On Logics, Tilings, and Automata

Wolfgang Thomas*)

Institut für Informatik und Praktische Mathematik
Christian-Albrechts-Universität Kiel
Olshausenstr. 40, W 2300 Kiel 1
e-mail: wt@informatik.uni-kiel.dbp.de

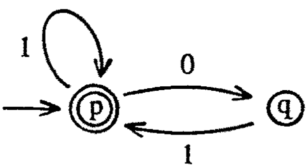
Abstract

We relate the logical and the automata theoretic approach to define sets of words, trees, and graphs. For this purpose a notion of "graph acceptor" is introduced which can specify monadic second-order properties and allows to treat known types of finite automata in a common framework. In the final part of the paper, we discuss infinite graphs that have a decidable monadic second-order theory.

1. Introduction

Many formalisms have been developed in theoretical computer science for specifying properties of words, trees, or graphs. The purpose of the present paper is to discuss and relate two of these approaches: the definition in certain logical systems and the recognition by finite automata.

Let us recall, by an example, a classical theorem of Büchi [Bü60] and Elgot [El61] which establishes a bridge between logic and automata in the domain of finite words: Consider the property of (nonempty) words over the alphabet $A = \{0,1\}$ to contain no segment 00 and to end with 1. It can be specified by the finite automaton with the following state graph:



For a logical description we identify a word $a_1...a_n$ with the relational structure $(\{1,...,n\},S,P_0,P_1)$ where S is the successor relation on $\{1,...,n\}$, $P_0 = \{i \mid a_i = 0\}$, and $P_1 = \{i \mid a_i = 1\}$. This word model can be viewed as a vertex labelled graph. A sentence φ in the

---

*) Research supported by EBRA Working Group 3166 "Algebraic and Syntactic Methods in Computer Science (ASMICS)"

corresponding first-order language (with relation symbols S, P_0, P_1 besides equality) defines a set $L(\varphi) \subseteq \{0,1\}^+$, containing the words whose associated graph satisfies φ. For the example language above one may choose as defining sentence

$$\neg\exists x\exists y(Sxy \wedge P_0x \wedge P_0y) \;\wedge\; \exists x(\neg\exists ySxy \wedge P_1x).$$

In general, not only the first-order variables x,y,z,... but also variables X,Y,Z,... for sets of positions (and corresponding atomic formulas Xx, Xy,...) are admitted, i.e. we allow monadic second-order logic over the considered signature. The characterization result due to Büchi and Elgot states that a set $L \subseteq \{0,1\}^+$ is recognized by a finite automaton iff it is monadic second-order definable in this signature. Subsequent work of Büchi [Bü61], Rabin [Ra69], and others showed that similar equivalences hold for sets of infinite words, sets of finite trees, and sets of infinite trees (see [Th90] for a survey).

These equivalence results are remarkable for several reasons. Originally, they served to show decidability of interesting logical theories. (We return to this aspect in Section 5 of the paper.) Perhaps more important, they connect formalisms of very different nature: logical formulas which are built up inductively and have an inductively defined semantics, and "unstructured" automata which are not easily decomposed into meaningful constituents. In another view, one may regard automata (transition systems) as "programs" and monadic second-order logic as a specification formalism for their behaviour. The equivalence theorems state that there is a perfect match between specifications and programs, and the proofs yield transformations in both directions. Many applications and refinements have been obtained in the verification of finite state programs (now an own field of research), often referring to systems of temporal logic instead of classical quantifier logic.

The essence of the transformations from logic to automata is the reduction of a global description of words or trees (using quantifiers that range over the whole set of positions) to a description which refers only to local checking of consecutive letters (plus finite memory and acceptance condition).

It is natural to ask for such a connection between monadic second-order logic and finite state recognizability in a more general context, in particular for classes of graphs. A next step starting from trees is the class of directed acyclic graphs (or the induced partial orders), which is also of special interest in the semantics of concurrency.

In a series of fundamental papers, Courcelle has investigated the relation between monadic second-order logic and an algebraic notion of recognizability for graphs (see [Co90] for a survey). He introduces an algebra of graphs, using three basic operations which suffice for constructing arbitrary graphs: disjoint union, fusion of vertices, and renaming of vertices. To handle the latter, the considered graphs have distinguished vertices. A set of graphs is called recognizable if it is the inverse homomorphic image of a many-sorted graph algebra which is finite in each sort. Since infinitely many sorts are admitted (and necessary for building up natural graph sets like the set of grids), this recognizability has an infinitary feature; it is strictly more powerful in expressiveness than monadic second-order logic.

In the present paper we pursue a complementary view: We stay inside monadic second-order logic, and try to approximate its expressive power "from below" using a notion of recognition by finite state acceptors. We introduce an acceptor model which (1) generalizes known models of automata on words, trees, and directed acyclic graphs, (2) allows only to specify monadic second-order definable sets of graphs, (3) captures first-order logic when only single state acceptors are considered. The idea of "local tests" is realized in the form of "tiling by transitions"; this extends the notion of tiling as known from the literature on domino games (e.g. [LP81,Ha86]) and from work of Muller and Schupp [MS85] on the monadic theory of context-free graphs.

