_s, c_f \in A$, and there is a path in the directed graph $(A; E)$ from c_s to c_f.
(vii) The k-ary *Ramsey quantifier* $\exists^{\text{k-ram}}$ is the class of structures $(A; E)$, $E \subset A^k$, for which there is an infinite $X \subset A$ such that for all pairwise distinct $x_1, \cdots, x_k \in X$, $E(x_1, \cdots, x_k)$.

The following general definition will allow us to compare the expressive strength of quantifiers.

Definition 1.4.3 Let Q be a quantifier, **Q** a collection of quantifiers, and τ the signature of Q. Say that Q *is definable in* **Q** if there is a sentence θ over the signature τ in the logic FO[**Q**] with $Q = \{\mathcal{A} \mid \mathcal{A} \models \theta\}$.

For instance, a structure $(A; X)$ satisfies $\exists^{(0,2)} z\, X(z) \;\vee\; \exists^{(1,2)} z\, X(z)$ if and only if X is finite. Hence $\exists^{\infty}$ is definable in $\{\exists^{(0,2)}, \exists^{(1,2)}\}$.

Of course the generalised quantifiers that interest us most are the ones, like $\forall$ and $\exists$, that preserve regularity.

Definition 1.4.4   Fix class $\mathcal{C}$ as one of `S-AutStr`, `T-AutStr`, ω`S-AutStr`, or `T-AutStr`. Let Q be a quantifier with signature $\tau = (R_i)_{i \leq k}$, where R_i has associated arity r_i. Say that quantifier Q *preserves regularity for the class* $\mathcal{C}$ if for every $n \in \mathbb{N}$, and every automatic presentation μ of a structure $\mathcal{A} \in \mathcal{C}$, every formula

$$Q\overline{x}_1, \cdots, \overline{x}_k (\Psi_1^{\mathcal{A}}(\overline{x}_1, \overline{z}), \cdots, \Psi_k^{\mathcal{A}}(\overline{x}_k, \overline{z}))$$

defines a relation R in $\mathcal{A}$ with $\mu^{-1}(R)$ regular (here $\overline{z} = (z_1, \cdots, z_n)$ and the Ψ_i are first-order $\mathcal{A}$-formulas).

Say that Q *preserves regularity effectively* if an automaton for $\mu^{-1}(R)$ can effectively be constructed from the automata of the presentation and the formulas Ψ_i.

Since not every structure is injectively presentable, we may restrict this definition to the class $\mathcal{C}$ of injectively presentable structures from ω`S-AutStr` (or ω`T-AutStr`). For this, replace 'automatic presentation' with 'injective automatic presentation' in the above definition.

Example 1.4.5   The reachability quantifier is not regularity preserving (for any of the classes). For otherwise, by Example 1.3.13, the set of starting configurations that drive a given Turing Machine to a halting state would be regular, and hence computable.

The first steps have been taken in exploring those quantifiers that preserve regularity.

Theorem 1.4.6   *Let $\mathcal{C}$ be any of the following classes of structures* `inj-`ω`T-AutStr`, ω`S-AutStr`, `T-AutStr`, `S-AutStr`.

(i) *The following unary quantifiers preserve regularity effectively for $\mathcal{C}$:* $\exists^\infty$, $\exists^{mod}$, $\exists^{\leq \aleph_0}$, $\exists^{> \aleph_0}$ *[20, 90, 94, 85, 9].*

(ii) *Every unary quantifier that preserves regularity for the class* `S-AutStr` *is already definable from* $\exists^{mod}, \exists^\infty$ *[125].*

The second item also implies that every unary quantifier that preserves regularity for the class `inj-`ω`S-AutStr` is already definable from $\exists^{mod}, \exists^\infty, \exists^{\leq \aleph_0}, \exists^{> \aleph_0}$. This is because for an ω-regular relation $R(\overline{x}, \overline{z})$ the cardinality of the set $R(-, \overline{c})$ (for any fixed parameter $\overline{c}$) is finite, countable or has size continuum [94].

Theorem 1.4.7 (see [125])   *Each k-ary Ramsey quantifier preserves regularity effectively for the class* `S-AutStr`.

Kuske and Lohrey observed that the proof of this theorem can be generalised to quantifiers of the form 'there exists an infinite set X satisfying θ', where θ is a property of sets closed under taking subsets. They use this to show that certain problems, while Σ_1^1-complete for recursive graphs, are decidable on automatic graphs [96].

Order-invariance

Definition 1.4.8 Fix a signature τ and a new symbol $\leq$. A formula $\phi(\overline{x})$ in the signature $\tau \cup \{\leq\}$ is called *order invariant* on a τ-structure $\mathcal{A}$ if for all tuples $\overline{a}$ from A and all linear orders $\leq_1$ and $\leq_2$ on A, we have that $(\mathcal{A}, \leq_1) \models \phi(\overline{a})$ if and only if $(\mathcal{A}, \leq_2) \models \phi(\overline{a})$. The *relation defined by the order invariant ϕ in $\mathcal{A}$* is the set of tuples $\overline{a}$ from A such that $(\mathcal{A}, \leq) \models \phi(\overline{a})$ for some (and hence all) linear orders $\leq$ on A.

The Fundamental Theorem can be extended on injective presentations to include order-invariant formulas in those cases where there is a regular linear ordering of the set $f^{-1}(A)$. On finite-words, finite-trees and ω-words there are regular linear orderings. However, we do not know if there is a regular linear ordering on the set of all ω-trees. On the other hand, certain separating examples from finite model theory are adaptable to the automatic world.

Proposition 1.4.9 ([5]) *There exists a structure $\mathfrak{B} \in$ S-AutStr and an order-invariant definable relation S^* in $\mathfrak{B}$ that is not definable in $\mathfrak{B}$ using any extension of* FO *with only unary quantifiers.*

1.4.2 Complexity of some problems

First-order theories

By Theorem 1.3.4 query-evaluation and model-checking for first-order formulas are effective on automatic structures. However, the complexity of these problems is in general non-elementary, i.e. it exceeds any fixed number of iterations of the exponential function. For instance the first-order theories of the universal structures $\mathcal{N}_k$ and $\mathcal{S}_{[k]}$ ($k \geq 2$) have non-elementary complexity [77] (cf. also the remark after Example 1.4.39).

There are various sensible ways of measuring model-checking complexity. First, one may fix a formula and ask how the complexity depends on the input structure. This measure is called *structure complexity*. On the other hand, *expression complexity* is defined relative to a fixed structure in terms of the length of the formula. Finally, one can look at the combined complexity where both parts may vary.

	Structure-Complexity[a]	Expression-Complexity
Model-Checking		
Σ_0	LOGSPACE-complete	ALOGTIME-complete
Σ_0 + func	NLOGSPACE	in quadratic time[16] and PTIME-complete
Σ_1	PTIME[17]	PSPACE-complete (EXPTIME-c. for `T-AutStr`)
Σ_2	PSPACE-complete[17]	EXPSPACE-complete (2EXPTIME-c. for `T-AutStr`)
Query-Evaluation		
Σ_0	LOGSPACE	PSPACE
Σ_1	PSPACE	EXPSPACE

Figure 1.3 Complexity of fragments of FO on automatic structures

[a] Structure complexity is measured in terms of the size of the largest deterministic automaton in the input presentation.

In [25] Blumensath and Grädel studied the expression and structure complexity of model-checking and query evaluation for quantifier-free and existential first-order formulas both in a relational signature and allowing terms in quantifier-free formulas. Their results are complemented by those of Kuske and Lohrey [95] on the expression complexity of Σ_1 (existential) and Σ_2 formulas of a relational signature over arbitrary word- and tree-automatic structures. Figure 1.3 provides a summary.

On certain subclasses of automatic structures there is better complexity. In section 1.3.7 above we have mentioned that the first-order theory of each structure allowing a word-automatic presentation of polynomial density is decidable in PSPACE. Kuske and Lohrey [101, 95] studied automatic structures whose Gaifman graphs are of *bounded degree*. Relying on locality of first-order logic they have identified the expression complexity of FO model checking on word-automatic and tree-automatic structures of bounded degree to be 2EXPSPACE-complete and 3EXPTIME-complete, respectively. The combined complexity remains 2EXPSPACE for word-automatic presentations and is in 4EXPTIME for tree-automatic presentations. For finer results we refer to [95].

[16] This is a generalisation of the quadratic solution of the word problem in automatic groups [31] (see Section 1.4.5).

[17] Model checking with a fixed Σ_1 formula reduces to a membership or non-emptiness test for an NFA. For fixed Π_2 formulas the problem is polynomially equivalent to the universality problem of NFAs, and thus PSPACE-complete. (We thank Anthony To for pointing out the error in [25].)

Beyond first-order

A fundamental problem in verification is deciding *reachability*: whether there is a path between specified source and target nodes. Since the configuration space of an arbitrary Turing machine is finite-word automatic, the halting problem can be reduced to the reachability problem on the configuration graph of a universal Turing-machine. Similar reductions show the undecidability, over (finite-word) automatic structures, of connectivity, isomorphism, bisimulation and hamiltonicity [25, 96].

On the other hand there are natural classes of automatic structures for which these problems become decidable (see Figure 1.1). For instance, VRA-equational graphs have a decidable FO-theory with reachability and are finite-tree automatic. Reachability and connectivity in locally-finite unary-automatic graphs are in fact decidable in PTIME. Bisimulation equivalence of HR-equational graphs of finite out-degree is decidable [128] (see section 1.2.2).

Finally we mention some cases where full MSO is decidable. Prefix recognisable structures (which include the unary automatic structures) are finite-word automatic. A structure of the form $(\mathbb{N}, <, C_1, \cdots, C_k)$ is called a colouring of the line. Every known finite-word automatic colouring of the line, and this includes every morphic sequence, has decidable MSO-theory (cf. Theorem 1.4.38 and see [7]). Furthermore, every word-automatic equivalence relation has a decidable MSO-theory. This follows from the above and the observation (Proposition 1.4.40) that if there are only finitely many infinite classes then the equivalence relation is FO-definable in some word-automatic colouring of the line [7].

Isomorphism problem

A measure of the complexity of a class of structures is the *isomorphism problem*, namely the problem of deciding, given two □-automatic presentations $\mathfrak{d}$ and $\mathfrak{d}'$, whether or not the structures they present are isomorphic.

The characterisations of the finite-word automatic Boolean algebras and ordinals [88, 63] imply that the isomorphism problem for each of these classes is decidable. Also, as noted, the isomorphism problem for equational graphs is decidable 1.2.10.

