Automata Theory on Trees and Partial Orders

Wolfgang Thomas

Christian-Albrechts-Universität zu Kiel
Institut für Informatik und Praktische Mathematik
D-24098 Kiel, Germany
E-Mail: wt@informatik.uni-kiel.de

Abstract. The paper reviews recent results which aim at generalizing finite automata theory from words and trees to labelled partial orders (presented as labelled directed acyclic graphs), with an emphasis on logical aspects. As an important type of labelled partial order we consider pictures (two-dimensional words). Graph acceptors and their specialization for pictures, "tiling systems", are presented, and their equivalence to existential monadic second-order logic is reviewed. Other restricted versions of graph acceptors are discussed, and an intuitive exposition of the recently established monadic quantifier alternation hierarchy over graphs is given.

1 Introduction

The computational model of finite automaton ows a lot of its wide applicability to its good logical and algorithmic properties. Such properties are, for instance, the expressive equivalence between finite automata and monadic second-order logic over words, and the decidability of the emptiness problem (as well as the inclusion problem) for languages recognized by finite automata.

These features of finite automata were essential when finite automata theory was extended from finite words (as inputs) to infinite words ([Bü62]), and from finite words to finite trees ([TW68], [Do70]) and infinite trees ([Ra69]). By the equivalence between logic and automata, monadic second-order formulas could be converted into automata. Via this transformation, the decidability of the emptiness problem for these types of finite automata provided proofs that certain monadic second-order theories are decidable (in particular, the theories S1S and S2S of one, respectively two successor functions). Similarly, the verification of finite-state programs with respect to monadic second-order or temporal logic specifications could be solved effectively, being reducible to the inclusion problem between automaton recognizable languages.

The purpose of this paper is to review some results which throw a light on these questions in the context of finite partial orders. The motivation to do this is twofold: partial orders are a natural "next step" beyond trees, so that a mathematical analysis over this domain should be tried, and partial orders allow to model several aspects of concurrent computations more naturally than words or trees.

We shall adopt a representation of partial orders by directed acyclic graphs with labelled vertices and labelled edges. Since a substantial theory of recognizable sets of infinite partial orders does not yet exist (excepting special cases such as infinite Mazurkiewicz traces and asynchronous automata), we confine ourselves to the case of finite partial orders in the present paper. Many approaches have been developed to obtain "natural" generalizations of finite automata theory to cover partial orders and graphs, among them [KS81], [Cou90], and [Th91]. We concentrate here on the last mentioned proposal, which is closely related to the "tiling systems" of [GR96] over labelled rectangular grids (pictures). This approach is based on the view that a finite automaton is a finite system which checks (by its finitely many transitions) local neighbourhoods of the input structure, thereby associating a state to each point of the input. As it turns out, recognizability by such automata (nondeterministic graph acceptors) is equivalent to definability in the existential fragment of monadic second-order logic. We shall sketch this equivalence proof; it supplies an alternative method compared to the classical approach in linking logic and automata. (While in the classical method complementation results are the essential point and a single inductive proof covers the logical side, the present method does not involve complementation and treats first-order logic and existential second-order quantifiers in two separate steps.)

A master example of partial orders where the new features appear in a transparent way is provided by the class of pictures. In this case, graph acceptors take the simple form of tiling systems. Undecidability results such as for the emptiness of recognizable sets are easy to show, and the recently established monadic quantifier alternation hierarchy is also set up in this domain.

The present paper is meant as an introduction, integrating results from [Th91], [PST94], [GRST96], [Th96a], and [MT96]. Most arguments are presented on the intuitive level, assuming that the reader is familiar with basic automaton constructions and simple facts of logic. In Section 2, we collect the necessary terminology, and in Section 3 we recapitulate some well-known results on tree automata and recognizable tree languages. In the subsequent section, these results are confronted with the corresponding statements on recognizable sets of pictures. In Section 5 we present the general model of finite-state graph acceptor, supplemented by a discussion of some natural restrictions in Section 6. In the last two sections, we outline the hierarchy proof on monadic quantifier alternation and discuss some (mostly ongoing) work concerning further types of labelled partial orders.

I thank Oliver Matz, Ina Schiering, and Sebastian Seibert for many useful discussions and helpful remarks on the subject of this paper.

2 Labelled Partial Orders and Words, Trees, Pictures

A partial order with node labels in a finite alphabet A can be represented as a relational structure $(V, \leq, (P_a)_{a \in A})$ where $\leq$ is a partial order on the nonempty

set V and for each $a \in A$, P_a is a subset of V, containing the elements $v \in V$ which are labelled by the letter a.

In the automata theoretic view, the idea of "local neighbourhood" enters, where one considers the "next-smaller" or "next-larger" neighbours of an element. Thus it is useful to identify a (discrete) partial order with a directed acyclic graph, taking as edge relation E the minimal relation which generates by its reflexive transitive closure the partial order $\leq$. (Thus $(u,v) \in E$ holds iff $u < v$ and there is no w with $u < w < v$.) To be able to handle orderings on neighbour vertices, not only vertices but also the edges are labelled (for example to distinguish between first and second successor in binary trees). Thus, the structures of this paper are vertex labelled and edge labelled graphs of the form

$$G = (V, (E_b)_{b \in B}, (P_a)_{a \in A})$$

where V is the set of vertices, the P_a are disjoint subsets of V whose union is V, and the E_b are disjoint non-reflexive binary relations over V. The edge set is the union $E = \bigcup_{b \in B} E_b$. So we consider a vertex v to be labelled with letter a if $v \in P_a$, and an edge (u,v) to be labelled with letter b if $(u,v) \in E_b$. In the sequel, such graphs are often assumed to be acyclic, i.e. where one obtains a partial order when forming the reflexive transitive closure E^* of the edge set E.

Let us mention some special cases: words, trees, pictures, and grids. A *word* w of length $n(>0)$ over the alphabet A is presented as the labelled acyclic graph $\underline{w} = (\{1,\ldots,n\}, S, (P_a)_{a \in A})$ where $1,\ldots,n$ are the letter positions of w, S is the successor relation on $\{1,\ldots,n\}$, and P_a is the set of positions i in w which carry letter a.

For convenience of notation, we consider as *trees* only binary ones (where each vertex has either two successors or is a leaf). In this case, nodes are representable as words over $\{1,2\}$, the root as the empty word, and the domain $\mathrm{dom}(t)$ of a tree t is a prefix-closed set K of words where for each word $w \in K$ either both or none of $w1, w2$ are in K. A tree $t : \mathrm{dom}(t) \to A$ is presentable as a relational structure $\underline{t} = (\mathrm{dom}(t), S_1, S_2, (P_a)_{a \in A})$, where S_1, S_2 are the two successor relations on $\mathrm{dom}(t)$ and P_a is defined as before. By the *frontier* of a tree we mean the sequence of its leaves in lexicographical order; the frontier word is the sequence of associated node labels. The *frontier language* of a tree language (set of trees) T is the set of frontier words of trees from T.