In the final part of the paper we return to the original motivation in reducing logical formulas to finite automata: decidability of monadic second-order theories. We sketch some recent work which aims at extending Rabin's Tree Theorem (decidability of the monadic second-order theory of the unvalued binary tree) to more complex structures, especially to certain infinite graphs.

The present treatment is a short summary, and several relevant aspects are not covered. For example, we do not discuss here the relation to graph grammars or to the theory of picture languages.

2. Graph recognizability by tilings

We consider directed graphs whose vertices and edges are labelled with symbols of finite alphabets A and B, respectively. Formally, these graphs are structures $G = (V,E,\beta,\alpha)$ where V is a nonempty and finite set, $E \subseteq V\times V$, and $\beta:E \to B$ and $\alpha:V \to A$ are the valuations. It will be convenient to represent these graphs as relational structures, namely in the form $(V,(E_b)_{b\in B},(P_a)_{a\in A})$ where the E_b are pairwise disjoint binary relations over V and the P_a are unary predicates which form a partition of V. The corresponding first-order or monadic second-order language has symbols for each of these predicates E_b and P_a. As in model theory we distinguish graphs only up to isomorphism.

For specifying graph properties by finite acceptors we start with the idea of "local tests" by transitions: a transition associates states with the vertices of a "local neighbourhood" in a graph, given by at least one vertex together with its adjacent vertices. If the admitted graph acceptors are finite, they should involve only finitely many transitions each of which is a finite object. Thus we consider only graphs whose degree is uniformly bounded by some constant k. If there is no bound to the degree of the vertices (number of adjacent vertices), one cannot in a direct way work with checks of local neighbourhoods by a finite recognition device.

Let $DG_k(A,B)$ be the class of finite directed graphs $(V,(E_b)_{b\in B},(P_a)_{a\in A})$ as above, where for each vertex x there are at most k vertices y with $(x,y) \in E_b$ or $(y,x) \in E_b$ for some $b\in B$. We speak of graphs of degree k.

2.1 Examples. (a) Words over an alphabet A: As in the introduction, a nonempty word $w = a_1...a_n$ can be represented by the graph $(\{1,...,n\},S,(P_a)_{a\in A})$ where S is the successor relation on $\{1,...n\}$ and $P_a = \{i \mid a_i = a\}$.

(b) Trees over an alphabet A: A k-ary tree whose vertices are valued in A and where the successors of vertices are ordered, is represented in the form $(V,(E_i)_{i\in\{1,...,k\}}, (P_a)_{a\in A})$ where $(x,y) \in E_i$ iff the i-th successor of x is y.

(c) Grids: These are graphs of the form $G_{m,n} = (\{1,...,m\}\times\{1,...,n\},E_1,E_2)$ where the edge sets E_1 and E_2 are given by

$$((x,y),(x+1,y)) \in E_1 \text{ for } 1\le x<m,\ 1\le y\le n, \quad ((x,y),(x,y+1)) \in E_2 \text{ for } 1\le x\le m,\ 1\le y<n$$

and where we assume a trivial vertex valuation.

We now turn to recognizability by graph acceptors.

2.2 Definition. A finite graph acceptor over $DG_k(A,B)$ is a triple $\mathcal{A} = (Q,\Delta,C)$ where Q is a finite set (of "states"), Δ is a finite set of connected graphs in $DG_k((Q\times A)\cup Q, B)$, called the set of transitions (or "tiles"), and C, called constraint, is a boolean combination of conditions of the form "there are $\ge$ n copies of transition τ" (where $\tau \in \Delta$).

Roughly speaking, the graph acceptor (Q,Δ,C) accepts a graph $G = (V,E,\beta,\alpha)$ if G can be "tiled coherently" by transitions from Δ obeying the constraint C; here "coherence" means that the A- and B-values of the transitions agree with the valuation of the underlying graph G and, concerning the Q-values, by overlapping transitions only one state is associated with each vertex of G. In other words, the tiling should define some "run" $\rho:V \to Q$.

Fix $G = (V,E,\beta,\alpha)$ and let $\rho:V \to Q$. Define the corresponding extended vertex valuation $\rho\times\alpha:V \to Q\times A$ by $\rho\times\alpha(x) = (\rho(x),\alpha(x))$. We denote by G_ρ the graph $(V,E,\beta,\rho\times\alpha)$. In order to describe the mentioned coherent tilings precisely, we introduce some terminology on subgraphs of G_ρ. A subgraph of $G_\rho = (V,E,\beta,\rho\times\alpha)$ is a graph $G' = (V',E',\beta',(\rho\times\alpha)')$ where $V' \subseteq V$, $E' = E \cap (V'\times V')$, and β', $(\rho\times\alpha)'$ are the restrictions of β, $\rho\times\alpha$ to V'. Vertex $x \in V'$ is called a border vertex if there is an edge (x,y) or (y,x) in E with $y \notin V'$. The core of G' is the set of vertices of G' which are not border vertices. We write [G'] for the graph which results from G' by erasing the A-values for the border vertices; thus [G'] has a vertex valuation in $(Q\times A)\cup Q$. Let us say that G' matches the transition τ if [G'] and τ are isomorphic (via a bijection preserving vertex and edge labels, hence mapping core to core and border to border).