Configuration spaces of Turing machines are locally finite and the complexity of the isomorphism problem for locally-finite directed graphs in S-AutStr is Π_3^0-complete [124]. However, by massaging the configuration spaces we get that the isomorphism problem for automatic graphs is as hard as possible: Σ_1^1-complete. This is done by reducing the isomorphism

problem for computable structures, known to be Σ_1^1-complete, to that of automatic structures.

Theorem 1.4.10 ([124])   *The complexity of the isomorphism problem for each of the following classes of* S-AutStr *structures is Σ_1^1-complete: (i) undirected graphs, (ii) directed graphs, (iii) successor trees, and (iv) lattices of height* 4.

Problem 1.4.11   What is the exact complexity of the isomorphism problem for the following classes: [18]

(i) Automatic equivalence structures (easily seen to be Π_1^0).
(ii) Automatic linear orders.

Traces

Infinite edge-labelled graphs, when viewed as infinite automata, can accept non-regular languages. Naturally, context-free graphs accept precisely the context-free languages. Though prefix-recognisable graphs form a structurally much richer class they have the same language accepting power as context-free graphs (cf. Theorem 1.2.11 items (1) and (6)). Graphs in the Caucal hierarchy have the same accepting power as higher-order pushdown automata (see Theorem 1.2.16) tracing languages on the corresponding levels of the OI-hierarchy of [62]. The traces of GTRS-graphs form a language class in between the context-free and context-sensitive classes of the Chomsky hierarchy [99]. Rational graphs accept precisely the context-sensitive languages [108]. All context-sensitive languages can in fact be accepted by word-automatic graphs [123], cf. also [35] for a more accessible proof and finer analysis. Meyer proved that the traces of tree-automatic graphs are those languages recognisable in ETIME, i.e. in $2^{\mathcal{O}(n)}$ time [103].

1.4.3 Non-automaticity via pumping and counting

It is usually quite simple to show that a structure has an automatic presentation (if indeed it does have one!). On the other hand, there are only a handful of elementary techniques for showing that a structure has no automatic presentation. Most rely on the pumping lemma of automata theory.

---

[18] While this work has been in print, Kuske, Liu and Lohrey have greatly contributed to settling these and related questions. We refer to their forthcoming paper.

Sometimes we can provide a full characterisation of classes of automatic structures. The first non-trivial characterisation was for the word-automatic ordinals (in the signature of order).

Theorem 1.4.12 (Delhommé [63])

(i) *An ordinal α is in* S-AutStr *if, and only if, $\alpha < \omega^\omega$.*
(ii) *An ordinal α is in* T-AutStr *if, and only if, $\alpha < \omega^{\omega^\omega}$.*

A relation R is *$(n + m)$ locally finite* if for every $(x_1, \ldots, x_n)$ there are only finitely many $(y_1, \ldots, y_m)$ such that $R(\overline{x}, \overline{y})$ holds. Obviously, every functional relation $f(\overline{x}) = y$ is locally finite. Other examples of locally finite relations are equal-length el, length comparison $|y| < |x|$, and the prefix relation $y \prec_{\texttt{prefix}} x$. Note that local finiteness depends on the partitioning of the variables, e.g. $x \prec_{\texttt{prefix}} y$ is not locally finite.

A simple pumping argument gives the following important tool.

Proposition 1.4.13 (Elgot and Mezei [66]) *Let $R \subseteq (\Sigma^*)^{n+m}$ be a regular and locally finite relation. Then there is a constant k such that for all $\overline{x}, \overline{y}$ satisfying R, $max_j |y_j| \leq max_i |x_i| + k$. In particular, if f is a regular function then there is a constant k such that for every $\overline{x}$ in its domain we have $|f(\overline{x})| \leq max_i |x_i| + k$.*

Growth of generations

Consider a structure $\mathfrak{A}$ with functions $\mathcal{F} = \{f_1, \ldots, f_s\}$ and a sequence $E = \{e_0, e_1, e_2, \ldots\}$ of elements of $\mathfrak{A}$. The generations of E with respect to $\mathcal{F}$ are defined recursively as follows.

$$
\begin{aligned}
G^0_{\mathcal{F}}(E) &= \{e_0\} \\
G^{n+1}_{\mathcal{F}}(E) &= G^n_{\mathcal{F}}(E) \bigcup \{e_{n+1}\} \\
&\quad \bigcup \{f(\overline{a}) \mid f \in \mathcal{F}, \ a_i \in G^n_{\mathcal{F}}(E) \text{ for each } i \leq |\overline{a}|\}
\end{aligned}
$$

We are interested in how fast $|G^n_{\mathcal{F}}(E)|$ grows as a function of n.

Example 1.4.14 (i) Free semigroup on m generators: here $\mathcal{F} = \{\cdot\}$ and $E = \{e_1, \cdots, e_m\}$. For $m \geq 2$, since $G^m_{\mathcal{F}}(E) \supset E$, the set $G^{m+n}_{\mathcal{F}}(E)$ includes all strings over E of length at most 2^n; thus the cardinality of $G^{m+n}_{\mathcal{F}}(E)$ is at least a double exponential in n.

(ii) If $p : D \times D \to D$ is injective then for $\mathcal{F} = \{p\}$ and $E = \{e_1, e_2\}$ (distinct elements of D) $|G^n_{\mathcal{F}}(E)|$ is at least a double exponential.

We now iterate Proposition 1.4.13.

Proposition 1.4.15 ([87],[20, 25]) *Let $\mathfrak{A} \in$ S-AutStr and consider an injective presentation $\mathfrak{d}$ with naming function f. Let $\mathcal{F}$ be a finite set of functions FO-definable in $\mathfrak{A}$ and $E = \{e_0, e_1, \ldots\}$ a definable set of elements ordered according to length in $\mathfrak{d}$, i.e. $|f^{-1}(e_0)| \leq |f^{-1}(e_1)| \leq \cdots$. Then there is a constant k such that for every n and for every $a \in G_{\mathcal{F}}^n$ $|f^{-1}(a)| \leq kn$. In particular, $|G_{\mathcal{F}}^n| = 2^{\mathcal{O}(n)}$.*

In other words, the number of elements that can be generated using functions is at most a single exponential in the number of iterations. Continuing the previous examples, neither the free semigroup nor any bijection $f : D \times D \to D$ (also called a pairing function) is word-automatic. It is trickier to apply the proposition to show that Skolem arithmetic $(\mathbb{N}, \times)$ is not word-automatic (see [20, 25]). It is nevertheless tree-automatic, cf. Example 1.3.9.

The application of propositions 1.4.13 and 1.4.15 has been pushed to their limits:

Proposition 1.4.16 (i) *If a group $(G, \cdot)$ is word-automatic then every finitely generated subgroup is virtually Abelian (has an Abelian subgroup of finite index). In particular, a finitely generated group is in* S-AutStr *if, and only if, it is virtually Abelian [116, 114].*

(ii) *A Boolean Algebra (in the signature $(\cup, \cap, \cdot^c, \bot, \top)$) is in* S-AutStr *if, and only if, it is a finite power of the Boolean Algebra of finite or cofinite subsets of $\mathbb{N}$ [88]. In particular, the countable atomless Boolean Algebra is not in* S-AutStr.

(iii) *There is no infinite integral domain in* S-AutStr *[88].*

(iv) *No word-automatic structure (D, R) has a subset $N \subset D$ such that (N, R) is isomorphic to $(\mathbb{N}, \cdot)$, cf. [114].*

The proof of the first item starts with the observation that every finitely-generated group $G \in$ S-AutStr has polynomial density - that is, for every finite set $A = \{a_1, \ldots, a_k\}$ the function

$$\gamma(n) = |\{\prod_{i<n} c_i^{\sigma_i} \mid \forall i < n \, : \, c_i \in A, \, \sigma_i \in \{1, -1\}\}|$$

is bounded by a polynomial (this exploits associativity of the group operation). The rest of the proof uses powerful theorems of Gromov and Ershov (see [114] for a survey of word-automatic groups).

Number of definable subsets

Various countable random structures, such as the random graph, do not have word- or tree-automatic presentations [88, 63]. The approach

to proving these facts has a model-theoretic flavour: for a purported automatic presentation, it involves counting the number of definable subsets of elements represented by words of bounded length.

Consider the usual definition of a set defined by φ with parameter b that remains fixed:

$$\varphi(-, b)^{\mathfrak{A}} = \{a \in \mathfrak{A} \mid \mathfrak{A} \models \varphi(a, b)\} .$$

A finite set $X \subset A$ is *fully shattered by φ* if the cardinality of the family

$$\{\varphi(-, b)^{\mathfrak{A}} \cap X \mid b \in A\}$$

is as large as possible, namely $2^{|X|}$. For instance, Benedikt et al. [16] observe that in $\mathcal{S}_{[2]}$ each of the sets $\{0, 00, \dots, 0^n\}$ can be fully shattered by the formula $\varphi(x, b) = \exists z(\mathtt{suc}_1 z \prec_{\mathtt{prefix}} b \wedge \mathtt{el}(z, x))$.

By contrast, in every automatic presentation with naming function f and domain $D \subseteq \Sigma^*$, the image under f of each $D_{\leq n} := D \cap \Sigma^{\leq n}$ can only be linearly shattered by definable families.

Proposition 1.4.17 ([88, 63])   *In every automatic presentation of a structure $\mathfrak{A}$ with naming function f and for every formula φ :*

$$|\{\varphi(-, b)^{\mathfrak{A}} \cap f(D_{\leq n}) \mid b \in A\}| = \mathcal{O}(|f(D_{\leq n})|) .$$

As an application recall that the random graph is characterised by the property that for every partition of a finite set X of vertices into sets U and V, there is a vertex b connected to all elements of U and to no element of V. In other words, every finite set X of vertices is fully shattered by the edge relation as the parameter b is varied. So by Proposition 1.4.17 the random graph has no word-automatic presentation. Similar reasoning yields the following.

Proposition 1.4.18 ([88, 63])   *The following are not in* S-AutStr*: the random graph, the random partial order, the random K_n-free graph.*

Using Theorem 1.3.25 one can established non-automaticity of the random graph in a far more general sense.

Theorem 1.4.19 ([47])   *Neither the random graph nor the the free monoid on two generators is finite-tree automatic with any oracle.*

In fact neither is ω-word automatic with any oracle, as witnessed by the following theorem which follows from the proof of Theorem 1.3.31.