By a *picture* over A we mean a matrix of letters from A; if such a matrix p has m rows and n columns, the picture domain is the set $\mathrm{dom}(p) = \{1,\ldots,m\} \times \{1,\ldots,n\}$, and the picture a map $p : \mathrm{dom}(p) \to A$. The corresponding relational structure is $\underline{p} = (\mathrm{dom}(p), S_1, S_2, (P_a)_{a \in A})$ where S_1, S_2 contain the pairs $((i,j),(i+1,j))$, respectively $((i,j),(i,j+1))$ (with components in $\mathrm{dom}(p)$), and where $P_a = \{(i,j) \mid p((i,j)) = a\}$. By a *grid* we mean an unlabelled picture (or equivalently a picture over a one-letter alphabet). Also for pictures we shall use the notion of *frontier word*, which we fix to be the first row of a picture. Thus the frontier language of a picture language L is the set of first rows of pictures from L.

It is useful to consider *boundary markers* added to trees and pictures. In the case of binary trees, this will mean that we add # to the alphabet A and add to

each leaf two successors, each labelled with this new boundary symbol #. The extended domain of the tree t including also the boundary positions is denoted $\mathrm{dom}^+(t)$. In the case of pictures we pass from $\mathrm{dom}(p) = \{1,\ldots,m\}\times\{1,\ldots,n\}$ to the set $\mathrm{dom}^+(p) = \{0,\ldots,m+1\}\times\{0,\ldots,n+1\}$ such that the added boundary points are again labelled with #.

3 Automata on Trees

A (nondeterministic) *tree automaton* over A has the form $\mathcal{A} = (Q, A, q_0, \Delta, F)$ where Q is the finite state set, A the alphabet of node labels, q_0 the initial state, $\Delta \subseteq ((A \cup \{\#\}) \times Q)^3$ the transition relation, and $F \subseteq Q$ the set of final states. We impose the restriction that # has to be matched by the initial state, i.e., a transition $((a,q),(\#,q'),(\#,q''))$ only occurs with $q' = q'' = q_0$. (In the literature, transitions from $Q \times A \times Q \times Q$ are usually considered; the present definition is an inessential modification which fits better for a generalization to tiling systems and graph acceptors.) The automaton accepts a tree t if there is a run $\rho : \mathrm{dom}^+(t) \to Q$ which maps the root to a state in F, such that the tree with the extended vertex labelling (now in the set $A \times Q$) can be covered (or: "tiled") by transitions from Δ. Formally, for each triple (u,v,w) of nodes from $\mathrm{dom}^+(t)$, where v and w are the first, respectively second successor of u, the triple $((t(u),\rho(u)),(t(v),\rho(v)),(t(w),\rho(w)))$ has to occur in Δ. A tree automaton is called *deterministic* if its transition set determines an evaluation procedure in the set Q, working from the tree frontier to the root. This means that for any states q', q'' and any input letter a, only one state q exists such that $((a,q),(a',q'),(a'',q'')) \in \Delta$ (independently of a', a''). The tree language recognized by the tree automaton $\mathcal{A}$ consists of those trees (over the given label alphabet A) which are accepted by $\mathcal{A}$, and two tree automata are equivalent if they recognize the same tree language.

The basic facts on tree automata are well-known (see, for example, [GS84]). Let us summarize some statements in the following theorem:

Theorem 1. (a) *For any nondeterministic tree automaton one can construct an equivalent deterministic tree automaton.*
(b) *The class of recognizable tree languages is closed under boolean operations and projection. (A tree language T_B over B is a projection of a tree language T_A over A if there is a map $\pi : A \to B$ such that the trees in T_B originate from the trees in T_A by applying π pointwise.)*
(c) *The emptiness problem for tree automata is decidable.*
(d) *The class of context-free languages coincides with the class of frontier languages of recognizable tree languages.*

Part (a) is shown by the subset construction for tree automata: Given a nondeterministic tree automaton $\mathcal{A}$ with state set Q, the corresponding deterministic automaton has subsets of Q as states, and a transition $((a,S),(a',S'),(a'',S''))$ is admitted if

$$S = \{q \in Q \mid \exists q' \in S' \exists q'' \in S'' : ((a,q),(a',q'),(a'',q'')) \in \Delta_{\mathcal{A}}\}.$$

We skip here further details, which are well-known and can be found in [GS84].

Finally we recall a natural example of a tree language which is not recognizable: the set of trees (over any given alphabet) which consist of a root with two identical subtrees t. A tree automaton which recognizes this set would also accept trees that are not of this form (by a simple pumping argument, applied to the case where the height of t exceeds the number of states). Intuitively, a connection (comparison) between the two subtrees is possible within the run of an automaton only via the transition at the root; the finite number of such transitions does not suffice to distinguish infinitely many trees t.

4 Automata on Pictures: Tiling Systems

In order to transfer nondeterministic automata from trees to pictures we use different transitions. Over the label alphabet A (extended by the boundary marker $\#$) and the state set Q, we consider (2×2)-matrices over $(A \cup \{\#\}) \times Q$. Due to the use of a boundary around pictures, it is not necessary to introduce initial and final states. (For example, the occurrence of an "initial state" at the top left corner of a picture and of a "final state" at the bottom right corner can be imposed by allowing corresponding transitions where $\#$ occurs on the top row and left column, respectively on the bottom row and right column.) Thus a *tiling system* over A is a triple $\mathcal{A} = (Q, A, \Delta)$ with finite state set Q and a finite set Δ of (2×2)-matrices over $(A \cup \{\#\}) \times Q$ (called *transitions* or *tiles*). A tiling system accepts a picture p over A if there is a run $\rho : \mathrm{dom}^+(p) \to Q$, inducing a labelling of $\mathrm{dom}^+(p)$ in $(A \cup \{\#\}) \times Q$, such that each (2×2)-submatrix of adjacent positions matches a transition from Δ. Such a covering of a picture over $(A \cup \{\#\}) \times Q$ is called a *tiling* by $\mathcal{A}$, and the corresponding run *accepting*. Again, the picture language recognized by $\mathcal{A}$ is the set of pictures (over A) accepted by $\mathcal{A}$, and a picture language is called recognizable it consists of the pictures accepted by some tiling system.

The notion of determinism for tiling systems is not as canonical as for tree automata. Here we call a tiling system *deterministic* if its tiles induce a unique tiling on any given picture over A, starting from the top left corner down-right towards the bottom right corner. More precisely, given the entries (a, q) for the upper row and left column of a tile, as well as the alphabet letter at the bottom-right, there is only one matching tile, i.e. a unique state assignment for the bottom-right position. Moreover, there is only one tile to be put on the top left corner of a picture, i.e., whose top and left alphabet letters are $\#$; and such a uniqueness condition also holds for the continuation of a tiling along the leftmost column and along the top row of a picture: so a left column tile is determined uniquely by its upper row entries of the form $(\#, q)$, (a, q'), and a top row tile is determined uniquely given its left column entries $(\#, q)$ and (a, q'). Such tiling systems are a version of the tessellation automata of [IN77].