We say that G_ρ satisfies the condition "there are $\ge$ n copies of τ" if there are $\ge$ n distinct occurrences of graphs [G'] isomorphic to τ within G. Applied to boolean combinations of such conditions, this fixes the meaning of the statement "G_ρ satisfies the constraint C".

2.3 Definition. The run $\rho:V \to Q$ of $\mathcal{A} = (Q,\Delta,C)$ on G is called successful if

(a) each vertex of V is in the core of a subgraph of G_ρ which matches some transition of Δ,

(b) G_ρ satisfies the constraint C.

Let us say that $\mathcal{A}$ accepts G if there is a successful run of $\mathcal{A}$ on G. Given a class **G** of graphs in $DG_k(A,B)$ and a graph set $L \subseteq$ **G**, $\mathcal{A}$ recognizes L relative to **G** if for any graph

G ∈ **G**, we have G ∈ L iff $\mathcal{A}$ accepts G. Then L is called recognizable by tilings (or short t-recognizable) relative to **G**.

The above definition is influenced by the work of Muller and Schupp [MS85]. In their study of the monadic second-order theory of context-free graphs (see Section 5 below) they define graph properties by "forbidden patterns" and close these definitions by the boolean operations and projection (existential quantification). In the present set-up we start with one existential quantification immediately, postulating a "run", and for the expansion of a graph by a run work with "allowed patterns" (given by the transitions). This is in accordance with the general idea that nondeterminstic finite automata specify projections of local properties. (For example, the regular sets of words, resp. trees, are just the projections of locally testable sets.) The case where no projection is applied (which amounts to using acceptors with one state only) is discussed in Section 4. The reason for introducing core vertices (as opposed to border vertices) in transitions is to enable us to specify a local neighbourhood completely by a single transition, i.e. without skipping edges which may be covered by other transitions; again this will be useful in Section 4.

2.4 Theorem. Any t-recognizable set of graphs in $DG_k(A,B)$ is definable in monadic second-order logic (using the signature with equality, the binary predicate symbols E_b for $b \in B$, and the unary predicate symbols P_a for $a \in A$).

Proof Hint. The existence of a run is expressible by existential quantifiers over n disjoint vertex sets if n states are involved (the i-th set containing the vertices where the i-th state is assumed). The conditions (a), (b) of Definition 2.3 are easily formalized in first-order logic.

We end this section with some remarks on graph acceptors applied to infinite structures, a subject which in the present paper is not treated in depth. Over infinite graphs, one should allow more general constraints, including conditions "there are infinitely many copies of transition τ". Considering the case of ω-words (viewed as infinite graphs), one obtains variants of the known models of sequential Büchi automaton, Rabin automaton, and Muller automaton (see e.g. [Th90]), by using this type of constraint or certain boolean combinations of it. In the familiar definitions of these automata, the occurrence of states and not of transitions in a run is constrained; however, both versions yield the same expressive power.

Over the trivially valued infinite grid (either with domain $Z \times Z$ or $\omega \times \omega$), graph acceptors define a generalization of tiling problems as known from the literature on domino games (e.g. [LP81], [Ha86]). A domino game is given by a finite set of quadratic tiles (or dominoes) with quadruples of "colors" (for the top, bottom, left, and right margin of a tile), and a tiling is a placement of copies of these dominoes on the euclidean plane (or one quadrant of it) such that adjacent colors coincide. Changing to the terminology of graph acceptors, quadruples of colors correspond to states, and dominoes correspond to transitions. These transitions have five vertices, namely a core vertex with its four neighbours, the central vertex is valued with a state and the trivial grid value, and the states of the neighbour vertices in a transition fit to the central state (e.g., the top component of the central state coincides with the bottom component of the state of the top neighbour). The standard constraints considered in the literature require that a certain domino be used at least once (the "origin constraint" over whole euclidean plane), or that a domino is used infinitely often ("recurring domino" in [Ha86]). If only a quadrant of the plane is considered, special transitions are introduced for the corner and margin positions, which amounts to an origin constraint.

Tree automata with Büchi acceptance or Rabin acceptance involve a more restrictive constraint than the requirement that certain transitions be used infinitely often in a run; instead the constraint applies to the individual paths of a run. An interesting question is to find suitable constraints on infinite graphs which have good logical properties (e.g. which allow closure under complementation).

3. Recognizability within special classes of graphs

Here we verify that relative to some classes of graphs, the graph acceptors introduced above have the same expressive power as known models of finite state automata.

3.1 Theorem. A set of words, resp. of trees of some bounded degree, is recognizable (by conventional finite automata, resp. finite tree automata) iff it is t-recognizable as a set of graphs.