Theorem 1.4.20   *If a countable structure is ω-word automatic with oracle, then it is also finite-word automatic with (the same) oracle.*

1.4.4 Comparing presentations

When we think of an automatic structure we frequently have a particular automatic presentation in mind. Some structures have *canonical* presentations. For instance, $(a^*, <_{len})$ is arguably the canonical presentation of $(\mathbb{N}, <)$ and $(\{0,1\}^*, \mathtt{suc}_0, \mathtt{suc}_1, \prec_{\mathtt{prefix}}, \mathtt{el})$ is the canonical presentation of itself. Some well-known structures have *natural* presentations, none of which can be indisputably called canonical. The base $k \in \mathbb{N}$ $(k > 1)$ presentations of $(\mathbb{N}, +)$ can be considered equally natural; but then what about the Fibonacci numeration system? The field of *regular numeration systems*, though using a somewhat different terminology, investigates automatic presentations of $(\mathbb{N}, +)$ and ω-word automatic presentations of $(\mathbb{R}, +)$. Finally, there are *pathological* presentations that are used to pin down the relationship between definability in a structure and regularity in its presentations [90].

How are we to compare different automatic presentations of the same structure? What are the crucial aspects of a presentation that distinguish it from others?

Canonical representations of context-free graphs were investigated by Sénizergues. In [127] a p-structure for a graph G is a PDA $\mathcal{A}$ (having no ϵ-transitions) together with an isomorphism between the configuration graph of $\mathcal{A}$ and G. Furthermore, a p-structure for G is *P-canonical* if the distance in G between a vertex v and the root is equal to the stack height of the configuration representing v (cf. [112]'s notion of a canonical automaton for a context-free graph; and [41, 44]). For a fixed graph G Sénizergues considers two p-structures equivalent if there is a rational isomorphism between them, and shows that every equivalence class of p-structures contains a P-canonical one [127].

An example from the theory of numeration systems is provided by the celebrated result of Cobham and Semenov. Recall that naturals p and q are called *multiplicatively independent* if they have no common power (ie. $p^k \neq q^l$ for all $k, l \geq 1$) and *multiplicatively dependent* otherwise.

Theorem 1.4.21 (Cobham-Semenov [19], cf. [26, 19, 109])
The following dichotomy holds for $p, q \geq 2$.

   (i) If p and q are multiplicatively dependent then a relation $R \subseteq \mathbb{N}^r$ is regular when coded in base p iff it is regular when coded in base q.

   (ii) If p and q are multiplicatively independent then a relation $R \subseteq \mathbb{N}^r$ is regular in both base p and base q iff R is FO-definable in $(\mathbb{N}, +)$.

---

[19] Cobham proved it for sets; Semenov later extended it to arbitrary relations.

The meaning of (i) is that, for instance, bases 2^l and 2^k are expressively equivalent. There is a very simple coding translating numerals between these bases, which bijectively maps blocks of k digits in the first system to blocks of l digits in the second system. Every pair of multiplicatively dependent numeration systems are linked by similar translations.

According to (ii) the base 2^k presentation is as different as it can be from, say, the base 3 presentation. This point is further stressed by the following result of Bés based on the work of Michaux and Villemaire.

Theorem 1.4.22 ([18])   *Let p and q be multiplicatively independent, and $R \subseteq \mathbb{N}^r$ regular when coded in base q, but not first-order definable in $(\mathbb{N}, +)$. Then the first-order theory of $(\mathbb{N}, +, |_p, R)$ is undecidable.*

On a similar note we introduce the following general notions.

Definition 1.4.23 (Subsumption and equivalence)
Consider two $\square$-automatic presentations of some structure $\mathfrak{A}$ with naming functions f and g, respectively. We say that f *subsumes* g ($g \preccurlyeq f$) if for every relation R over the domain of $\mathfrak{A}$, if $g^{-1}(R)$ is $\square$-regular then $f^{-1}(R)$ is $\square$-regular. If both $f \preccurlyeq g$ and $g \preccurlyeq f$ then we say that the two presentations are *equivalent* and write $f \sim g$. Moreover, we say that a $\square$-automatic presentation of $\mathfrak{A}$ is *prime* if it is subsumed by all other $\square$-automatic presentations of $\mathfrak{A}$.

word-automatic presentations

The definition of equivalence of automatic presentations is modelled on case (i) of Theorem 1.4.21. In [5] it has been shown that two finite-word automatic presentations are equivalent if and only if the transduction translating names of elements from one presentation to the other is computable by a *semi-synchronous transducer*: a two-tape finite automaton processing its first tape in blocks of k letters and its second tape in blocks of l letters for some fixed positive k and l. (Note that, except in trivial cases, k/l is uniquely determined [5].)

Theorem 1.4.24 ([5])   *Two finite-word automatic presentations of some $\mathfrak{A} \in \texttt{S-AutStr}$ with naming functions $f_i : D_i \to A$, $i \in \{1, 2\}$, are equivalent if, and only if, the transduction $T = \{(x, y) \in D_1 \times D_2 \mid f_1(x) = f_2(y)\}$ translating names of elements from one presentation to the other is semi-synchronous rational.*

Corollary 1.4.25   *Let f_1 and f_2 be naming functions of equivalent automatic presentations of $\mathfrak{A}$. Then there is a constant C such that*

for every n-ary relation R over $dom(\mathfrak{A})$ and for every automaton $\mathcal{A}_1$ recognising $f_1^{-1}(R)$ there is an automaton $\mathcal{A}_2$ of size $|\mathcal{A}_2| \leq C^n \cdot |\mathcal{A}_1|$ recognising $f_2^{-1}(R)$, and vice versa.

Let $\mathfrak{U}$ be one of the universal finite-word automatic structures $\mathcal{S}_\Sigma$ (for $|\Sigma| > 1$), $\mathcal{P}_f(\Delta_1)$, or $(\mathbb{N}, +, |_k)$ (for $k > 1$). Using semi-synchronous translations one can establish the following.

Theorem 1.4.26 ([5, 6])   *The universal structure $\mathfrak{U}$ has only a single word-automatic presentation up to equivalence.*

The assertion of the theorem can be reformulated as follows.

Corollary 1.4.27   *For a relation R, the expansion $(\mathfrak{U}, R)$ is in* S-AutStr *if, and only if, R is* FO-*definable in $\mathfrak{U}$.*

The prime presentation of a structure, if one exists, is unique up to equivalence, hence may as well be called canonical. The unary presentation of $(\mathbb{N}, <)$ is a prime word-automatic presentation. It is, however, not a prime presentation of $(\mathbb{N}, \mathtt{suc})$, which allows, for every $m > 1$ a word-automatic presentation in which divisibility by m is not regular [90]. It can be inferred that $(\mathbb{N}, \mathtt{suc})$ has no prime presentation.

Recall Theorem 1.3.27 stating that each word-automatic presentation, of structure $\mathfrak{A}$, over a domain of polynomial density of degree d directly corresponds to a d-dimensional interpretation of $\mathfrak{A}$ in the structure $\mathfrak{M} = (\mathbb{N}, <, \{\equiv_{(\bmod m)}\}_{m>1})$, and hence also in $(\mathbb{N}, +)$. So every p-automatic structure has infinitely many pairwise incomparable word-automatic presentations 'inherited' from $(\mathbb{N}, +)$, namely, based on different numeration systems.

In fact, $\mathfrak{M}$ allows a non-trivial 2-dimensional interpretation in itself. Simply consider the lexicographic ordering of all pairs (n_1, n_2) such that $n_1 \geq n_2$ as an interpretation of $(\mathbb{N}, <)$ and observe that moduli of positions within the lexicographic ordering of tuples can be expressed in terms of moduli of their components. Thus, by composing interpretations, every p-automatic presentation of $\mathfrak{M}$ is properly subsumed by other p-automatic presentations with domains of asymptotically greater polynomial densities. This carries over to all p-automatic structures.

In contrast, from results of [5, 8] it follows that $g \preccurlyeq f$ implies $g \sim f$ for any two word-automatic presentations of a given structure, provided that either both f and g have domains of exponential density, or both have a domain of polynomial density of the same degree.

Therefore, the height of the partial order of word-automatic presen-

tations of $\mathfrak{A}$ under subsumption and modulo equivalence is ω if $\mathfrak{A}$ is p-automatic and 1 if $\mathfrak{A}$ is not p-automatic. It is not known whether the width of the subsumption order modulo equivalence is always one or infinite for word-automatic structures that are not p-automatic.

tree-automatic presentations

Colcombet and Löding [47] investigated the power of finite-subset interpretations applied to arbitrary trees. In our terminology these are tree-automatic presentations with arbitrary oracles.

In the tree-automatic model the analogue of Theorem 1.4.26 does not hold. A tree-automatic presentation of $\mathcal{P}_f(\Delta_2)$ incomparable with the natural one can be forged simply by 'folding each tree in half about the vertical axis', i.e. taking the mirror image of the subtree below the right child of the root and smoothly combing it together with the untouched left half, e.g. as in Example 1.3.11(3). Despite this, the fact concerning primality of the natural presentation of the universal structure holds in an even stronger sense.

Proposition 1.4.28 ([47, Lemma 5.6]) *The natural tree-automatic presentation with oracle O and with the identity naming function of the finite-subset envelope $\mathcal{P}_f(\mathfrak{T}_O)$ of the oracle tree $\mathfrak{T}_O$ is a prime presentation with respect to tree-automatic presentations with arbitrary oracle.*

In particular, 'the' word-automatic presentation of $\mathcal{P}_f(\Delta_1)$ and the natural tree-automatic presentation of $\mathcal{P}_f(\Delta_2)$ are both prime even among tree-automatic presentations with arbitrary oracles. This is complemented by the following result of [47].

Theorem 1.4.29 *All tree-automatic presentations of $\mathcal{P}_f(\Delta_1)$ are equivalent.*

Therefore, the same holds true for all of the universal structures from Theorem 1.4.26.

1.4.5 Other notions of automaticity

Specific automatic presentations have been employed in other mathematical fields: computational group theory [31], symbolic dynamics [13], numeration systems (of integers or reals) [76], and infinite sequences represented in natural numeration systems [2, 26, 4]. In this section we survey natural presentations of certain structures that have mostly been considered independently of the general theory of automatic structures.

Automatic groups

Thurston (1986) motivated by work of Cannon on hyperbolic groups introduced the notion of automatic groups. A finitely generated group G is *automatic* in this sense if for some set of semigroup generators S and associated canonical homomorphism $f : S^* \to G$

(i) there is a regular language $W \subset S^*$ so that f restricted to W is surjective,

(ii) for every s a generator from S or the group identity, the following binary relation over W is regular:

$$\{(u,v) \mid f(u) = f(v)s\}.$$

This is in fact an algebraic notion: it does not depend on the particular choice of generators. From the automata presenting the group one can extract a finite presentation of the group, and a quadratic-time algorithm deciding the word problem.