As we now verify, the preceding theorem concerning tree automata fails for tiling systems in the parts (a), (b), (c); only for part (d) a corresponding (but modified) claim can be stated. The following statement combines proofs of

[PST94] (for (a)), [GR96] or [GRST96] (for (b) and (c)), and [LS96] (for (d)). (As the author recently learned, all these results already appear, presented in a different terminology, in the unpublished dissertation [Sp85], partly communicated in [Sp86].)

Theorem 2. (a) *Not for each tiling system there is an equivalent deterministic tiling system.*
(b) *The class of recognizable picture languages is closed under union, intersection, and projection, but not under complement.*
(c) *The emptiness problem for tiling systems is undecidable.*
(d) *The class of context-sensitive languages coincides with the class of frontier languages of recognizable picture languages.*

Proof. (a): A suitable example is the set L of all quadratic pictures where label b occurs once on the rightmost column and once on the bottom row, moreover these two b are on the same counterdiagonal (i..e., in the same distance to the bottom right corner), and where label a occurs at all other positions.

Recognizability of L by a (nondeterministic) tiling system is verified easily: A special kind of state is propagated along the diagonal (starting from the top left corner), and a point on this diagonal is guessed (by nondeterminism), from which two "signals" are sent (again in the form of special states), one horizontally to the right, one vertically to the bottom. If at the two border points where these signals arrive letter b occurs, this information can be transmitted to the bottom right corner (where the transitions are defined as to check this). The test that otherwise letter a occurs is easily implemented, as well as the test that the picture is a square (using a continuation of the first mentioned "diagonal signal").

Now suppose that a deterministic tiling system recognizes L. By determinism, on a picture from L the states of an accepting run are uniquely determined except for the last column and the last row (where transitions may depend on the occurrences of label b). By finiteness of the state set, one can find on a sufficiently large square two different placements ("options") of the b in the last column, and correspondingly in the last row, such that in the associated accepting runs the penultimate state of the last column is the same. Consider the picture which results by taking the first option for b in the last column, the second option for b for the last row. On this picture an accepting run exists (take the run according to the first option and insert on the last row the state sequence according to the second option). This contradicts the definition of L.

(The preceding proof shows that the subset construction fails for the domain of pictures, in contrast to the case of trees.)

(b): It is easy to verify closure under union, intersection and projection. To show non-closure under complement consider the set K of pictures pq over the alphabet $\{a, b\}$ where p and q are both square pictures and $p \neq q$. To check this by a tiling system, two properties have to be verified: the size of the matrix should be $n \times 2n$ for some n, and there should be in p and q a pair of corresponding positions carrying different letters. The first property is checked, for example, by two "diagonal signals" as mentioned above (from the top left corner to the

bottom row and up again to the top right corner). For the second, the position u in p is guessed by nondeterminism, while the corresponding position v in q is determined by the coincidence of two further "signals", one passing horizontally from u to the right, the other passing down-right from u along a diagonal, from there vertically upwards to the first row, and from there down-right as a diagonal again.

Let us show that the complement of K is not recognizable. The pictures of size different from $n \times 2n$ are detected easily (use a diagonal signal down-right and then up-right and check that it does not arrive at the top-right corner). Thus it suffices to show that the set of pictures of the form pp, where p is square, is not recognizable. A tiling system can transfer the information from the left square grid (of size $n \times n$) to the right square grid only via the two stripes of states along the border between the two half pictures (of square form). Assuming k states and l alphabet letters, the number of such stripes of tiles is $(k \cdot l)^{2(n+2)}$. However, the number of possible $(n \times n)$–squares over the label alphabet grows by the rate 2^{n^2}. Thus, for sufficiently large n we find distinct squares p and q of side length n such that on tilings over pp and qq the central two stripes of states are identical. Thus pq and qp also admit tilings, a contradiction.

(This argument is an adjustment of the above-mentioned idea to obtain non-recognizable tree languages: There the set of trees with two identical subtrees below the root was used, and the inability of tree automata to provide enough flow of information between these two subtrees was noted. Similarly, the two $(n \times n)$-pictures as considered here are connected only via the central (one-dimensional) stripe of transitions, which is again insufficient to provide enough information flow between the two pictures.)

(c): For any Turing machine $\mathcal{M}$ we can construct a tiling system $\mathcal{A}_\mathcal{M}$ over an appropriate label alphabet which accepts some picture iff $\mathcal{M}$ halts when started on the empty tape. The idea is to let $\mathcal{A}_\mathcal{M}$ accept the pictures which code halting computations of $\mathcal{M}$ started on the empty tape. Such a halting computation is finite in space and time (the two dimensions of the picture). Thus, the first line of such a picture represents the initial configuration: a sequence of blanks together with one pair (s_0, blank) (where s_0 is the initial state of $\mathcal{M}$). The correct succession of Turing machine configurations can be checked using (2×2)-square transitions. That the picture is sufficiently large to include all work cells of the computation is guaranteed by excluding transitions for border points of the picture which code work cells. Finally the last line should include a final (= halting) state of $\mathcal{M}$.

(d) Given a linear bounded automaton (LBA) $\mathcal{M}$, defining a context-sensitive language L, one can apply the idea of the previous proof to construct a tiling system which accepts a picture p iff the first row of p is a word accepted by $\mathcal{M}$. Conversely, the computation by a tiling system has to be simulated by an LBA. The main problem here is the parallel mode in which states are assigned to vertices row by row. This has to be broken up into a (nondeterministic) sequential process as performed by an LBA. Since there is no bound on the computation time of the desired LBA, this is possible. □

Over binary trees and over pictures, there is an internal structure of transitions: the role of each vertex in a transition is fixed by means of the two successor relations S_1, S_2. This can be taken as a motivation to work with even simpler transitions or tiles, in which only two vertices appear (rather than three or four as in tree automata or tiling systems), without a loss of expressive power. In both cases, for trees and pictures, it suffices to have transitions from $((A \cup \{\#\}) \times Q)^2$, i.e. connecting two vertices only, together with the distinction whether the first is related to the second via S_1 or via S_2. In the case of pictures, one speaks of the reduction of tiling systems to "domino systems" (cf. [GR96]).

5 Graph Acceptors and Existential Monadic Logic

Over words or trees, there is a simple argument to show that monadic second-order formulas and finite automata are expressively equivalent: Acceptance of a word or tree by a finite automaton can be described with an existential monadic second-order formula (where the existential set quantifiers are used to express the existence of a run, i.e. of a state sequence). Conversely, the construction of automata for given monadic second-order formulas is done inductively over the construction of formulas. To simplify the technical details, one first eliminates first-order variables (in terms of second-order variables which range over singletons); then it suffices to treat atomic formulas with set variables only, and as induction step the treatment of the boolean connectives "not", "or", and of the existential set quantifier suffices. By the closure of automaton definable sets under complement, union, and projection, these induction steps are trivial.