Proof Hint. The direction from right to left is clear from Theorem 2.4 and the characterization of recognizable word sets and tree sets in monadic second-order logic. It is instructive to explain the converse direction (which is also easy) for an example. We consider the automaton defined in the Introduction. It induces the following transitions in a corresponding graph acceptor (the pairs (state, vertex label) are simply written p0, p1, q0, q1; moreover, * indicates one of the two states p,q):

(p0,q), (p1,p)	as initial transitions,
(*,p0,q), (*,p1,p), (*,q1,p)	as intermediate transitions,
(*,p1), (*,q1)	as final transitions,
(p1) for words of length 1, (p0,q1), (p1,p1) for words of length 2.	

For the word w = 01101, acceptance (with trivial constraint "true") is verified by covering w with these transitions in the following way (building up the run p q p p q):

0	1	1	0	1
(p0,	q)			
(p,	q1,	p)		
	(q,	p1,	p)	
		(p,	p0,	q)
			(p,	q1)

Note that no initial or final states are used in graph acceptors. Since the initial and final letter of a word can only be matched by special transitions (due to the distinction between core and border vertices), the information about initial and final states can be included in transitions. One also notes that over words the transitions of graph acceptors contain redundant components (the first component in the intermediate and final transitions). This is due to the linear structure of word graphs which allows to propagate the relevant information already via pairs of vertices. Another fact is also clear from the example: For the simulation of word automata (as well as tree automata), constraints are not needed; formally, we work with a trivially true condition.

In contrast, we give a trivial example which shows that constraints are needed in general, if monadic second-order definable (or just first-order) properties are to be described. Consider the set of completely unconnected graphs in which there is a vertex labelled a and another one labelled b. The only transitions applicable to accept such graphs are of the form (qa), (qb). Without a constraint also the singletons valued a, resp. b, would be accepted. The reason for including more complex constraints will be clear in the next section.

We call a graph acceptor elementary if it contains only transitions with just one core vertex and only edges which start or end in this vertex, excepting the transitions with empty border. Call a set recognized by an elementary graph acceptor e-recognizable. The example acceptor in Theorem 3.1 above is elementary. It turns out that all recognizable sets of words or trees are e-recognizable (relative to the class of word graphs, resp. tree graphs).

The elementary graph acceptors are close to (and equivalent to) finite automaton models that have been considered on directed acyclic graphs ("dags"). We mention the pdag-automata (working on planar dags) studied in [KS81] and [BDW]. In these papers, a vertex label is a doubly ranked symbol, where the two ranks fix the in-degree and out-degree of vertices carrying this symbol (i.e., the number of ingoing and outgoing edges). A transition for symbol "a" of rank (r,s) has the form (p,a,q) where p is an r-tuple and q an s-tuple of states. Runs are built up by associating a state with each edge (instead of each vertex). Without going into details here, we note that one can simulate these pdag automata by elementary graph acceptors in our sense. For this purpose, it is necessary to use a graph representation which determines an ordering of ingoing and outgoing vertices. This can be done by partitioning the edge set into sets E_{ij}, where $(x,y) \in E_{ij}$ iff (x,y) is the i-th edge with source x and the j-th edge with target y. Moreover, the assignment of states to edges has to be replaced by an assignment of state-tuples to vertices (using (r+s)-tuples for vertices with in-degree r and out-degree s). Our decision to use runs which are assignments of states to vertices is motivated by the formulation of acceptance in monadic second-order logic with quantification over vertex sets.

We also mention briefly that the asynchronous automata considered in trace theory (see [Ma89]) can be viewed as elementary graph acceptors that work on dependency graphs associated with traces (dags in which the edge relation is determined in a certain way by the vertex labeling). This case is of particular interest, because by the deep theorem of Zielonka [Zi87] a reduction to deterministic acceptors is possible (relative to the class of dependency graphs of traces). It follows that a trace language is recognizable in the sense of [Ma89] iff its associated graph set is e-recognizable iff its associated graph set is monadic second-order definable. More details are given in [Th90a].

Relative to the class of arbitrary directed acyclic graphs, however, the power of elementary graph acceptors is very limited, and one has to pass to more general graph acceptors:

3.2 Example. (a) The set of grids $G_{m,2}$ with vertex set $\{1,...,m\}\times\{1,2\}$ (the set of "ladders") is not e-recognizable relative to the class of dags.
(b) The set of grids $G_{m,2}$, the set of all grids, and the set of quadratic grids are t-recognizable relative to the class of finite graphs.