Proposition 1.4.30 (k-fellow traveler property)   *A group G with semigroup generators $S = \{s_1, \ldots, s_r\}$ is automatic if, and only if, there exists a regular set $W \subseteq S^*$ and $k \in \mathbb{N}$ such that $f \mid_W$ is surjective and W satisfies the k-fellow traveler property:*

$$\forall u, v \in W \ \text{ with } d(u,v) \leq 1 \ \forall i \leq \max\{|u|, |v|\} : d(u_1 \ldots u_i, v_1 \ldots v_i) \leq k$$

where $d(u,v)$ denotes the length of the shortest path between u and v in the Cayley graph of G with generators S.
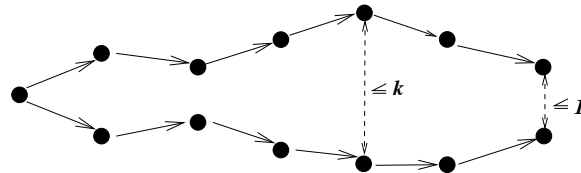

Figure 1.4 k-fellow traveler property.

Virtually Abelian groups and Gromov's word hyperbolic groups constitute important examples of automatic groups in this sense. Major results of this programme are presented in [31] (see also the introductions by Farb [71] and by Choffrut [46]).

More recently, this notion has been extended to semigroups [29, 30, 84, 28] and monoids [83, 129, 102].

Let us compare the following three notions: (i) groups whose multiplication function admits a word-automatic presentation, (ii) finitely generated automatic groups, and (iii) finitely generated groups with a Cayley graph admitting a word-automatic presentation. It is known [116] that a finitely generated group allows a word-automatic presentation of type (i) iff it is virtually Abelian. All virtually Abelian finitely generated groups are automatic in the sense of this subsection. Hence (i) implies (ii) for finitely generated groups. Furthermore, by definition, the Cayley graph of every automatic group has a word-automatic presentation. Hence (ii) implies (iii), but the converse fails. As Sénizergues has pointed out the Heisenberg group is not automatic even though its Cayley graph has an automatic presentation. For further reading we recommend the survey by Nies [114].

Generalised numeration systems

The theory of *generalised numeration systems* [76] is concerned with representations of $\mathbb{N}$ and $\mathbb{R}$ in various bases and using different (possibly negative) digits. In general, the basis $U_0 < U_1 < U_2 < \ldots$ of the system does not have to be the sequence of powers of a natural. One considers bases satisfying appropriate linear recursions, or alternatively powers of a base β which is the greatest root of a polynomial of a certain type. The study of generalised numeration systems goes back to Rényi who in 1957 introduced β-expansions.

Without going into the particulars of this very rich field we point out that a number may have more than one representation in a given numeration system. Thus from a practical perspective one is interested in *normalised* numerals obtained via the *greedy* algorithm. Normalised numerals are ordered according to $<_{\mathrm{llex}}$ (length and then lexicographically, most significant digit first). A regular set of (normalised) numerals $N \subseteq [d]^*$ over the set of digits $0, \ldots, d-1$ is simply an automatic copy of $(\mathbb{N}, <)$ of the form $(N, <_{\mathrm{llex}})$.

A fundamental question in this context asks under which circumstances addition can be computed by a synchronous finite automaton. When this is the case one speaks of a *regular numeration system*. On this matter we refer to [76] and the references therein.

Example 1.4.31   The *Fibonacci numeration system* is a prominent example of a regular numeration system. It has the Fibonacci numbers $1, 2, 3, 5, 8, \ldots$ as its basis, and the binary digit set. The normalised numerals delivered by the greedy algorithm are $\varepsilon, 1, 10, \ 100, 101, 1000,$

1001, 1010, 10000, 10001, . . . in the length-lexicographic ordering. They are the binary strings avoiding 11 as a factor since greedy normalisation prefers 100 to 11. Naturally, 10^n represents the nth Fibonacci number.

More generally we ask how can one classify the word-automatic presentations of $(\mathbb{N}, +)$? Or those of $(\mathbb{N}, <)$? Below we survey known classes of automatic presentations of expansions of $(\mathbb{N}, <)$ by unary predicates, i.e. infinite sequences.

Automatic sequences

The theory of *automatic sequences* [2] studies ω-words representable in more-or-less standard numeration systems. Presentations of primary concern are those of base $k \in \mathbb{N}$, or of base $-k$, and possibly involving negative digits.

Definition 1.4.32 A sequence $s : \mathbb{N} \to \Sigma$ is *k-automatic* if for every $a \in \Sigma$ the set N_a of numerals in the standard base k numeration system representing all positions n such that $s(n) = a$ constitutes a regular language.

These *k-automatic sequences* have been characterised in both algebraic and logical terms. In order to formulate another characterisation some notions are required. A morphism $\varphi : \Gamma^* \to \Sigma^*$ is said to be *k-uniform* if $|\varphi(a)| = k$ for each $a \in \Gamma$. *Codings* are 1-uniform morphisms. A morphism $\varphi : \Gamma^* \to \Gamma^*$ is *prolongable* on some $a \in \Gamma$ if a is the first symbol of $\varphi(a)$. In this case the sequence $(\varphi^n(a))_{n \in \mathbb{N}}$ converges to either a finite or infinite word, which is a fixed point of φ, denoted $\varphi^\omega(a)$.

Theorem 1.4.33 ([26, 2]) *For any sequence $s : \mathbb{N} \to \Sigma$ the following are equivalent:*

(1) s is k-automatic;
(2) the k-kernel of s: $\{(s_{nk^m+r})_n \mid r, m \in \mathbb{N},\ r < k^m\}$ is finite;
(3) the sets $s^{-1}(a)$ are FO-definable in $(\mathbb{N}, +, |_k)$ for each $a \in \Sigma$;
(4) $s = \sigma(\tau^\omega(a))$ for some k-uniform morphism τ on some Γ^* and a coding $\sigma : \Gamma \to \Sigma$;
(5) (assuming k is a prime and $\Sigma \subseteq \{0, \ldots, k-1\}$): the formal power series $S(x) = \sum_n s_n x^n \in \mathbb{F}_k[[x]]$ is algebraic over $\mathbb{F}_k[x]$.

For example, consider the morphism $\tau : 0 \mapsto 01, 1 \mapsto 10$. Its fixed point $\tau^\omega(0)$ is the *Thue-Morse sequence* $t = 01101001100101101001\ldots$. This is a truly remarkable sequence bearing a number of characterisations and combinatorial properties [3]. For instance, its nth digit is 1 if, and only

if, the binary numeral of n contains an odd number of 1's. The 2-kernel of t is $\{t, \bar{t}\}$, where $\bar{t}$ is obtained from t by flipping every bit.

Morphic words

One obtains a definition of *morphic words* by relaxing characterisation (4) of the above theorem. Morphic words thus constitute a generalisation of automatic sequences. They and their relatives have been extensively studied in the context of formal language theory, Lindenmayer systems and combinatorics on words.

Definition 1.4.34 *Morphic words* are those of the form $\sigma(\tau^\omega(a))$ for arbitrary homomorphism τ prolongable on a and arbitrary homomorphism $\sigma : \Gamma^* \to \Sigma^*$ extended to ω-words in the obvious way.

Example 1.4.35 Consider $\tau : a \mapsto ab, b \mapsto ccb, c \mapsto c$ and $\sigma : a, b \mapsto 1, c \mapsto 0$ both homomorphically extended to $\{a, b, c\}^*$. The fixed point of τ starting with a is the word $abccbccccbc^6b\ldots$, and its image under σ, $110010^410^610^81\ldots$, is the characteristic sequence of the set of squares. In general, for every strictly positive $\mathbb{N}$-rational sequence (s_k) the characteristic sequence of the set $\{\sum_{k=0}^n s_k \mid n \in \mathbb{N}\}$ is morphic [38]. This result also follows from Proposition 1.4.37.

While k-automatic sequences allow automatic presentations over the set of standard base k numerals, the above example suggests that morphic words may need generalised numeration systems. Indeed, every morphic word is automatically presentable in the following sense.

Consider a finite ordered alphabet $\Gamma = \{a_1 < a_2 < \ldots < a_r\}$. In the induced *length-lexicographic order*, denoted $<_{\mathrm{llex}}$, words over Γ are ordered according to their length first, while words of the same length are ordered lexicographically. Thus $(D, <_{\mathrm{llex}})$ provides an automatic presentation of $(\mathbb{N}, <)$ for every infinite regular language D over Γ. Base k as well as so called generalised numeration systems are special cases of this scheme. The following notion thus generalises Definition 1.4.32.

Definition 1.4.36 We say that an ω-word $w : \mathbb{N} \to \Sigma$ is *length-lexicographically presentable* if there is an automatic presentation $(D, <_{\mathrm{llex}})$ of $(\mathbb{N}, <)$ with naming function $f : D \to \mathbb{N}$ such that the sets $f^{-1}(w^{-1}(a))$ are regular for each $a \in \Sigma$.

It is not hard to see that an ω-word is length-lexicographically presentable if and only if it is morphic. There is a perfectly natural correspondence between the morphisms generating a word and the automaton

recognising the set of 'numerals', which, when length-lexicographically ordered, give an automatic presentation of the morphic word.

Proposition 1.4.37 ([122]) *An ω-word w is length-lexicographically presentable if, and only if, w is morphic.*

We illustrate the transformation from one formalism to the other on the characteristic sequence of squares from Example 1.4.35. Recall that it is generated by the following morphism τ and final substitution σ

$$\begin{aligned} \tau : \quad & a \mapsto ab \quad b \mapsto ccb \quad c \mapsto c \\ \sigma : \quad & a \mapsto 1 \quad\;\; b \mapsto 1 \quad\;\;\; c \mapsto 0 \end{aligned}$$

The idea is to interpret symbols $\{a, b, c, 0, 1\}$ as states. Without loss of generality, the alphabets of the ranges of σ and τ are disjoint. The alphabet Γ of the automatic presentation consists of digits ranging from 0 to $|\tau| + |\sigma| - 1$, where $|\tau|$ is the maximum of $|\tau(x)|$ with $x \in \{a, b, c\}$ and $|\sigma|$ is defined similarly. Letters of the alphabet, ordered as usual, are used to index positions within the right-hand side of a τ-rule, or, when larger, positions inside the right-hand side of a substitution via σ.
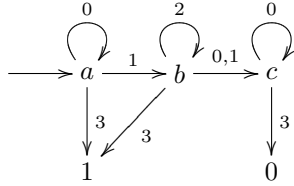
The domain D of the presentation is recognised by the above automaton with both 1 and 0 as final states. With only 1 as a terminal state, the automaton recognises the numerals representing a square relative to the length-lexicographic enumeration of D. Starting with a deterministic automaton this transformation can be reversed producing a morphism τ representing the transition function linearised according to the ordering on the alphabet and with σ identified by the terminal states.