For pictures (and hence for acyclic graphs or partial orders in general), this inductive argument fails because the class of automaton definable sets is not closed under complement. So a logic which matches recognizability should not be closed under negation. It turns out that existential monadic second-order logic is appropriate (over graphs of bounded degree); and that a different proof strategy from formulas to automata can be adopted. It starts with a characterization of *first-order logic* and treats monadic second-order quantification separately. This approach to the proof, where first-order logic is not eliminated but emphasized, yields a uniform characterization of finite-state acceptors by existential monadic second-order logic (over labelled graphs of bounded degree, thus including the cases of words and trees). Only in special cases, where a closure result for negation also holds (as it happens over words and trees) the characterization extends to cover entire monadic second-order logic. In this sense, existential monadic second-order logic is a more natural counterpart to finite automata than full monadic second-order logic.

Let us explain this approach in a little more detail. (For a full technical treatment, see e.g. [Th96].) First we briefly fix the logical terminology. Over graphs $G = (V, (E_b^G)_{b \in B}, (P_a^G)_{a \in A})$, formulas of monadic second-order logic involve variables $x, y, \ldots$ for vertices and $X, Y, \ldots$ for sets of vertices; they are built up from atomic formulas

$$P_a(x) \text{ (for } a \in A), \; E_b(x,y) \text{ (for } b \in B), \; x = y, \; X(y)$$

by means of the connectives $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$ and the quantifiers $\exists, \forall$ which may be applied to either kind of variable. The notation $\varphi(x_1, \ldots, x_m, X_1, \ldots, X_n)$ indicates that in the formula φ at most the variables $x_1, \ldots, x_m, X_1, \ldots, X_n$ occur free, i.e., not in the scope of a quantifier. Formulas without free variables are called sentences. If $G = (V, (P_a^G)_{a \in A}, (E_b^G)_{b \in B})$ is a graph, $v_1, \ldots, v_m \in V$, $V_1, \ldots, V_n \subseteq V$, the satisfaction relation

$$(G, v_1, \ldots, v_m, V_1, \ldots V_n) \models \varphi(x_1, \ldots x_m, X_1, \ldots, X_n)$$

holds if φ is formed for the signature given by the label alphabets A, B and satisfied in G when interpreting x_i by v_i, X_i by V_i, and of course "=" by equality, P_a by P_a^G, and E_b by E_b^G. The superscripts G thus distinguish the relations in interpretations from relation symbols in formulas; they are omitted (as done also before) when no confusion arises.

Let $\mathcal{K}$ be a class of graphs. Relative to $\mathcal{K}$, a sentence φ defines the (graph) language $L(\varphi) = \{G \in \mathcal{K} \mid G \models \varphi\}$. A language $L \subseteq \mathcal{K}$ is called definable in monadic second-order logic if some sentence φ with $L = L(\varphi)$ exists.

A formula

$$\exists \overline{X_1} \forall \overline{X_2} \ldots \exists / \forall \overline{X_k} \varphi(\overline{X_1}, \overline{X_2}, \ldots \overline{X_k}, \overline{Y}),$$

where the $\overline{X_i}$ and $\overline{Y}$ are blocks of second-order variables (possibly of different length) and φ is a first-order formula, is called a Σ_k-formula. Σ_1-formulas are also called existential monadic second-order formulas (EMSO-formulas). A set of graphs is said to be Σ_k-definable if a defining Σ_k-sentence exists.

To establish a general bridge between EMSO-formulas and "automata", we recall the notion of graph acceptor (following the idea of [Th91]). As input graphs we allow graphs (not necessarily acyclic) whose degree is bounded by a constant d (which means that for any vertex u there are at most d neighbours connected by an edge from or to u, in either direction). This boundedness of degree reflects our motivation to define graph properties by checking local neighbourhoods with a finite device: If there was no bound on degree, a finite device (being able to store only a finite amount of "local neighbourhoods") will confuse different neighbourhoods presented as inputs. Of course, such an approach is also conceivable, but for the present treatment we prefer to avoid the resulting complications.

As a precise description of "local neighbourhoods" in graphs we use the notion of r-sphere. Call (for $r \geq 0$) *r-sphere around vertex v in the graph G* the induced subgraph over those vertices in G which have distance $\leq r$ to v, and with v as designated center. (The distance of u to v is $\leq r$ if there is a path $v_0 v_1 \ldots v_k$ with $k \leq r$, $v_0 = v$, $v_k = u$, and $(v_i, v_{i+1}) \in E$ or $(v_{i+1}, v_i) \in E$ for $i < k$.) Clearly, if the graphs under consideration are of bounded degree (and of a fixed signature regarding the labellings), there are only finitely many possible isomorphism types of r-spheres.

The automata to be introduced now accept graphs by associating states to vertices (as in tree automata) and by checking the existence (or nonexistence) of local neighbourhoods in the graph with this state assignment. An important feature is that the checking process may "count" occurrences of local neighbourhoods up to a certain fixed threshold number. Formally, a *graph acceptor* over

the label alphabets A, B has the form $\mathcal{A} = (Q, A, B, \Delta, Occ)$ where Q is a finite set (of "states"), Δ is, for some $r \geq 0$, a finite set of r-spheres with vertex labels in $A \times Q$ and edge labels in B, and Occ is a boolean combination of conditions "there are $\geq n$ occurrences of spheres of type τ" (where τ is an r-sphere isomorphism type over the label alphabets $A \times Q$ for vertices and B for edges).

As for tiling systems, we call Δ the set of *transitions* (or *tiles*). The item Occ is called the *occurrence constraint.*

The graph acceptor $\mathcal{A}$ *accepts* the graph G if it can be "tiled by transitions" such that a consistent assignment of states to vertices (a "run") is defined by this tiling and such that the occurrence constraint is satisfied. Formally, there should be a run $\rho : V \to Q$ such that each r-sphere of the expanded graph G_ρ with vertex labels in $A \times Q$ matches a transition from Δ, and the occurrence numbers of these spheres are compatible with the constraint Occ. We call this covering of G_ρ an "accepting tiling" of G. The graph language recognized by $\mathcal{A}$ (relative to the graph class $\mathcal{K}$) is $L_\mathcal{K}(\mathcal{A}) = \{G \in \mathcal{K} \mid \mathcal{A} \text{ accepts } G\}$. We say that $L \subseteq \mathcal{K}$ is *recognizable* iff $L = L_\mathcal{K}(\mathcal{A})$ for some graph acceptor $\mathcal{A}$.

Now the equivalence between EMSO-logic and graph acceptors reads as follows (cf. [Th91], [Th96]):

Theorem 3. *For any class $\mathcal{K}$ of graphs of bounded degree, a graph language $L \subseteq \mathcal{K}$ is recognizable by a graph acceptor iff L is EMSO-definable in $\mathcal{K}$.*

Proof. We shall only give a rough outline of the proof, which also shows that a graph language L is recognizable by a graph acceptor with only one state iff L is first-order definable.