Proof Hint. (a) Assume there is an elementary graph acceptor $\mathcal{A}$ which accepts exactly the set of grids $G_{m,2}$. For simplicity we suppose that there is no constraint; the general case works by the same idea but is more technical. If m is chosen sufficiently large, a successful run of $\mathcal{A}$ on $G_{m,2}$ will have the same pair of transitions on a vertex pair ((i,0),(i,1)) and on ((j,0),(j,1)) with $i < j$. Change the considered grid by modification of two edges: Replace the edge ((i,0),(i,1)) by ((i,0),(j,1)), and replace ((j,0),(j,1)) by ((j,0),(i,1)). Obviously the new graph is not isomorphic to a grid $G_{m,2}$ but still accepted by $\mathcal{A}$, a contradiction. (The new graph violates a first-order sentence which is true for grids, namely: "for any vertex x, a vertex reached from x via an E_1-edge and then an E_2-edge coincides with the vertex reached from x via an E_2-edge and then an E_1-edge". Since the negation of this condition can be checked by an elementary graph acceptor, the e-recognizable dag sets are not closed under complement w.r.t. the class of dags.)

(b) We consider the grids $G_{m,n}$ with $m,n \geq 3$. (The other cases need extra treatment similar to the words of length 1 or 2 in the example of Theorem 3.1.) For the grids $G_{m,n}$ with $m,n \geq 3$, one may work with five states, to be used respectively for the bottom vertices, the top vertices, the left vertices (excepting top and bottom), the right vertices (excepting top and bottom), and all remaining vertices inside. The core of each transition is a square of four vertices, and the border contains the adjacent vertices in all possible 9 versions. We omit the details and also leave it to the reader to refine these transitions (by additional information along the diagonal) for specifying the quadratic grids.

The set of grids clarifies the difference between graph acceptors and the algebraic approach to recognizability of Courcelle. Clearly there are only countably many t-recognizable sets of grids (each of which is recursive), whereas all subsets of the set of quadratic grids are recognizable in the sense of [Co90]. Presently it is not clear how far the two approaches diverge. Many nontrivial monadic second-order formulas can be translated to graph acceptors. But it remains open relative to which graph classes the graph acceptors exhaust the power of monadic second-order logic. The critical unsettled property is closure of t-recognizable sets under complement. By adapting an argument in [KS81, Thm 8.2], one sees that we cannot expect to obtain complementation by reducing the graph acceptors to deterministic ones.

A question of independent interest is to analyze the emptiness problem for particular classes of graph acceptors. It is clearly undecidable if the set of grids can be specified, because in this case the halting problem for Turing machines can be coded in the emptiness problem, using the idea of the undecidability proof for the domino problem (see [LP81]).

4. First-order logic and single state graph acceptors

If $G = (V,(E_b)_{b\in B},(P_a)_{a\in A})$ is a graph (not necessarily finite) and $x\in V$, define the n-sphere around x, written $S(x,n)$, to be the subgraph of G whose vertices are those of distance $\leq n$ to x (where edges may be traversed in both directions). Inductively, define the vertex set $V(x,n)$ of $S(x,n)$ by $V(x,0) = \{x\}$, and $V(x,n+1) = V(x,n) \cup \{y\in V \mid \exists z\in V(x,n)\exists b\in B: (z,y)\in E_b \text{ or } (y,z)\in E_b\}$. We call a subgraph obtainable in this manner a sphere. A key result in the first-order model theory of graphs, due to Hanf [Hf65], states that arbitrary first-order formulas can be reduced to local properties, provided the n-spheres are finite. Here we formulate it for (finite) graphs in $DG_k(A,B)$:

4.1 Sphere Lemma. Let G and H be two graphs in $DG_k(A,B)$. G and H satisfy the same first-order sentences of quantifier-depth n provided the following holds: G and H contain, for each $m < n$ and each isomorphism type σ of a 3^m-sphere, the same number $\leq n\cdot k^r$ of spheres of type σ, or G and H both contain $> n\cdot k^r$ spheres of type σ, where $r = 3^{m+1}$.

The proof given in [Hf65, Lemma 2.3] uses the "Ehrenfeucht-Fraissé game" for first-order logic. A different approach and a more detailed analysis (by syntactic quantifier elimination) is presented by Gaifman [Ga82], focussing on finite graphs and giving applications in graph theory and set theory. Applications to infinite graphs were studied in the seventies by the (East) Berlin model theory group (Hauschild, Herre, Rautenberg, Seese, and others); see for example [HR72], or the bibliography [Mü87, p.100 ff.] for a list of references.

Within the class of finite graphs, we conclude from the sphere lemma that the meaning of an arbitrary first-order formula amounts to a statement on the number of occurrences of certain subgraphs, counted only up to some finite threshold.

4.2 Corollary. In the class $DG_k(A,B)$, a first-order sentence is equivalent to a boolean combination of sentences of the form

$\varphi_{n,H}$: "there are $\geq n$ occurrences of H as a sphere".

For word graphs, a sphere is a segment (or factor). The class of word languages that are determined by occurrences of finitely many given factors, counted up to a finite threshold, has been studied by Beauquier and Pin [BP89]; they call such languages locally threshold testable. Let us generalize this terminology to graphs in $DG_k(A,B)$, replacing word segments by spheres. Then 4.2 says:

4.3 Corallary. A set of graphs in $DG_k(A,B)$ is definable in first-order logic iff it is locally threshold testable.

An analysis of Definition 2.3 above shows that the graphs G_ρ which represent successful runs of a graph acceptor (i.e., satisfy (a) and (b) of Definition 2.3) form a locally threshold testable set. Therefore we have, by 4.3,

4.4 Corollary.

(a) A set of graphs in $DG_k(A,B)$ is t-recognizable iff it is the projection of a first-order definable set of graphs.