The MSO-theory of the structure $(\mathbb{N}, <, (w^{-1}(a))_a)$ for morphic w is decidable [38]. Moreover, the class of morphic words is closed under MSO-definable recolourings, i.e. under *deterministic generalised sequential mappings* [118]. These results are generalised by the following one, which can be seen as an extension of the Fundamental Theorem 1.3.4.

Theorem 1.4.38 ([7]) *Let $\mathfrak{d} = (D, <_{\mathrm{llex}}, \overline{P})$ be a length-lexicographic presentation of a morphic word w and let $\varphi(\overline{x})$ be an $\mathrm{MSO}[<, \overline{P}]$-formula*

having only first-order variables free. Then there is an automaton $\mathcal{A}$, computable from $\mathfrak{d}$ and φ and such that $(\mathfrak{d}, \mathcal{A})$ is a word-automatic presentation of w expanded by the relation defined by φ.

Caucal has shown that morphic sequences can be constructed as graphs on the second level of the pushdown hierarchy (cf. Definition 1.2.14) [43]. However, there are automatically presentable ω-words on higher levels as well.

Higher-order morphic words

Higher-order morphic words were introduced in [4, 7]. Morphic words of order k can be defined either in the style of Definition 1.4.34 based on a notion of 'morphisms of order-k stacks' or similar rules, or as in Definition 1.4.36 as those having an automatic presentation using the 'k-fold nested length-lexicographic order' induced by an ordered alphabet. Theorem 1.4.38 extends to these automatic presentations of higher-order morphic words. The classes of order k morphic words form an infinite hierarchy, and are constructible on the $2k$-th level of the pushdown hierarchy [7].

Example 1.4.39 As an example we mention the Champernowne word (cf. Example 1.3.23) obtained by concatenating decimal numerals in their usual order:

$$C = 1234567891011121314\ldots$$

It is on the second level of this hierarchy (and on the fourth level of the pushdown hierarchy). Consider the level 2 morphism Δ given by the following intuitive production rules

$$\begin{aligned} S_x &\rightarrow S_x A_{\tau_1(x)} \ldots A_{\tau_9(x)} \\ A_x &\rightarrow A_{\tau_0(x)} A_{\tau_1(x)} \ldots A_{\tau_9(x)} \end{aligned}$$

where each τ_i is a (level 1) morphism of words in the usual sense mapping each digit $d \in \{0, \ldots, 9\}$ to d and $\#$ to $i\#$. Applying Δ repeatedly to the initial level 2 stack $S_\#$ yields the following converging sequence

$$\begin{aligned} S_\# &\rightarrow S_\# A_{1\#} A_{2\#} \ldots A_{9\#} \\ &\rightarrow S_\# A_{1\#} A_{2\#} \ldots A_{9\#} A_{10\#} \ldots A_{19\#} \cdots \cdots A_{90\#} \ldots A_{99\#} \\ &\rightarrow \cdots \end{aligned}$$

Hence C can be specified as $C = \sigma(\Delta^\omega(S_\#))$ with the morphism σ erasing all $\#$'s while preserving the other (level 1) symbols.

To give a word-automatic presentation we take the domain D to be

comprised of all words of the form $d_1 m_1 d_2 m_2 \ldots d_s m_s$ with $d_1 d_2 \ldots d_s$ a conventional decimal numeral and $m_1 m_2 \ldots m_s = \mathtt{o}^i \mathtt{xo}^{s-i-1}$ a marker indexing the ith digit of this numeral. Elements of the domain are ordered using the length-lexicographic ordering in a nested fashion: comparing numerals (i.e. odd positions) first, and then according to the position of the marker $\mathtt{x}$.

The Champernowne word contains every finite word over $\{0, 1, \ldots, 9\}$ as a factor. The satisfiability problem of first-order logic on finite words, known to be non-elementary [79], is thus expressible in the FO theory of the Champernowne word, which is therefore also non-elementary. For the same reason the Champernowne word is not morphic. Every morphic word is MSO-definable in the Champernowne word, and every word-automatic equivalence structure having only finitely many infinite equivalence classes is interpretable in a second-order morphic word [7].

Proposition 1.4.40   *Consider $\mathfrak{A} = (A, E)$ with E an equivalence relation having, for each $n > 0$, $f(n) \in \mathbb{N}$ many equivalence classes of size n, and no infinite classes. Then $\mathfrak{A} \in \mathtt{S\text{-}AutStr}$ if, and only if, there is a second-order morphic word $w = 0^{m_0} 1 0^{m_1} 1 0^{m_2} 1 \ldots$ such that $f(n) = |\{i \mid m_i = n\}|$.*

It remains open whether the decidability and definability results for MSO hold for all word-automatic infinite sequences. We are intrigued whether the isomorphism problem of automatic ω-words, or more broadly for automatic scattered linear orders, is decidable. Already for morphic words this is a notorious long-standing open problem.

1.5 Automatic Model Theory

We may reformulate the original problem — we seek a class of finitely-presentable structures $\mathcal{C}$ that has an interesting model theory and lies somewhere between the finite structures (finite model theory) and all structures (classical model theory).

The richest and oldest class consists of the computable structures — these are structures whose domain and atomic relations are computable by Turing machines [70]. In computable model theory, a common theme is to take classical results from mathematics and model theory and to see to what extent they can be made effective. Here are two illustrative observations:

(i) A computable (consistent) first-order theory has a computable model. Indeed, Henkin's construction can be seen as an algorithm computing the domain and atomic relations.

(ii) Every two computable presentations of the rational ordering $(\mathbb{Q}, <)$ are computably isomorphic. Again, the standard back-and-forth argument can be seen as an algorithm building the isomorphism.

The program of feasible mathematics in the 1980's included the development of polynomial-time model theory [45]. However, every relational computable structure is isomorphic (in fact computably isomorphic) to a polynomial-time structure. Automatic structures can be seen as a further restriction of this class, and in fact this is the motivation in [87]. In this section we discuss some aspects of the model theory of automatic structures, a subject still in its infancy.

We split our discussion along two lines: model theory of the class `S-AutStr`, and model theory of the particular universal structure $\mathcal{S}_{[2]}$ (cf. Theorem 1.3.17).

1.5.1 Model theory restricted to the class of word-automatic structures

Blumensath shows that, as expected, certain notions of model theory fail when restricted to the class of automatic structures.

Proposition 1.5.1 (i) *It is undecidable whether an* FO-*formula has a word-automatic model.*

(ii) *The following properties fail on the class of word automatic structures: compactness, Beth, Interpolation, and Łos-Tarski.*

The proofs are based on the observation that there is a FO formula which has automatic models of every finite cardinality but no infinite automatic models.

Löwenheim-Skolem

An automatic version of the Downward Löwenheim-Skolem Theorem would say that every uncountable ω-automatic structure has a countable elementary substructure that is also ω-automatic. Unfortunately this is false since there is a first-order theory with an ω-automatic model but no countable ω-automatic model. Indeed, consider the first-order theory of atomless Boolean Algebras. Kuske and Lohrey [94] have observed that it

has an uncountable ω-automatic model (namely the algebra from Example 1.3.12.4). However, Khoussainov et al. [88] show that the countable atomless Boolean algebra is not automatic and so, by Theorem 1.4.20, not ω-automatic either.

Here is the closest we can get to an automatic Downward Löwenheim-Skolem Theorem for ω-automatic structures.

Proposition 1.5.2 ([85])   *Let $(D, \approx, \{R_i\}_{i \leq \omega})$ be an omega-automatic presentation of $\mathfrak{A}$ and let $\mathfrak{A}_{up}$ be its restriction to the ultimately periodic words of D. Then $\mathfrak{A}_{up}$ is a countable elementary substructure of $\mathfrak{A}$.*

Proof   Relying on the Tarski-Vaught criterion for elementary substructures we only need to show that for all first-order formulas $\varphi(\overline{x}, y)$ and elements $\overline{b}$ of $\mathfrak{A}_{\mathrm{up}}$

$$\mathfrak{A} \models \exists y \varphi(\overline{b}, y) \quad \Rightarrow \quad \mathfrak{A}_{\mathrm{up}} \models \exists y \varphi(\overline{b}, y) \ .$$

By Theorem 1.3.4 $\varphi(\overline{x}, y)$ defines an omega-regular relation and, similarly, since the parameters $\overline{b}$ are all ultimately periodic the set defined by $\varphi(\overline{b}, y)$ is omega-regular. Therefore, if it is non-empty, then it also contains an ultimately periodic word, which is precisely what we needed. ◁

An identical proposition, also independently noted by Khoussainov and Nies, holds for $\mathfrak{A} \in \omega\texttt{T-AutStr}$ with regular trees in place of ultimately periodic words.

Consider the natural, say, binary ω-automatic presentation of $(\mathbb{R}, +)$. Its restriction to the set of elements represented by ultimately periodic ω-words is isomorphic to the additive group of the rationals $(\mathbb{Q}, +)$. Tsankov [136] has shown that there is no automatic divisible torsion-free Abelian group (DTAG). Hence the theory of DTAGs is another example of a first-order theory having an uncountable ω-automatic model but no countable $(\omega$-$)$automatic models.

Automatic theorems
Kőnig's Lemma

Kőnig's Lemma says that an infinite finitely-branching tree has an infinite path. We split our discussion of automatic analogues along two lines, depending on whether the signature is that of partial order $(T, \preceq)$ or successor (T, S).

Theorem 1.5.3 ([91])   *If $\mathcal{T} = (T, \preceq)$ is an automatic copy of an infinite finitely-branching tree, then $\mathcal{T}$ has a regular infinite path. That is, there exists a regular set $P \subseteq T$ where P is an infinite path of $\mathcal{T}$.*

Proof   Define a set P as those elements x such that $\exists^\infty w[x \prec w]$ and for which every $y \prec x$ satisfies that

$$\forall z, z' \in S(y)[z \preceq x \Rightarrow z \leq_{llex} z'].$$

Then P is the length-lexicographically least infinite path of $\mathcal{T}$ (in the ordering induced by the finite strings presenting the tree).   ◁

However, using the 2-Ramsey quantifier we can do more.