For the direction from left to right we code states by 0-1-vectors (say of length m), whence state assignments to the graph vertices correspond to m-tuples of subsets of the vertex set. Thus, acceptance by a graph acceptor can be formalized by a statement $\exists X_1 \ldots X_m \varphi(X_1, \ldots, X_m)$, where φ is a boolean combination of statements "r-sphere τ occurs $\geq n$ times". Such a boolean combination is directly expressible as a first-order formula (starting with the quantifiers "there are distinct vertices $x_1, \ldots, x_n$").

For the converse direction we have to transform an EMSO-sentence into an equivalent graph acceptor. It will suffice to transform a *first-order* sentence into an equivalent one-state graph acceptor, since existential set quantifiers express the same as the requirement that an assignment of states to vertices (i.e., a run) should exist. A one-state graph acceptor can specify graph properties which fix the occurrence numbers of r-spheres up to a threshold t. Properties determined in this simple way (for suitable r and t) are called *locally threshold testable.* We are done if we can verify that any first-order formula can only specify a locally threshold testable graph property.

This claim is the content of "Hanf's Theorem" (shown already 1965 in the context of first-order model theory). The proof has to establish the following, for any given m: If two graphs are distinguishable by first-order formulas of quantifier-depth m, then, for some r and t, the occurrence numbers of r-spheres in the two graphs, counted up to threshold t, differ. In other words: For any m

there should exist r and t such that the same occurrence numbers of r-spheres in graphs G, G' counted up to threshold t imply that G, G' satisfy the same sentences of quantifier-depth m. In this form, the claim can be shown by an application of the Ehrenfeucht-Fraissé-game. For the details we refer the reader to [EF95] or [Th96]. □

Finally, let us compare graph acceptors with automata over words and trees, and with tiling systems over pictures. Clearly, conventional transitions of finite automata over words and trees can be captured within 1-sphere transitions in graph acceptors. Moreover, the use of initial and final states is superfluous in graph acceptors: For example, over words an initial state should only occur at a 1-sphere center which has no left neighbour in the 1-sphere (with respect to successor, the edge relation). Thus the special role of initial and final states is captured by their occurrence in special transitions. By the same reason, pictures can be recognized without the use of a boundary symbol #, and again it turns out that 1-spheres suffice. In all these cases (when simulating classical automata and tiling systems) the occurrence constraints of graph acceptors are not needed. In the next section we shall see that over unrestricted acyclic graphs (or partial orders), these constraints are indispensable.

6 Restricted Models of Graph Acceptors

The purpose of this section is to analyze the model of graph acceptor in some more detail. In particular, we show that the two "complicated" features of graph acceptors are necessary over general acyclic graphs, if the expressive power should correspond to EMSO-logic: the admission of arbitrary sphere radius in the transitions, and the occurrence constraints.

First we consider the issue of restricted sphere radius in transitions.

Proposition 4. *Let L_n be the set of "n-supergrids", which have vertex label "a" throughout and are obtained from standard grids by substituting for any edge an edge sequence of length n (called "superedge"). L_n is recognizable (in the class of partial orders) by a graph acceptor with $2n$-sphere transitions, but not by graph acceptors with 1-sphere transitions.*

Proof. Clearly, L_n is recognizable by a graph acceptor with $2n$-sphere transitions. For contradiction, consider a graph acceptor $\mathcal{A}$ which recognizes L_n (say for $n \geq 4$) with 1-sphere transitions. In an accepting run of a large enough n-supergrid, one can pick two occurrences of the same 1-sphere transition at the central positions of two superedges which are located between the same two columns of a n-supergrid (provided there is a sufficiently high number of rows in the supergrid). Obtain a new graph by exchanging the targets of the outgoing edges of the two 1-spheres covered by these transitions. The new graph is still acyclic and is again accepted by $\mathcal{A}$. Since the new graph is no more a supergrid, we obtain the desired contradiction. □

An adaptation of the argument shows the same claim for any given sphere radius r (instead of $r = 1$). As remarked before, sphere radius 1 suffices over words, trees, and pictures. We do not know a precise description of the class of acyclic graphs where in graph acceptors the use of 1-sphere transitions suffices. It seems that planarity conditions are useful in this context (as they occur also in the set-up of regular expressions for describing languages of labelled acyclic graphs, cf. [BDW95]).

Let us turn to the occurrence constraints. In the next proposition we use graphs G_n which are made up of vertices $u_1, \ldots, u_n$ and $v_1, \ldots, v_n$ as follows: From u_i there are two edges, one to v_i (labelled 0) and one to $v_{((i+1) \bmod n)}$ (labelled 1). Imagine the u_i and the v_i arranged in two circles (modulo n), with two pointers from each vertex of the first circle to the second circle.

Proposition 5. *Let L be the set of acyclic graphs G_n where at least one vertex u_i is labelled b and the remaining vertices (not labelled b) are labelled a. L is recognizable by a graph acceptor, however not by a graph acceptor without occurrence constraint.*

Proof. The set of the graphs G_n is recognizable even without occurrence constraints, when the vertex labellings are discarded. Inclusion of such a constraint (requiring at least one transition with vertex label b) then serves recognize L. Now, for a contradiction suppose that L is recognizable without occurrence constraints. Consider the graphs G_n over $u_1, \ldots, u_n$ and $v_1, \ldots, v_n$ with precisely one label b, say at u_1. For sufficiently large n, there will be an accepting run (and corresponding tiling) where a transition is repeated, say with centers at u_i and u_j and such that u_1 is not covered by these two copies of the transition. Then the graph with vertices $u_{i+1}, \ldots, u_j, v_{i+1}, \ldots v_j$ (built up modulo $j - i$), which has no label b, admits also an accepting tiling, a contradiction. □

In some situations, however, the occurrence constraints can be eliminated (at the cost of more states in graph acceptors). In particular, this applies to graph acceptors over words, trees, and pictures. The idea is to implement a threshold counting procedure within the transitions, using the partial order to avoid loops in the counting process. It is essential that the overall counting result can be collected at some special vertex and that the intermediate counting results are propagated without duplication. (So we refer to a "designated" outgoing edge of each vertex, which has to be determined uniquely in terms of the edge labelling.)