(b) A set of graphs in $DG_k(A,B)$ is recognized by a graph acceptor with only one state iff it is first-order definable.

Part (a) of the preceding result allows to restate the complementation problem for graph acceptors: Relative to which classes of graphs is the complement of a projection of a first-order property again a projection of a first-order property? Another interesting question asks for effective procedures which decide whether a monadic second-order property (or a projection of a first-order property) defines in fact a first-order, i.e. locally threshold testable, set. In the domain of words, a positive anwer is given by [BP89]. Already for trees the question is open. In the context of infinite words this problem is raised by Wilke [Wi91].

In monadic second-order logic, the transitive closures of the edge relations can be defined. (For words, this transitive closure is the linear ordering of the positions, for trees the partial tree ordering, and for dags the induced partial ordering.) In first-order logic, the expressive power is increased when the transitive closure of the edge relation is included. For the class of words, this means to proceed from the locally threshold testable to the star-free languages, and special properties of word acceptors are known which characterize the star-free languages. For more general cases (like trees and dags) corresponding characterizations of acceptors are unknown.

5. On infinite words, trees, graphs, and their monadic theory

In this section we give a brief survey on some decidability results for monadic second-order theories. As mentioned in the Introduction, this was the original motivation for the attempt to reduce monadic second-order formulas to finite automata. We consider graphs $G = (V,(E_b)_{b\in B},(P_a)_{a\in A})$ where V is infinite. The monadic second-order theory of G (short: the "monadic theory" of G) is the set of sentences in the monadic second-order language (for the signature under consideration) which hold in G. Büchi [Bü61] and Rabin [Ra69] proved decidability of S1S and S2S, the monadic theories of (ω,S) and $(\{1,2\}^*,S_1,S_2)$, where S is the successor relation on the set ω of natural numbers, and S_1, S_2 are the two successor relations over $\{1,2\}^*$.

In order to discuss extensions of these results, we start with a statement for the case of one successor, assuming that the reader is familiar with Büchi automata over ω-words. For a structure $(\omega,S,P_1,\ldots,P_n)$ with $P_i \subseteq \omega$, denote by $w(P_1,\ldots,P_n)$ the ω-word over $\{0,1\}^n$ which has 1 in the j-th component of position i iff $i \in P_j$. Now the main result in [Bü61] states:

> For any monadic second-order formula $\varphi(X_1,\ldots,X_n)$ in signature S with free set variables $X_1,\ldots,X_n$ one can construct a Büchi automaton $\mathcal{A}$ over $\{0,1\}^n$ such that for all $P_1,\ldots,P_n \subseteq \omega$:
>
> φ holds in $(\omega,S,P_1,\ldots,P_n)$ (with X_i interpreted by P_i) iff $\mathcal{A}$ accepts $w(P_1,\ldots,P_n)$.

For a trivial predicate, say P = ∅, this yields decidability of S1S. Applied to any fixed predicate $P \subseteq \omega$, one obtains that the monadic theory of (ω,S,P) is decidable iff for an arbitrary Büchi automaton $\mathcal{A}$ it can be tested effectively whether $\mathcal{A}$ accepts w(P). For many sets P, the monadic theory of (ω,S,P) has been shown decidable in this way (see [ER66], [Si69]).

5.1 Theorem. ([ER66],[Si69]) The monadic theory of (ω,S,P) is decidable for the following sets P: the set of factorial numbers, the set of powers of k (for any given k), the set of k-th powers (for any given k), and the value set of any polynomial with coefficients in ω.

There are also simple recursive sets P for which the monadic theory of (ω,S,P) is undecidable; the recursion theoretic complexity of such theories is analyzed in [Th78].

For the binary tree, expansions by fixed unary predicates did not attract much attention. More work was devoted to the step from trees to graphs. However, all known decidability proofs for monadic theories of infinite graphs still rest on a reduction to Rabin's Tree Theorem, and recent results of Seese [Se90] seem to indicate that this is necessarily so.

An important example are the context-free graphs introduced by Muller and Schupp [MS85]. They are infinite graphs where one vertex is designated as "origin". (Keeping the framework of the present paper, this vertex may be represented formally by a singleton predicate.) Context-free graphs are characterized in many different ways: (1) by a finiteness condition (see [MS85], similar to regular infinite trees as the ones having only finitely many distinct subtrees), (2) in terms of deterministic graph grammars with "regular" rules (see [Ca90, p.99/100]), (3) as transition graphs of pushdown automata (where each vertex represents a total state, i.e. a pair (automaton state, pushdown content)). For a clear exposition of these equivalences see [Ca90]. The name "context-free graph" refers to still another characterization, stating that they are the Cayley graphs of groups that have a context-free word problem (cf. [BB90]). Intuitively speaking, context-free graphs "deviate in a finite manner" from the tree structure, and one might well be tempted to call them "regular infinite graphs" (which unfortunately would cause confusion).