Theorem 1.5.4 ([91])   *If $\mathcal{T} = (T, \preceq)$ is an automatic copy of a tree with countably many infinite paths, then every infinite path is regular.*

Proof   Denote by $E(\mathcal{T}) \subseteq T$ the set of elements of a tree $\mathcal{T}$ that are on infinite paths. It is definable in $\mathcal{T}$ using the 2-Ramsey quantifier, so Theorem 1.4.7 gives that $E(\mathcal{T})$ is regular. Then every isolated path of $\mathcal{T}$ is regular, since it is definable as $\{x \in E(\mathcal{T}) \mid p \preceq x\} \cup \{x \in E(\mathcal{T}) \mid x \prec p\}$, for suitable $p \in E(\mathcal{T})$. Replace $\mathcal{T}$ by its derivative $d(\mathcal{T})$, which is also automatically presentable. Since the CB-rank of $\mathcal{T}$ is finite [91] and $d^{\mathrm{CB}(\mathcal{T})}(\mathcal{T})$ is the empty tree, every infinite path is defined in this way. ◁

However, automatic successor trees behave more like computable trees:

Theorem 1.5.5 ([96])   *The problem of deciding, given automata presenting a successor tree (T, S), whether or not it has an infinite path, is Σ^1_1-complete.*

The proof consists of a reduction from the problem of whether a non-deterministic Turing machine visits a designated state infinitely often.

We compare with the computable case.[20] Fix the computable presentation of the full binary tree as consisting of the finite binary sequences with the immediate successor relation (so in fact the prefix relation is also computable). To stress this presentation, we refer to the tree as 2^ω. Similarly fix a natural computable presentation ω^ω of the ω-branching tree. A *computable subtree* of either of these trees is a computable prefix-closed subset.

(i) There is an infinite computable subtree of 2^ω with no computable infinite path.
(ii) There is a computable subtree of ω^ω with exactly one infinite path, and this path is not computable.
(iii) The set of indices of computable subtrees of the binary tree 2^ω with at least one infinite path is Π^0_2-complete.

[20] Thanks to Frank Stephan for discussions concerning this case.

(iv) The set of indices of computable subtrees of ω^ω with at least one infinite path Σ_1^1-complete.

Cantor's Theorems

One of Cantor's theorems says that every countable linear ordering embeds in the rational ordering $\mathbb{Q}$. The standard proof is easily seen to be effective given a computable presentation of $(\mathbb{Q}, <)$.

There are potentially a variety of automatic versions. The following proposition is the best known.

Proposition 1.5.6 [93] *Every automatic copy $\mathcal{M}$ of a linear order can be embedded into some automatic copy of $\mathbb{Q}$ by a function $f : \mathcal{M} \to \mathbb{Q}$ with the following properties:*

 (i) *The function f is continuous with respect to the order topology.*
(ii) *The graph of f is regular.*

It is not known whether there is a single automatic copy of $\mathbb{Q}$ that embeds, in the sense above, all automatic copies of all automatically presentable linear orders $\mathcal{M}$.

Cantor also proved that $\mathbb{Q}$ is homogeneous: For every two tuples $x_1 < \cdots < x_m$ and $y_1 < \cdots < y_m$ there is an automorphism $f : \mathbb{Q} \to \mathbb{Q}$ with $f(x_i) = y_i$ for $i \le m$. Again there might be a number of automatic variations. Call an automatic copy of $\mathbb{Q}$ *automatically homogeneous* if for every two tuples there is an automorphism as above that is also regular.

Proposition 1.5.7 [93] *There is an automatic copy of $\mathbb{Q}$ that is automatically homogeneous. There is an automatic copy of $\mathbb{Q}$ that is not automatically homogeneous.*

Scott ranks

Every countable structure $\mathcal{A}$ has a sentence of the infinitary logic $L_{\omega_1,\omega}$ (it allows, in addition to FO, countable disjuncts but still only finitely many free variables) that characterises $\mathcal{A}$ up to isomorphism. The *Scott rank* of $\mathcal{A}$ is the minimal quantifier rank amongst all such sentences.

Theorem 1.5.8 ([86])   *For every computable ordinal there is an automatic structure of Scott Rank at least α.*

The idea is to massage the configuration space of Turing machines presenting a computable structure (having Scott Rank α) to get an automatic structure of similar rank.

1.5.2 On the universal word-automatic structure

We conclude by highlighting some model-theoretic properties of the universal structure $\mathcal{S}_{[2]}$.

(i) $\mathcal{S}_{[2]}$ has infinite VC-dimension [15]. That is, there is a formula $\phi(x, z)$ that defines a family of sets of the form $\phi(-, z)^{\mathcal{S}_{[2]}}$ as one varies the parameter z, and this family fully shatters arbitrarily large finite sets.

(ii) $\mathcal{S}_{[2]}$ admits quantifier elimination (QE) in the expansion of all definable unary predicates and binary functions. In fact, no expansion with definable unary functions (and arbitrary predicates) admits QE [15].

Blumensath [20, p. 67] raised the question of whether there are nonstandard models of the theory of the universal structure $\mathcal{S}_{[2]}$ in S-AutStr. Here we sketch an argument resting on Theorem 1.4.26 that shows that there are no word-automatic non-standard models. This result was obtained in discussions with Bakhadyr Khoussainov.

Theorem 1.5.9 $\mathcal{S}_{[2]}$ *is the only word-automatic model of its theory.*

Proof Assume, for a contradiction, an automatic presentation of a nonstandard elementary extension of $\mathcal{S}_{[2]}$. By 'component' we mean a maximal set of elements connected by successor relations. Every elementary extension of $\mathcal{S}_{[2]}$ consists of the standard component isomorphic to $\mathcal{S}_{[2]}$ (containing the root), and any number of non-standard components, that are, as unlabelled graphs, all isomorphic to one-another. The non-standard components are distinguished by the infinite sequences of 0-1 successors ascending towards the root.

(0) The set of representatives of elements of each component is regular.

Indeed, the equivalence relation of belonging to the same component is FO + $\exists^{\infty}$-definable in the model (by saying that there is a common ancestor having finite distance from both elements), hence regular in the representation.

(1) There is a non-standard element below every standard node.

This follows from the fact that the formula

$$\forall x, x', y : \mathtt{el}(x, x') \wedge x \prec y \rightarrow \exists y' : \mathtt{el}(y, y') \wedge x' \prec y'$$

being true in $\mathcal{S}_{[2]}$ must also hold in every non-standard model.

Combining observation (0) and Theorem 1.4.26 we may assume that the presentation restricted to the standard component is the natural one having the identity as naming function. The binary ω-sequence naturally associated with an infinite branch of the standard component provides a representation of the set of nodes along that branch consistent with the assumed presentation of the model. Denote by Π the set of paths with a non-standard element below them.

(2) The set Π is ω-regular.

Indeed, a Büchi-automaton is built to guess a finite word representing a non-standard element and to check, using the automata of the assumed presentation, that it is a descendant of all finite prefixes of the input path. Given that our model is countable, hence so is Π, we have the following consequence of claim (2).

(3) Every path in Π is ultimately periodic with a period of bounded length.

To close the circle, consider for each $n \in \mathbb{N}$ the sentence

$$\forall x \exists y \, |y| > |x| \wedge 0^n 1 \preceq_{\texttt{prefix}} y \wedge (\forall z \prec_{\texttt{prefix}} y)[\mathrm{end}_1(z) \to z 0^n 1 \preceq_{\texttt{prefix}} y]$$

where $\mathrm{end}_1(z)$ is shorthand for saying that the last letter of z is 1. This sentence expresses that for every length $|x|$ there is a longer word y with as many initial prefixes in $(0^n 1)^*$ as possible. In particular this sentence holds for non-standard elements x. Consequently,

(4) for every $n \in \mathbb{N}$ there is an infinite branch of the standard component with label $(0^n 1)^\omega$ and having non-standard elements below it.

This contradicts observation (3).   ◁

Therefore, by Theorem 1.4.20, there are no countable ω-word automatic non-standard models either. Furthermore, using Theorem 1.4.29 in place of Theorem 1.4.26 in the argument shows there are no non-standard finite-tree automatic models of $\mathcal{S}_{[2]}$. To prove that there are no uncountable ω-word automatic non-standard models of $\mathcal{S}_{[2]}$ one tightens (4) and exploits that all automatic presentations of non-standard components are equivalent.

References

[1] K. Aehlig, J. G. de Miranda, and C.-H. L. Ong. Safety is not a restriction at level 2 for string languages. In *FoSSaCS*, pages 490–504, 2005.

[2] J.-P. Allouche and J. Shallit. *Automatic Sequences, Theory, Applications, Generalizations*. Cambridge University Press, 2003.

[3] J.-P. Allouche and J. O. Shallit. The Ubiquitous Prouhet-Thue-Morse Sequence. In C. Ding, T. Helleseth, and H. Niederreiter, editors, *Sequences and Their Applications: Proceedings of SETA '98*, pages 1–16. Springer-Verlag, 1999.

[4] V. Bárány. A hierarchy of automatic ω-words having a decidable MSO theory. Journées Montoises '06, Rennes, 2006.

[5] V. Bárány. Invariants of automatic presentations and semi-synchronous transductions. In *STACS '06*, volume 3884 of *LNCS*, pages 289–300, 2006.

[6] V. Bárány. *Automatic Presentations of Infinite Structures*. Phd thesis, RWTH Aachen University, 2007.

[7] V. Bárány. A hierarchy of automatic ω-words having a decidable MSO theory. *R.A.I.R.O. Theoretical Informatics and Applications*, 42:417–450, 2008.

[8] V. Bárány. Semi-synchronous transductions. *Acta Informatica*, 46(1):29–42, 2009.

[9] V. Bárány, Ł. Kaiser, and A. Rabinovich. Eliminating cardinality quantifiers from MLO. Manuscript, 2007.

[10] V. Bárány, Ch. Löding, and O. Serre. Regularity problems for visibly pushdown languages. In *STACS '06*, volume 3884 of *LNCS*, pages 420–431, 2006.

[11] K. Barthelmann. On equational simple graphs. Tech. Rep. 9, Universität Mainz, Institute für Informatik, 1997.

[12] K. Barthelmann. When can an equational simple graph be generated by hyperedge replacement? In *MFCS*, pages 543–552, 1998.

[13] M.-P. Béal and D. Perrin. Symbolic Dynamics and Finite Automata. In A. Salomaa and G. Rosenberg, editors, *Handbook of Formal Languages, Vol. 2*, pages 463–503. Springer Verlag, 1997.