Proposition 6. *Let $\mathcal{K}$ be a class of acyclic graphs which have a designated out-edge for each vertex and furthermore a vertex which is reachable from any vertex by a path (i.e., a greatest element of the associated partial order). Then a language $L \subseteq \mathcal{K}$ is recognizable iff it is recognizable by a graph acceptor without occurrence constraints. (The same holds if all vertices of the graphs under consideration have a designated in-edge and a smallest element in the associated partial order.)*

Proof. Consider a graph acceptor with state set Q, transitions $\tau_1, \ldots, \tau_k$ (say of radius r), and occurrence constraint Occ in which t is a threshold such that occurrence numbers $\geq t$ are not distinguished in Occ. We construct a new graph acceptor whose states are vectors $(q, n_1, \ldots, n_k)$ with $n_i \leq t$ for $i = 1, \ldots, k$. At vertex v this vector indicates that state $q \in Q$ is assumed and "up to now" the transition τ_i has occurred n_i times. These occurrence numbers are updated following the paths of the partial order of the input graph. The designated out-edge serves to avoid double-counting: The accumulated occurrence numbers are transferred further only along the designated outgoing edge. Thus, for an r-sphere of type τ_i whose center has no incoming edges, only the vector $(n_1, \ldots, n_k)$ with $n_i = 1$ and $n_j = 0$ for $j \neq i$ is allowed. Any given r-sphere, say of type τ_i, which has incoming edges, is (in its center) supplied with a vector $(n_1, \ldots, n_k)$ where each n_j is the sum of the j-th components of the sources of incoming edges which are designated, and where furthermore 1 is added to n_i (to capture that the present type is τ_i). Finally, r-sphere transitions for the greatest element (the unique vertex without outgoing edges) are allowed only for the case that the center vertex is labelled with some vector $(n_1, \ldots, n_k)$ which satisfies Occ.

The proof for the case of designated in-edges and the existence of a smallest element in the partial order is analogous. □

It is clear that graph acceptors over words, trees, and pictures are subsumed under the preceding proposition, so that occurrence constraints can indeed be eliminated in these cases. Formally, for trees one applies the second case of the proposition, taking the (unique, if existing) incoming edge of a vertex as designated. Over pictures one takes as designated edges the horizontal ones, except for the vertices of the last column (detected by the lack of a horizontal out-edge in a transition) where the vertical out-edge is taken as designated. A detailed construction is given in [GRST96]; it shows that graph acceptors and tiling systems have the same expressive power over pictures.

7 Beyond Recognizability: The Monadic Hierarchy

We have seen in Section 4 that the class of recognizable picture languages, or equivalently the class of EMSO-definable picture languages, is not closed under complement. How large is the gap between recognizability and definability in full monadic second-order logic? As shown in [MT96], this gap is as large as possible: one obtains an infinite hierarchy above the recognizable picture languages, induced by the alternating application of complementation and projection (or speaking logically, by the alternating application of existential and universal set quantifiers). In other words, the classes of Σ_k-definable picture languages form an infinite hierarchy for increasing k. Moreover, this holds even over unlabelled pictures, i.e. rectangular grids. The proof involves a nice application of automata theory to a "purely logical" question.

To explain this result, note that any pair (m, n) of (positive) natural numbers fixes uniquely a grid, which we shall denote by $G(m, n)$; it is the grid with m

rows and n columns. Any binary relation over the positive natural numbers thus corresponds to a grid language. We consider unary functions as special binary relations and thus associate with the function f over the positive natural numbers the grid language

$$L[f] = \{G(m, f(m)) \mid m > 0\}.$$

Now the hierarchy is witnessed by the grid languages $L[f_k]$, where f_k is a variant of the k-fold exponential function over 2 (called s_k in the sequel). Inductively, we define

$$s_0(m) = m, \quad s_{k+1}(m) = 2^{s_k(m)}, \quad f_0(m) = m, \quad f_{k+1}(m) = f_k(m)2^{f_k(m)}$$

Now the main result of [MT96] reads as follows:

Theorem 7. (a) *If $L[f]$ is Σ_k-definable, then $f(m)$ is in $s_k(\mathcal{O}(m))$.*
(b) *The grid language $L[f_k]$ is Σ_{2k+3}-definable.*
(c) *The hierarchy of the classes of Σ_k-definable grid languages (for $k = 1, 2, \ldots$) is infinite.*

Proof. First we note that from (a) and (b) we obtain (c), using that $f_{k+1}(m)$ is not $s_k(\mathcal{O}(m))$. It remains to show (a) and (b).

(a): Let $\varphi(Y_1, \ldots, Y_n)$ be a Σ_k-formula, defining pictures over the label alphabet $\{0,1\}^n$. For any given column length m we shall transform φ into a finite *word* automaton $\mathcal{A}_m$ which scans pictures of column length m from left to right, column by column; so the input letters are columns of length m with entries in $\{0,1\}^n$, and the state set also depends on the column length m. It suffices to show that

> (+) for column length m there is a nondeterministic finite automaton $\mathcal{A}_m$ which is equivalent to φ over pictures of column length m and has $s_{k-1}(c^m)$ states for some constant c (depending only on φ).

Then the shortest word accepted by $\mathcal{A}_m$ has length $\leq s_{k-1}(c^m)$. Hence, if φ defines a grid language $L[f]$, then $f(m)$ is $s_k(\mathcal{O}(m))$, as was to be shown.

The claim (+) is proved by induction on k, which is the number of set quantifier blocks in

$$\varphi(Y_1, \ldots, Y_n) = \exists \overline{X_k}\, \forall \overline{X_{k-1}} \ldots \exists/\forall \overline{X_1}\, \psi(Y_1, \ldots, Y_n, \overline{X_k}, \ldots, \overline{X_1}),$$

equivalently

$$\exists \overline{X_k} \neg \exists \overline{X_{k-1}} \ \ldots \ \neg \exists \overline{X_1}\, \psi'(Y_1, \ldots, Y_n, \overline{X_k}, \ldots, \overline{X_1}),$$

with first-order kernel ψ, respectively ψ'. For simplicity assume that the variable blocks $\overline{X_i}$ all have the same length l. Consider the case $k = 1$. The formula $\exists \overline{X_1} \psi'(Y_1, \ldots, Y_n, \overline{X_1})$ defines a picture language over the alphabet $\{0,1\}^n$ which (by Theorem 3 and the last remark of Section 6) is recognizable, say with a tiling system over the state set Q. (We suppose, without loss of generality, that

on the boundary of a picture only a dummy state is assumed.) When reading a picture column by column from left to right, one can view the tiling system as a nondeterministic finite word automaton; for column length m a state of this automaton is an m-tuple over Q. Thus for column length m the automaton has $c^m (= s_0(c^m))$ states where c is a constant depending only on the tiling system (and hence on φ).

In the induction step (from k to $k+1$) we use the fact that a complementation step (absorbing $\neg$) and a projection step (absorbing l existential quantifiers) have to be carried out. For nondeterministic automata, the first step, involving the subset construction, increases the number of states by an exponential (thus passing from the bound $s_{k-1}(\mathcal{O}(m))$ to $s_k(\mathcal{O}(m))$). Since the second step leaves the respective number of states as it is, we obtain the bound on the number of states as required in (+).

(b): For each $k > 0$ we have to provide a Σ_{2k+3} formula which defines the grid language $L[f_k]$. The idea is to describe by such a formula, given any grid say of column length m, a counting process depending on k and m: On the grid, we imagine writing binary numbers of length $f_{k-1}(m)$ on the top row, in succession from 0 up to $2^{f_{k-1}(m)} - 1$. To "write" means to describe a corresponding subset Z_k of the top row (which induces by its characteristic function a sequence of bits). The overall length of the sequence of all these binary numbers is then $f_{k-1}(m) \cdot 2^{f_{k-1}(m)}$, which is $f_k(m)$, i.e. the desired row length for a grid of column length m.