5.2 Theorem. ([MS85]) The monadic theory of a context-free graph is decidable.

Courcelle [Co90] proves a still more general result, covering all equational graphs. These are obtained as solutions of graph rewriting systems and may have vertices of infinite degree. On the other hand, by [Ca90], the equational graphs of bounded degree are the context-free ones.

Do these results exhaust the graphs with a decidable monadic theory? A negative answer is provided by the following graph which is not context-free (also not equational). It is the transition graph G_{cs} of a recognizer of the context-sensitive language $\{a^ib^ic^i \mid i \in \omega\}^*$. G_{cs} is obtained from the binary tree by taking the leftmost branch with a-edges, using the rightmost branch with reversed edges labelled c, deleting the rest, and inserting, on the i-th level, i edges from left to right labelled b (for all i > 0).

5.3 Example. The monadic theory of G_{cs} is decidable.

The claim can be shown by an interpretation of the monadic theory of G_{cs} in the monadic theory of a suitable structure (ω,S,P). For this purpose, number the vertices of G_{cs} level by level from left to right. In this numbering, mark those numbers which correspond to left branch vertices of G_{cs}. The marked numbers are those of the form $n(n+1)/2$. Let P be the set of these numbers. Within (ω,S,P), one can define the three edge relations of G_{cs} by monadic second-order formulas. By [Si69], the monadic theory of (ω,S,P) is decidable. It follows that the monadic theory of G_{cs} is decidable.

Another approach to find graphs with a decidable monadic theory is to start with a graph where decidability is known and to apply a construction which preserves the decidability. A transfer result of this kind has been shown by Shelah and Stupp (stated in [Sh75], and proved in the unpublished [St75]). Roughly, the theorem states: if a structure $\mathcal{M}$ has a decidable monadic second-order theory, so has the "tree over $\mathcal{M}$", whose elements are the finite sequences over $\mathcal{M}$. In the precise formulation we write, given $x = (m_1,\dots,m_n)$ from M^* and $m \in M$, x^m for the sequence $(m_1,\dots,m_n,m)$.

5.4 Theorem. ([Sh75], [St75]) Let $\mathcal{M} = (M,(R_i)_{i<r},(P_i)_{i<p})$ be a relational structure, where $R_i \subseteq M \times M$ and $P_i \subseteq M$. Define $\mathcal{M}^* = (M^*,S,(R_i^*)_{i<r},(P_i^*)_{i<p})$ by

S x y iff for some $m \in M$ x^m = y
R_i^* x y iff there are $z \in M^*$, $m,m' \in M$ with x = z^m, y = z^m', R_i m m',
P_i^* x iff there are $z \in M^*$, $m \in M$ with x = z^m and P_i m.

If the monadic theory of $\mathcal{M}$ is decidable, so is the monadic theory of $\mathcal{M}^*$.

A similar construction is the "unravelling" of a structure. Given $\mathcal{M}$ as in Theorem 5.4, define $\mathcal{M}' = (M^*,(R_i')_{i<r},(P_i')_{i<p})$ by

R_i' x y iff there are $z \in M^*$, $m,m' \in M$ with x = z^m, y = z^m^m', R_i m m',
P_i' x iff there are $z \in M^*$, $m \in M$ with x = z^m, P_i m.

It is not known for which structures $\mathcal{M}$ we can conclude that $\mathcal{M}'$ has a decidable monadic theory if $\mathcal{M}$ has.

This question brings us back to the decision problem for the monadic theory of structures $(\{1,2\}^*,S_1,S_2,P)$, expansions of the binary tree by fixed predicates. Courcelle raised the question whether the monadic theory of an "algebraic tree" is decidable. A (binary) algebraic tree can be regarded as a structure $(\{1,2\}^*,S_1,S_2,P)$ where $P \subseteq \{1,2\}^*$ is a set of words accepted by a pushdown automaton (see [Co83] for details). We can obtain an algebraic tree $(\{1,2\}^*,S_1,S_2,P)$ by unravelling the transition graph of a pushdown automaton which accepts the language P (where the accepting total states form a designated subset). Thus an interesting case of the above question is to consider the monadic theory of structures $\mathcal{M}'$ for context-free graphs $\mathcal{M}$.

Acknowledgement

I thank Bruno Courcelle, Andreas Potthoff, and Thomas Wilke for many helpful and pleasant discussions.

References

[BB90] J. Berstel, L. Boasson, Context-free languages, in: Handbook of Theoretical Computer Science, Vol. B (J. v. Leeuwen, ed.), Elsevier, Amsterdam 1990, pp. 61-102.

[BP89] D. Beauquier, J.E. Pin, Factors of words, Proc. 16th ICALP (G. Ausiello et al., eds.), Springer LNCS 372 (1989), 63-79.