[14] M. Benedikt and L. Libkin. Tree extension algebras: logics, automata, and query languages. In *Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science* (LICS), pages 203–212, 2002.

[15] M. Benedikt, L. Libkin, Th. Schwentick, and L. Segoufin. A model-theoretic approach to regular string relations. In Joseph Halpern, editor, *LICS 2001*, pages 431–440. IEEE Computer Society, June 2001.

[16] M. Benedikt, L. Libkin, Th. Schwentick, and L. Segoufin. Definable relations and first-order query languages over strings. *J. ACM*, 50(5):694–751, 2003.

[17] D. Berwanger and A. Blumensath. The monadic theory of tree-like structures. In E. Grädel, W. Thomas, and T. Wilke, editors, *Automata, Logics, and Infinite Games*, number 2500 in LNCS, chapter 16, pages 285–301. Springer Verlag, 2002.

[18] A. Bès. Undecidable extensions of Büchi arithmetic and Cobham-Semënov theorem. *Journal of Symbolic Logic*, 62(4):1280–1296, 1997.

[19] A. Bés. An Extension of the Cobham-Semënov Theorem. *J. of Symb. Logic*, 65(1):201–211, 2000.

[20] A. Blumensath. Automatic Structures. Diploma thesis, RWTH-Aachen, 1999.

[21] A. Blumensath. Prefix-Recognisable Graphs and Monadic Second-Order Logic. Technical report AIB-2001-06, RWTH Aachen, 2001.

[22] A. Blumensath. Axiomatising Tree-interpretable Structures. In *STACS*, volume 2285 of *LNCS*, pages 596–607. Springer-Verlag, 2002.

[23] A. Blumensath, Th. Colcombet, and Ch. Löding. Logical theories and compatible operations. In J. Flum, E. Grädel, and T. Wilke, editors, *Logic and Automata: History and Perspectives*, Texts in Logic and Games, pages 73–106. Amsterdam University Press, 2007.

[24] A. Blumensath and E. Grädel. Automatic structures. In *LICS 2000*, pages 51–62. IEEE Computer Society, 2000.

[25] A. Blumensath and E. Grädel. Finite presentations of infinite structures: Automata and interpretations. *Theory of Comp. Sys.*, 37:641 – 674, 2004.

[26] V. Bruyère, G. Hansel, Ch. Michaux, and R. Villemaire. Logic and p-recognizable sets of integers. *Bull. Belg. Math. Soc.*, 1:191 – 238, 1994.

[27] J. R. Büchi. Weak second-order arithmetic and finite automata. *Zeit. Math. Logih Grund. Math.*, 6:66–92, 1960.

[28] A. J. Cain, E. F. Robertson, and N. Ruskuc. Subsemigroups of groups: presentations, malcev presentations, and automatic structures. *Journal of Group Theory*, 9(3):397–426, 2006.

[29] C. M. Campbell, E. F. Robertson, N. Ruskuc, and R. M. Thomas. Automatic semigroups. *Theor. Comput. Sci.*, 250(1-2):365–391, (2001).

[30] C. M. Campbell, E. F. Robertson, N. Ruskuc, and R. M. Thomas. Automatic completely-simple semigroups. *Acta Math. Hungar.*, 96:201–215, 2002.

[31] J.W. Cannon, D.B.A. Epstein, D.F. Holt, S.V.F. Levy, M.S. Paterson, and W.P. Thurston. *Word processing in groups*. Jones and Barlett Publ., Boston, MA, 1992.

[32] A. Carayol. Regular sets of higher-order pushdown stacks. In *Proceedings of Mathematical Foundations of Computer Science (MFCS 2005)*, volume 3618 of *LNCS*, pages 168–179, 2005.

[33] A. Carayol and Th. Colcombet. On equivalent representations of infinite structures. In *ICALP*, volume 2719 of *LNCS*, pages 599–610. Springer, 2003.

[34] A. Carayol and A. Meyer. Linearly bounded infinite graphs. In *MFCS*, volume 3618 of *Lecture Notes in Computer Science*, pages 180–191. Springer, 2005.

[35] A. Carayol and A. Meyer. Context-Sensitive Languages, Rational Graphs and Determinism. *Logical Methods in Computer Science*, 2(2), 2006.

[36] A. Carayol and C. Morvan. On rational trees. In Z. Ésik, editor, *CSL 06*, volume 4207 of *LNCS*, pages 225–239, 2006.

[37] A. Carayol and S. Wöhrle. The Caucal hierarchy of infinite graphs in terms of logic and higher-order pushdown automata. In *FSTTCS*, volume 2914 of *LNCS*, pages 112–123. Springer, 2003.

[38] O. Carton and W. Thomas. The monadic theory of morphic infinite words and generalizations. *Information and Computation*, 176(1):51–65, 2002.

[39] A. Caryol and Ch. Löding. MSO on the Infinite Binary Tree: Choice and Order. In *CSL*, volume 4646 of *LNCS*, pages 161–176, 2007.

[40] D. Caucal. Monadic theory of term rewritings. In *LICS*, pages 266–273. IEEE Computer Society, 1992.

[41] D. Caucal. On the regular structure of prefix rewriting. *Theor. Comput. Sci.*, 106(1):61–86, 1992.

[42] D. Caucal. On infinite transition graphs having a decidable monadic theory. In *ICALP'96*, volume 1099 of *LNCS*, pages 194–205, 1996.

[43] D. Caucal. On infinite terms having a decidable monadic theory. In *MFCS*, pages 165–176, 2002.

[44] D. Caucal. Deterministic graph grammars. In J. Flum, E. Grädel, and T. Wilke, editors, *Logic and Automata: History and Perspectives*, Texts in Logic and Games, pages 169–250. Amsterdam University Press, 2007.

[45] D. Cenzer and J.B. Remmel. Complexity-theoretic model theory and algebra. In *Handbook of Recursive Mathematics, Vol. 1*, volume 138 of *Studies in Logic and the Foundations of Mathematics*, pages 381–513. North-Holland, Amsterdam, 1998.

[46] Ch. Choffrut. A short introduction to automatic group theory, 2002.

[47] T. Colcombet and C. Löding. Transforming structures by set interpretations. *Logical Methods in Computer Science*, 3(2), 2007.

[48] Th. Colcombet. On families of graphs having a decidable first order theory with reachability. In *ICALP*, volume 2380 of *LNCS*, pages 98–109. Springer, 2002.

[49] Th. Colcombet. Equational presentations of tree-automatic structures. In Workshop on Automata, Structures and Logic, Auckland, NZ, 2004.

[50] Th. Colcombet. *Propriétés et représentation de structures infinies*. Thèse de doctorat, Université Rennes I, 2004.

[51] Th. Colcombet. A combinatorial theorem for trees. In *ICALP*, volume 4596 of *LNCS*, pages 901–912. Springer, 2007.

[52] H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree Automata Techniques and Applications. In preparation, draft available online at http://www.grappa.univ-lille3.fr/tata/.

[53] B. Courcelle. *Graph algebras and monadic second-order logic*. Cambridge University Press, in writing...

[54] B. Courcelle. The definability of equational graphs in monadic second-order logic. In *ICALP*, volume 372 of *LNCS*, pages 207–221. Springer, 1989.

[55] B. Courcelle. Graph rewriting: An algebraic and logic approach. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics*, pages 193–242. Elsevier and MIT Press, 1990.

[56] B. Courcelle. Recursive applicative program schemes. In J. v.d. Leeuwen, editor, *Handbook of Theoretical Computer Science, Vol. B*, pages 459–492. Elsevier and MIT Press, 1990.

[57] B. Courcelle. The monadic second-order logic of graphs ix: Machines and their behaviours. *Theoretical Computer Science*, 151(1):125–162, 1995.

[58] B. Courcelle. Finite model theory, universal algebra and graph grammars. In *LFCS '97, Proceedings of the 4th International Symposium on Logical Foundations of Computer Science*, pages 53–55, London, UK, 1997. Springer-Verlag.

[59] B. Courcelle. The Expression of Graph Properties and Graph Transformations in Monadic Second-Order Logic. In G. Rozenberg, editor, *Handbook of graph grammars and computing by graph transformations, vol. 1: Foundations*, pages 313–400. World Scientific, New-Jersey, London, 1997.

[60] B. Courcelle and J. A. Makowsky. Fusion in Relational Structures and the Verification of Monadic Second-Order Properties. *Mathematical Structures in Computer Science*, 12(2):203–235, 2002.

[61] B. Courcelle and I. Walukiewicz. Monadic second-order logic, graph coverings and unfoldings of transition systems. *Annals of Pure and Applied Logic*, 92:35–62, 1998.

[62] W. Damm. The IO- and OI hierarchies. *Theoretical Computer Science*, 20(2):95–208, 1982.

[63] C. Delhommé. Automaticité des ordinaux et des graphes homogènes. *Comptes Rendus Mathematique*, 339(1):5–10, 2004.

[64] M.J. Dunwoody. The accessibility of finitely presented groups. *Inventiones Mathematicae*, 81(3):449–457, 1985.

[65] S. Eilenberg, C.C. Elgot, and J.C. Shepherdson. Sets recognised by n–tape automata. *Journal of Algebra*, 13(4):447–464, 1969.

[66] C. C. Elgot and J. E. Mezei. On relations defined by generalized finite automata. *IBM J. Research and Development*, 9:47 – 68, 1965.

[67] C. C. Elgot and M. O. Rabin. Decidability and undecidability of extensions of second (first) order theory of (generalized) successor. *Journal of Symbolic Logic*, 31(2):169–181, 1966.

[68] C.C. Elgot. Decision problems of finite automata design and related arithmetics. *Trans. Amer. Math. Soc.*, 98:21–51, 1961.

[69] J. Engelfriet. Context-free graph grammars. In *Handbook of formal languages, vol. III*, pages 125–213. Springer-Verlag New York, Inc., New York, NY, USA, 1997.

[70] Y.L. Ershov, S.S. Goncharov, A. Nerode, and J.B. Remmel, editors. *Handbook of Recursive Mathematics, Vol. 1*, volume 138 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam, 1998.

[71] B. Farb. Automatic Groups: A Guided Tour. *L'Enseignment Math.*, 38:291–313, 1992.

[72] S. Fratani. *Automates à Piles de Piles ... de Piles*. Thèse de doctorat, Université Bordeaux 1, 2005.

[73] S. Fratani. Regular sets over tree structures. Rapport Interne 1358-05, LaBRI, Université Paris 7, 2005.

[74] S. Fratani. The theory of successor extended by severals predicates. Journées Montoises '06, Rennes, 2006.