For the definition of the set Z_k, one proceeds by induction on k. We shall indicate how to obtain a monadic second-order formula; the detailed analysis leading to the formula complexity Σ_{2k+3} will not be presented here.

As a preparation, it is useful to recall the definition of transitive closure in monadic logic. If R is a (definable) binary relation and u an element (vertex), we can define the set of vertices v which are reachable from u via a path made up of pairs from R. Namely, a vertex v is reachable from u in this sense iff it belongs to all sets X which contain u and such that for any pair $(w, w') \in R$ with $w \in X$, also w' belongs to X. This type of definition allows, for instance, to describe the elements of the down right diagonal starting at a given vertex u. In the sequel we shall assume the definability of such sets tacitly.

Let us show the claim for $k = 1$. We have to describe, over column length m, the counting process from 0 to $2^m - 1$, using binary numbers of length m; this will fix the row length to be $m2^m (= f_1(m))$. We do this by describing two subsets Y_1, Z_1 of the first row, which we identify with two 0-1-sequences. The first sequence Y_1 is in $(10^{m-1})^*$, i.e. it marks by its entries 1 the first digits of the binary numbers, while the second sequence Z_1 is the sequence of these binary numbers, each of length m, in succession. Now a position is in Y_1 if it belongs to the transitive closure of the first top row position under taking the m-th horizontal successor. This is expressible by a monadic formula if the m-th horizontal successor is definable. For this, one observes that two top row positions u, v are in distance m (over a grid of column length m) if there is a third position w on the bottom row which is reached simultaneously in two ways: along

a down right diagonal from u and along a vertical line down from v. Transitive closure definitions can be used to express this, thus Y_1 is definable. Turning to the definition of Z_1, the essential point is to describe that two successive number representations stand for numbers i, j with $i + 1 = j$. But this is easy because we can say (as above) when two top row positions are in distance m (and then can fix corresponding bits in two successive binary numbers).

On grids whose row length is greater than $m2^m$ we need these definitions in slightly more general form: We have to refer to an iterated concatenation of the sequences Y_1 and Z_1 (by stipulating the successor of $2^m - 1$ to be 0 again). We denote these iterated versions of Y_1, Z_1 on longer grids by Y_1', Z_1' (which are again definable in monadic logic). Furthermore, we need to compare bit-sequences of length m over a longer distance than m, if they start at corresponding positions u, v of the Y_1'-marking, where u comes before v. We say that v is reachable from u by proceeding to the right, and express that the m bits from vertex u (where Y_1' is 1) up to u' coincide with the m bits from v (where Y_1' is also 1) up to v' as follows: for any two "corresponding" vertices $\hat{u}$ and $\hat{v}$ between u and u', respectively between v and v' (inclusive), the bits at $\hat{u}$ and $\hat{v}$ coincide. Vertices $\hat{u}$ and $\hat{v}$ "correspond" if they are connected via two auxiliary vertices u^* and v^* as follows: u^* is reached vertically downwards from u and down-left diagonally from $\hat{u}$, similarly v^* is reached vertically downwards from v and down-left diagonally from $\hat{v}$, and v^* is reachable from u^* by passing horizontally to the right. All these connections are describable using transitive closure definitions.

The induction step is explained just for the case $k = 2$ (this suffices to clarify the general construction). We have to describe the counting process from 0 up to $2^{m2^m} - 1$ using binary numbers of length $m2^m$. Here we use the sequences Y_1', Z_1'. We have to define two subsets Y_2, Z_2 of the top row (again viewed as 0-1-sequences) where Y_2 is in $(10^{m2^m-1})^*$, i.e. marks the starting points of the binary numbers of length $m2^m$, and Z_2 is the sequence of these binary numbers. The definition of Y_2 and Z_2 can be done as before, once the distance $m2^m$ between two top row positions becomes definable. This, however, is possible by using the numbers coded in Z_1', given by induction hypothesis, as *addresses* of positions in Y_2, Z_2. For instance, consider two positions u, v where the Y_1'-bit is 1. They have distance $m2^m$ if the block of m bits in Z_1' which starts at u coincides with the corresponding block of m bits starting at v and such that this block of length m does not occur inbetween, starting at a Y_1'-position. The equality of blocks of length m in turn can be described as explained in the preceding paragraph. □

As a consequence of the theorem we obtain that over (acyclic) graphs in general, the hierarchy of Σ_k-definable sets is *strict*. (For grids, a strictness proof has recently been announced by Nicole Schweikardt, Mainz.)

The theorem above shows that the gap between automata over graphs (equivalent to the Σ_1-fragment of monadic second-order logic) and full monadic second-order logic is large, when the input structures are grids, pictures, or more complicated graphs. The hierarchy theorem sharpens the classical results on limits of Σ_1-definability, originating in Fagin's work (see [Fag75], [FSV95]). There it was shown that connectivity is a monadic graph property which is not Σ_1-definable.

In [Fag74] Fagin had shown that the Σ_1-fragment of *unrestricted* second-order logic (where second-order quantifiers range over relations, rather than sets), characterizes NP; as a consequence the n-th level of the polynomial time hierarchy is characterized by the Σ_n-fragment of unrestricted second-order logic. So the Σ_n-hierarchy of monadic logic is the "monadic analogue" of the polynomial time hierarchy.

A closer analysis of the hierarchy proof above shows, however, that the relation between the polynomial time hierarchy and its monadic version is very loose. The defining formulas for the witness languages $L[f_k]$ as used above can all be written as formulas $\exists X_1 \ldots \exists X_n \varphi(X_1, \ldots, X_n)$ where φ belongs to the extension of first-order logic by the transitive closure operator (see [EF95] for definitions). As a consequence, all the sets $L[f_k]$ belong to NP, the first level of the polynomial time hierarchy. In fact, even for the non-elementary function $f : m \mapsto f_m(m)$ the set $L[f]$ is in NP. On the other hand, recent work of Ajtai, Fagin, and Stockmeyer shows that for each level of the polynomial time hierarchy some complete set exists which is definable in monadic second-order logic. So the monadic alternation hierarchy result seems to be far away from the open problem whether the polynomial time hierarchy is infinite.

8 Concluding Remarks

The emphasis of this paper was to present a general approach to "recognizable" sets of labelled partial orders by means of a model of graph acceptor and some variants of it, and to show over the domain of pictures and grids that central statements of classical automata theory fail.

It is interesting to analyze the situation for other classes of labelled partial orders. A rough distinction of such classes may be done in three categories: (1) Classes where the central facts on word automata and tree automata are preserved, (2) classes which are "opposite", such as pictures and grids, where neither closure under complement holds nor the emptiness problem is decidable, and (3) classes with a "mixed situation".

A natural case of the first category is given by the Mazurkiewicz traces, viewed as partial orders (presented as dependency graphs). Several chapters of [DR95] develop this theory of recognizability. The trace dependency graphs exhaust rather well the range of the first category; it seems that by any substantial generalization of trace dependency graphs one leaves the framework of classical automata theory.