[BDW] F. Bossut, M. Dauchet, B. Warin, Automata and Rational expressions on planar graphs, in: Proc. MFCS 1988 (M.P. Chytil et al., eds.), Springer LNCS 324 (1988), 190-200.

[Bü60] J.R. Büchi, Weak second-order arithmetic and finite automata, Z. Math. Logik Grundlagen Math. 6, 66-92.

[Bü61] J.R. Büchi, On a decision method in restricted second order arithmetic, in: Logic, Methodology and Philosophy of Science. Proc. 1960 Intern. Congr. (E. Nagel et al., eds.), Stanford Univ. Press, 1-11.

[Ca90] D. Caucal, On the regular structure of prefix rewriting, in: CAAP 90 (A. Arnold, ed.), Springer LNCS 431 (1990), 87-102.

[Co83] B. Courcelle, Fundamental properties of infinite trees, Theor. Comput. Sci. 25 (1983), 95-169.

[Co90] B. Courcelle, Graph rewriting: An algebraic and logic approach, in: Handbook of Theoretical Computer Science, Vol. B (J. v. Leeuwen, ed.), Elsevier, Amsterdam 1990, pp. 193-242.

[El61] C.C. Elgot, Decision problems of finite automata design and related arithmetics, Trans. Amer. Math. Soc. 98 (1961), 21-52.

[ER66] C.C. Elgot, M.O. Rabin, Decidability and undecidability of extensions of second (first) order theories of (generalized) successor, J. Symb. Logic 31 (1966), 169-181.

[Ga82] H. Gaifman, On local and non-local properties, in: Proc. of the Herbrand Symposium, Logic Colloquium '81 (J. Stern, ed.), North-Holland, Amsterdam 1982, pp. 105-135.

[Ha86] D. Harel, Effective Transformations of infinite trees, with applications to high undecidability, dominoes, and fairness, J. Assoc. Comput. Mach. 33 (1986), 224-248.

[Hf65] W.P. Hanf, Model-theoretic methods in the study of elemenatry logic, in: The Theory of Models (J.W. Addison, L. Henkin, A. Tarski, eds.), North-Holland, Amsterdam 1965, pp. 132-145.

[HR72] K. Hauschild, W. Rautenberg, Interpretierbarkeit und Entscheidbarkeit in der Graphentheorie II, Zeitschr. Math. Logik Grundl. Math. 18 (1972), 457-480.

[KS81] T. Kamimura, G. Slutzki, Parallel and two-way automata on directed ordered acyclic graphs, Inform. Contr. 49 (1981), 10-51.

[LP81] H.R. Lewis, C.H. Papadimitriou, Elements of the Theory of Computation, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[Ma89] A. Mazurkiewicz, Basic notions of trace theory, in: Linear Time, Branching Time, and Partial Order in Logics and Models of Concurrency (J.W. de Bakker et al., eds.), Springer LNCS 354 (1989), 285-363.

[MS85] D.E. Muller, P.E. Schupp, The theory of ends, pushdown automata, and second-order logic, Theor. Comput. Sci. 37 (1985), 51-75.

[Mü87] G.H. Müller (ed.), Ω-Bibliography of Mathematical Logic, Vol. III (Model Theory), Springer-Verlag, Berlin, Heidelberg, New York 1987.

[Ra69] M.O. Rabin, Decidability of second-order theories and automata on infinite trees, Trans. Amer. Math. Soc. 141 (1969), 1-35.

[Se90] D. Seese, The structure of models of decidable monadic theories of graphs, Ann. of Pure Appl. Logic, to appear.

[Sh75] S. Shelah, The monadic theory of order, Ann. Math. 102 (1975), 379-419.

[Si69] D. Siefkes, Decidable extensions of monadic second-order successor arithmetic, in: "Formale Sprachen und Automatentheorie" (K. Dörr, G. Hotz, eds.), Bibliographisches Institut, Mannheim, 1969, pp. 441-472.

[St75] J. Stupp, The lattice-model is recursive in the original model, manuscript, The Hebrew University, 1975.

[Th78] W. Thomas, The theory of successor with an extra predicate, Math. Ann. 237 (1978), 121-132.

[Th90] W. Thomas, Automata on infinite objects, in: Handbook of Theoretical Computer Science, Vol. B (J. v. Leeuwen, ed.), Elsevier, Amsterdam 1990, pp. 135-191.

[Th90a] W. Thomas, On logical definability of trace languages, in: Proc. ASMICS Workshop "Free Partially Commutative Monoids" (V. Diekert, ed.), Rep. TUM-I9002, TU München 1990.

[Wi91] Th. Wilke, An Eilenberg Theorem for ∞-languages, in: Proc. ASMICS Workshop "Logics and Recognizable Sets", Tech. Rep. 9104, Inst. f. Informatik u. Prakt. Math., Universität Kiel, 1991 (Extended Abstract in these Proceedings).

[Zi87] W. Zielonka, Notes on finite asynchronous automata, RAIRO Inform. Theor. 21 (1987), 99-135.