1

Automata-based presentations of infinite structures

VINCE BÁRÁNY[1], 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 natural 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

[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

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 $\phi(X_1, \ldots, X_j, x_1, \ldots, x_k)$ interpreted in $\mathfrak{B}$ *defines* the set $\phi^{\mathfrak{B}} := \{(B_1, \ldots, B_j, b_1, \ldots, b_k) \mid B_i \subset B, b_i \in B, \mathfrak{B} \models \phi(B_1, \ldots, B_j, b_1, \ldots, b_k)\}$. 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

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

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 $[r] := \{0, \dots, r-1\}$, 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 $\Phi_{R_i}(\overline{x}_1, \dots, \overline{x}_{r_i})$ 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 $\psi(x_1, \dots, x_k)$ in the signature of $\mathfrak{B}$ and all k-tuples $\overline{b}$ of elements of*

$\mathfrak{B}$ *it holds that*

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

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 $\mathfrak{B} \leq^{\mathcal{I}}_{\text{set}} \mathfrak{A}$ or $\mathfrak{B} \leq^{\mathcal{I}}_{\text{fset}} \mathfrak{A}$. 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 $(\mathbb{N}, +) \leq^{\mathcal{I}}_{\text{fset}} (\mathbb{N}, 0, \texttt{suc})$ 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

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

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

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

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].

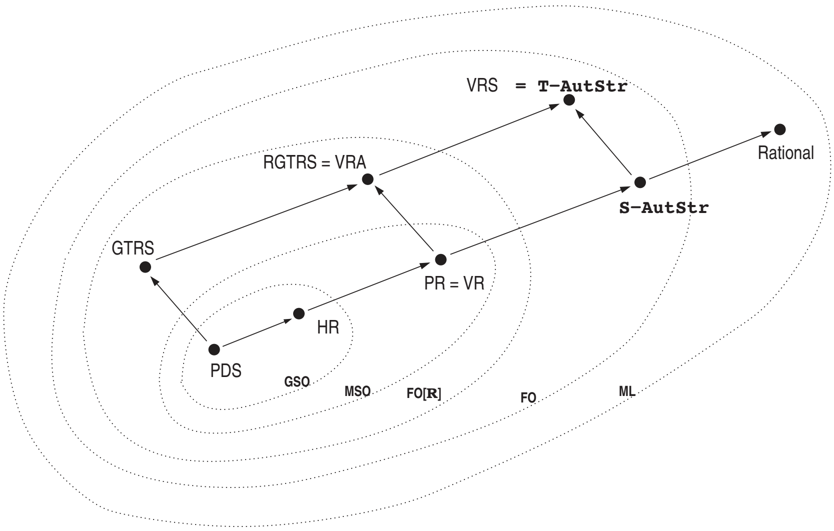


Figure 1.1 Relationship of graph classes and logical decidability boundaries.

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 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].

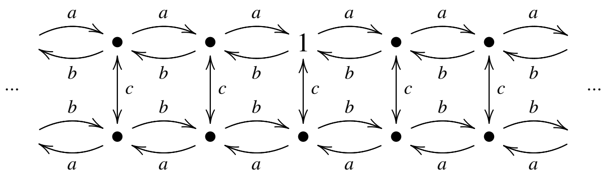
A *context-free group* is a finitely generated group G such that, for some set S of semigroup generators of G, the set of words $w \in S^*$ representing the identity element of G forms a context-free language. This is independent of the choice of S. Moreover, a group is context-free if and only if its 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]

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.

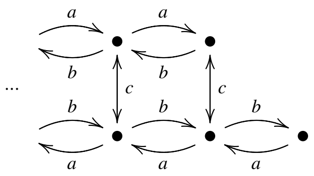
Example 1.2.1 Consider the group G given by the finite presentation $\langle a, b, c \mid ab, cc, acac, bcbc \rangle$. The Cayley graph $\Gamma(G, S)$ of G with respect

[4] connectedness is taken with respect to the underlying undirected graph.
[5] Originally [111] proved this under the assumption of *accessibility*, a notion related to group decompositions introduced by Wall who conjectured that all 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.

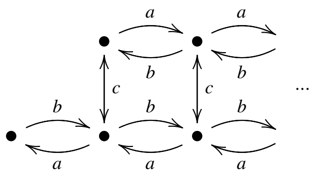
to the set of semigroup generators $S = \{a, b, c\}$ is depicted below.



Notice that $\Gamma(G, S)$ has two ends, for any n-neighbourhood of the identity with $n > 1$. These are



and

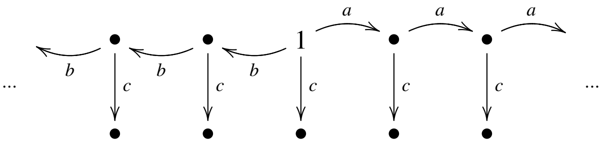


A word $w \in \{a, b, c\}^*$ represents the identity of G if, and only if, w has an even number of c's and the number of a's equals the number of b's. We present a pushdown automaton $\mathcal{A}$ which recognises this set of words and, moreover, has a configuration graph that is isomorphic to $\Gamma(G, S)$. 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:

internal:	$1\,\theta$	$\xrightarrow{c}$	$c\,\theta$	
internal:	$c\,\theta$	$\xrightarrow{c}$	$1\,\theta$	
push:	$q\,\sigma\theta$	$\xrightarrow{\sigma}$	$q\,\sigma\sigma\theta$	for $q = 1, c$ and $\sigma = a, b$
push:	$q\,\bot$	$\xrightarrow{\sigma}$	$q\,\sigma\bot$	for $q = 1, c$ and $\sigma = a, b$
pop:	$q\,\sigma\theta$	$\xrightarrow{\bar{\sigma}}$	$q\,\theta$	for $q = 1, c$ and $\{\sigma, \bar{\sigma}\} = \{a, b\}$

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 $R^{(\bar{i})} = \{(u_{i_1}, \ldots, u_{i_m}) \mid (u_1, \ldots, u_m) \in R\} \in \mathsf{PR}$;
- 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 $<_{\text{lex}}$ 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 $[\![t(\overline{x})]\!]$ 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 $\{[\![t]\!] \mid t \in L\}$. 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