A candidate for the second category is, besides pictures and grids, the class of "mirror-concatenated trees" ([Th96a]); they are obtained from two labelled ordered trees with identical numbers of leaves by identifying the frontiers (which is done order preserving) and by reversing the edge direction in one tree (so that the roots of the two trees give a smallest and a greatest element in the resulting partial order). Surprisingly, these simple partial orders lead to recognizable sets with undecidable emptiness problem. (Namely, for any pair G_1, G_2 of context-free grammars, the emptiness of the intersection $L(G_1) \cap L(G_2)$ can

be checked by forming the set $S(G_1, G_2)$ of mirror-concatenated derivation trees from G_1, G_2, which have a common frontier word, and by testing for the emptiness of $S(G_1, G_2)$. The set $S(G_1, G_2)$ is recognizable by a graph acceptor, constructible from G_1, G_2.) We conjecture that over this domain also the closure under complementation fails for recognizable sets.

A "mixed situation" (as in the third category above) occurs over directed acyclic graphs of *bounded tree-width*. A graph is of tree-width k if there is a partition of its edge set into "clusters" (also called tree decomposition) and an undirected edge relation R on the collection of clusters such that three properties are satisfied: the clusters together with R define an undirected tree t, each cluster contains at most k vertices, and the clusters in which a given vertex v occurs form a connected subset of the tree t. Over graphs of bounded tree-width, the emptiness problem for monadic second-order properties is decidable ([Cou89], [See92]). On the other hand (as ongoing work of I. Schiering shows), existential monadic second-order sets of graphs of bounded tree-width need not be closed under complement; the complementation property can be saved, however, when the partition of the tree decomposition has only clusters which form connected subsets of the given graph.

There are further directions of work which have not been touched in this paper and which deserve more study. Already in the introduction we mentioned the subject of monadic second-order properties of infinite partial orders. Another track is the description of properties by calculi of regular expressions (as pursued in [BDW95]), or by algebraic notions of recognizability (as developed in Courcelle's work [Cou90],[Cou96]). Finally, for applications in decision problems of logic or in program verification the complexity of the transformation procedures from logical formulas to finite-state acceptors need to be analyzed.

References

[BDW95] F. Bossut, M. Dauchet, B. Warin, A Kleene Theorem for a class of planar acyclic graphs, *Inform. and Comput.* **117** (1995), 251-265.

[Bü62] J.R. Büchi, On a decision method in restricted second-order arithmetic, in: *Proc. 1960 Int. Congr. for Logic, Methodology, and Philosophy of Science*, Stanford Univ. Press, Stanford 1962, pp. 1-11.

[Cou89] B. Courcelle, The monadic second-order theory of graphs II: Infinite graphs of bounded width, *Math. Syst. Theory* **21** (1989), 187-221.

[Cou90] B. Courcelle, The monadic second-order logic of graphs I: recognizable sets of finite graphs *Inform. and Comput.* **85** (1990), 12-75.

[Cou96] B. Courcelle, The expression of graph properties and graph transformations in monadic secodn-order logic, in: *Handbook of Graph Transformations, Vol. I: Foundations* (G. Rozenberg, Ed.), World Scientific, Singapore 1996.

[Do70] J. Doner, Tree acceptors and some of their applications, *J. Comput. System Sci.* **4** (1970), 406-451.

[DR95] V. Diekert, G. Rozenberg (Eds.), *The Book of Traces*, World Scientific, Singapore 1995.

[EF95] H.D. Ebbinghaus, J. Flum, *Finite Model Theory*, Springer-Verlag, New York 1995.

[Fag74] R. Fagin, Generalized first-order spectra and polynomial-time recognizable sets, in: *Complexity and Computation* (R. Karp, Ed.), *SIAM-AMS Proceedings* **7** (1974), pp. 43-73.

[Fag75] R. Fagin, Monadic generalized spectra, Z. math. Logik u. Grundl. Math. 21 (1975), 123-134.

[FSV95] R. Fagin, L.J. Stockmeyer, M.Y. Vardi, On monadic NP versus monadic co-NP, *Information and Computation* **120** (1995), 78-92.

[GR96] D. Giammarresi, A. Restivo, Two-dimensional languages, in: *Handbook of Formal Language Theory*, Vol. III (G. Rozenberg, A. Salomaa, Eds.), Springer-Verlag, New York (to appear).

[GRST96] D. Giammarresi, A. Restivo, S. Seibert, W. Thomas, Monadic second-order logic over rectangular pictures and recognizability by tiling systems, *Information and Computation* **125** (1996), 32-45.

[GS84] F. Gécseg, M. Steinby, *Tree Automata*, Akadémiai Kiodó, Budapest 1984.

[IN77] K. Inoue, A. Nakamura, Some properties of two-dimensional tessellation acceptors, *Inform. Sci.* **13** (1977), 95-121.

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

[LS96] M. Latteux, D. Simplot, Context-sensitive languages and recognizable picture languages, Tech. Rep. IT-96-298, L.I.F.L., University of Lille 1.

[MT96] O. Matz, W. Thomas, The monadic quantifier alternation hierarchy over graphs is infinite, manuscript, Univ. of Kiel, 1996 (submitted).

[PST94] A. Potthoff, S. Seibert, W. Thomas, Nondeterminism versus determinism of finite automata over directed acyclic graphs, *Bull. Belg. Math. Soc. Simon Stevin* **1** (1994), 285-298.

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

[See92] D. Seese, Interpretability and tree automata: a simple way to solve algorithmic problems on graphs closely related to trees, in: *Tree Automata and Languages* (M. Nivat, A. Podelski, Eds.), Elsevier, 1992, pp. 83-114.

[Sp85] H. Sperber, *Idealautomaten*, Dissertation, Univ. Erlangen-Nürnberg, 1985.

[Sp86] H. Sperber, Finite automata accepting animals and lower sets, in: *Sém. Lotharingien et Combinatoire* (G. Nicoletti, ed.), Publ. Inst. Rech. Math. Avancée 316/S-13 (1986), Univ. Strasbourg, Dép. de Math., pp. 107-112.

[Th91] W. Thomas, On logics, tilings, and automata, in: *Automata, Languages, and Programming* (J. Leach et al., Eds.), Lecture Notes in Computer Science **510**, Springer-Verlag, Berlin 1991, pp. 441-453.

[Th96] W. Thomas, Languages, automata and logic, in: *Handbook of Formal Language Theory*, Vol. III (G. Rozenberg, A. Salomaa, Eds.), Springer-Verlag, New York (to appear).

[Th96a] W. Thomas, Elements of an automata theory over partial orders, in: *Proc. Workshop on Partial Order Methods in Verification* (D. Peled, Ed.), DIMACS Ser. in Discr. Math. (to appear).

[TW68] J.W. Thatcher, J.B. Wright, Generalized finite automata with an application to a decision problem of second order logic, *Math. Syst. Theory* **2** (1968), 57-82.