[75] S. Fratani and G. Sénizergues. Iterated pushdown automata and sequences of rational numbers. *Ann. Pure Appl. Logic*, 141(3):363–411, 2006.

[76] Ch. Frougny. Numeration systems. In M. Lothaire, editor, *Algebraic Combinatorics on Words*. Cambridge University Press, 2002.

[77] E. Grädel. Simple interpretations among complicated theories. *Information Processing Letters*, 35:235–238, 1990.

[78] E. Grädel, P. G. Kolaitis, L. Libkin, M. Marx, J. Spencer, M. Vardi, Y. Venema, and S. Weinstein. *Finite Model Theory and Its Applications*. Springer-Verlag, 2007.

[79] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games*, volume 2500 of *LNCS*. Springer-Verlag, 2002.

[80] A. Habel. *Hyperedge Replacement: Grammars and Languages*, volume 643 of *Lecture Notes in Computer Science*. Springer, 1992.

[81] M. Hague, A.S. Murawski, C.-H. L. Ong, and O. Serre. Collapsible pushdown automata and recursion schemes. In *LICS'08*. IEEE Computer Society, 2008.

[82] G. Hjorth, B. Khoussainov, A. Montalbán, and A. Nies. From automatic structures to Borel structures. In *23rd Symposium on Logic in Computer Science (LICS)*, 2008.

[83] M. Hoffmann, D. Kuske, F. Otto, and R. M. Thomas. Some relatives of automatic and hyperbolic groups, 2002.

[84] M. Hoffmann and R. M. Thomas. Notions of automaticity in semigroups. *Semigroup Forum*, 66:337–367., (2003).

[85] Ł. Kaiser, S. Rubin, and V. Bárány. Cardinality and counting quantifiers on ω-automatic structures. In *STACS '08*, volume 08001 of *Dagstuhl Seminar Proceedings*, pages 385–396. Internationales Begegnungs- und

Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2008.

[86] B. Khoussainov and M. Minnes. Model theoretic complexity of automatic structures. *Annals of Pure and Applied Logic*, To appear, 2008.

[87] B. Khoussainov and A. Nerode. Automatic presentations of structures. In *LCC '94*, volume 960 of *LNCS*, pages 367–392. Springer-Verlag, 1995.

[88] B. Khoussainov, A. Nies, S. Rubin, and F. Stephan. Automatic structures: Richness and limitations. In *LICS'04*, pages 44–53, 2004.

[89] B. Khoussainov and S. Rubin. Graphs with automatic presentations over a unary alphabet. *Journal of Automata, Languages and Combinatorics*, 6(4):467–480, 2001.

[90] B. Khoussainov, S. Rubin, and F. Stephan. Definability and regularity in automatic structures. In *STACS '04*, volume 2996 of *LNCS*, pages 440–451, 2004.

[91] B. Khoussainov, S. Rubin, and F. Stephan. Automatic linear orders and trees. *ACM Transactions on Computational Logic*, 6(4):675–700, 2005.

[92] T. Knapik, D. Niwinski, and P. Urzyczyn. Higher-order pushdown trees are easy. In *FoSSaCS'02*, volume 2303 of *LNCS*, pages 205–222, 2002.

[93] D. Kuske. Is cantor's theorem automatic? In *LPAR*, volume 2850 of *LNCS*, pages 332–345. Springer, 2003.

[94] D. Kuske and M. Lohrey. First-order and counting theories of ω-automatic structures. In *FoSSaCS*, pages 322–336, 2006.

[95] D. Kuske and M. Lohrey. Automatic structures of bounded degree revisited. arXiv:0810.4998, 2008.

[96] D. Kuske and M. Lohrey. Hamiltonicity of automatic graphs. In *FIP TCS 2008*, 2008.

[97] H. Lauchli and Ch. Savioz. Monadic Second Order Definable Relations on the Binary Tree. *J. of Symbolic Logic*, 52(1):219–226, 1987.

[98] S. Lifsches and S. Shelah. Uniformization and skolem functions in the class of trees. *Journal of Symbolic Logic*, 63:103–127, 1998.

[99] Ch. Löding. *Infinite Graphs Generated by Tree Rewriting*. Doctoral thesis, RWTH Aachen, 2003.

[100] Christof Löding. Reachability problems on regular ground tree rewriting graphs. *Theor. Comp. Sys.*, 39(2):347–383, 2006.

[101] M. Lohrey. Automatic structures of bounded degree. In *LPAR*, volume 2850 of *LNCS*, pages 346–360. Springer, 2003.

[102] M. Lohrey. Decidability and complexity in automatic monoids. In *Developments in Language Theory*, pages 308–320, 2004.

[103] A. Meyer. Traces of term-automatic graphs. *R.A.I.R.O. Theoretical Informatics and Applications*, 42, 2008.

[104] C. Michaux and F. Point. Les ensembles k-reconnaissables sont définissables dans $\langle \mathbf{N}, +, V_k \rangle$. *C. R. Acad. Sci. Paris Sér. I Math.*, 303(19):939–942, 1986.

[105] Ch. Morvan. *Les graphes rationnels*. Thèse de doctorat, Université de Rennes 1, Novembre 2001.

[106] Ch. Morvan. Classes of rational graphs. Journées Montoises '06, Rennes, 2006.

[107] Ch. Morvan and Ch. Rispal. Families of automata characterizing context-sensitive languages. *Acta Informatica*, 41(4-5):293–314, 2005.

[108] Ch. Morvan and C. Stirling. Rational graphs trace context-sensitive languages. In A. Pultr and J. Sgall, editors, *MFCS 01*, volume 2136 of *LNCS*, pages 548–559, 2001.

[109] A. A. Muchnik. The definable criterion for definability in Presburger arithmetic and its applications. *Theor. Comput. Sci.*, 290(3):1433–1444, 2003.

[110] D. E. Muller and P. E. Schupp. Context-free languages, groups, the theory of ends, second-order logic, tiling problems, cellular automata, and vector addition systems. *Bull. Amer. Math. Soc.*, 4(3):331–334, 1981.

[111] D. E. Muller and P. E. Schupp. Groups, the theory of ends, and context-free languages. *J. Comput. Syst. Sci.*, 26(3):295–310, 1983.

[112] D. E. Muller and P. E. Schupp. The theory of ends, pushdown automata, and second-order logic. *Theor. Comput. Sci.*, 37:51–75, 1985.

[113] A. A. Nabebin. Expressibility in a restricted second-order arithmetic. *Siberian Mathematical Journal*, 18(4):588–593, 1977.

[114] A. Nies. Describing groups. *Bulletin of Symbolic Logic*, 13(3):305–339, 2007.

[115] D. Niwiński. On the cardinality of sets of infinite trees recognizable by finite automata. In *Proceedings of the 16th International Symposium on Mathematical Foundations of Computer Science, MFCS'91*, volume 520, pages 367–376. Springer, 1991.

[116] G. P. Oliver and R. M. Thomas. Finitely generated groups with automatic presentations. In *STACS 2005*, volume 3404 of *LNCS*, pages 693–704. Springer, 2005.

[117] C.-H. L. Ong. On model-checking trees generated by higher-order recursion schemes. In *LICS*, pages 81–90. IEEE Computer Society, 2006.

[118] J.-J. Pansiot. On various classes of infinite words obtained by iterated mappings. In *Automata on Infinite Words*, pages 188–197, 1984.

[119] A. Rabinovich. On decidability of monadic logic of order over the naturals extended by monadic predicates. Unpublished note, 2005.

[120] A. Rabinovich and W. Thomas. Decidable theories of the ordering of natural numbers with unary predicates. Submitted, 2006.

[121] M. Rigo. Numeration systems on a regular language: Arithmetic operations, recognizability and formal power series. *Theoretical Computer Science*, 269:469, 2001.

[122] M. Rigo and A. Maes. More on generalized automatic sequences. *J. of Automata, Languages and Combinatorics*, 7(3):351–376, 2002.

[123] Ch. Rispal. The synchronized graphs trace the context-sensitive languages. *Electronic Notes in Theor. Comp. Sci.*, 68(6), 2002.

[124] S. Rubin. *Automatic Structures*. Phd thesis, University of Auckland, NZ, 2004.

[125] S. Rubin. Automata presenting structures: A survey of the finite-string case. *Bulletin of Symbolic Logic*, 14(2):169–209, 2008.

[126] A. L. Semenov. Decidability of monadic theories. In *Mathematical Foundations of Computer Science, Prague, 1984*, volume 176 of *LNCS*, page 162?175. Springer, Berlin, 1984.

[127] G. Sénizergues. Semi-groups acting on context-free graphs. In *ICALP '96: Proceedings of the 23rd International Colloquium on Automata, Languages and Programming*, pages 206–218, London, UK, 1996. Springer-Verlag.

[128] G. Sénizergues. The bisimulation problem for equational graphs of finite out-degree. *SIAM J. Comput.*, 34(5):1025–1106, 2005.

[129] P. V. Silva and B. Steinberg. A geometric characterization of automatic monoids. *The Quarterly Journal of Mathematics*, 55:333–356, 2004.

[130] J. Su and S. Grumbach. Finitely representable databases (extended abstract. In *In Proc. 13th ACM Symp. on Principles of Database Systems*, 1994.

[131] A. Szilard, Sh. Yu, K. Zhang, and J. Shallit. Characterizing regular languages with polynomial densities. In *MFCS*, pages 494–503, 1992.

[132] W. Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics*, pages 133–192. Elsevier and MIT Press, 1990.

[133] W. Thomas. Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages, volume III*, pages 389–455. Springer, New York, 1997.

[134] W. Thomas. Constructing Infinite Graphs with a Decidable MSO-Theory. In *MFCS*, volume 2747 of *LNCS*, pages 113–124, 2003.

[135] B.A. Trahtenbrot. Finite automata and the logic of one-place predicates. Russian. *Siberian Mathematical Journal*, 3:103–131, 1962. English translation: American Mathematical Society Translations, Series 2, 59 (1966), 23-55.

[136] T. Tsankov. The additive group of the rationals is not automatic. manuscript, 2009.

[137] R. Villemaire. The theory of $\langle \mathbf{N}, +, V_k, V_l \rangle$ is undecidable. *Theoretical Computer Science*, 106:337–349, 1992.

[138] I. Walukiewicz. Monadic second-order logic on tree-like structures. *Theoretical Computer Science*, 275:311–346, 2002.

[139] S. Wöhrle and W. Thomas. Model checking synchronized products of infinite transition systems. In *LICS '04*, pages 2–11, Washington, DC, USA, 2004. IEEE Computer Society.