DIMACS Series in Discrete Mathematics
and Theoretical Computer Science
Volume 00, 19xx

Elements of an Automata Theory Over Partial Orders

Wolfgang Thomas

Abstract. A model of nondeterministic finite automaton over (finite) partial orders is introduced. It captures existential monadic second-order logic in expressive power and generalizes classical word automata and tree automata. Special forms, such as deterministic automata, are discussed, and logical and algorithmic properties are analyzed, like closure under complement and decidability of the nonemptiness problem. These questions are studied in the context of different classes of partial orders, such as trees, Mazurkiewicz traces, or rectangular grids.

1. Introduction

While automata over strings and trees are a well-known, widely used, and robust model, with many applications in the specification and verification of concurrent programs, the area of "finite automata over partial orders" cannot be called an established subject, despite the fact that partial orders are a natural domain for the study of concurrency. A possible reason for this is that many properties of finite automata which are essential in logical or algorithmic applications fail to hold when partial orders are considered as inputs (instead of strings or trees). Such properties are: equivalence between the deterministic and the nondeterministic model, closure under operations like complementation or projection, characterization by natural logical systems (like monadic second-order logic), and decidability of the nonemptiness problem (in logical terms: satisfiability problem). A possible remedy in this situation is to confine oneself to a narrower view of partial orders, for instance by extracting only sets of paths from partial orders, which brings back the framework of classical formal language theory.

In the present paper we stay with proper partial orders as inputs of automata and try to set up connections between such generalized automata and logical systems. We suggest a model of finite automaton which keeps the basic intuitive idea of nondeterministic automata on words: It is a device which scans "local neighbourhoods" in a given partial order while (nondeterministically) assigning states to the points of this partial order. We show that the details of this idea can be fixed in such a way as to allow a clear connection to logical descriptions: A set of (finite and labelled) partial orders is recognizable by such a finite automaton iff it is

1991 *Mathematics Subject Classification.* Primary 54C40, 14E20; Secondary 46E25, 20C20.

definable in existential monadic second-order logic (i.e., by a sentence which begins with a prefix of existential set quantifiers, followed by a first-order formula). If the structures under consideration are even linearly ordered (i.e., words) or if they are labelled trees, this result can be sharpened to the well-known equivalence between automata and (full) monadic second-order logic. So, regarding automata theory in a general context, existential monadic second-order logic can be considered as more basic than unrestricted monadic second-order logic.

In the automata theoretic view, where the notion of "local neighbourhood" is essential, it is useful to identify a (discrete) partial order $\leq$ with an acyclic directed 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 and v are distinct, $u \leq v$, and there is no w with $u < w < v$.)

We shall confine ourselves to finite acyclic graphs of this form in the present paper. While the basic ideas are easily transferred also to infinite structures, some additional difficulties arise in connection with logic, namely the choice of appropriate acceptance conditions in automata. It is (as yet) not clear whether simple acceptance conditions exist which lead to a characterization of interesting logical systems (as, for example, the model of tree automaton with the Rabin acceptance condition of [**Rab69**] characterizes monadic second-order logic over infinite labelled binary trees).

As it turns out, the properties of automaton definable sets depend on the particular class of partial orders (or acyclic graphs) which are allowed as inputs. Special cases of such classes are: words, trees, Mazurkiewicz trace graphs, and labelled rectangular grids. We investigate two basic questions: Are the automaton recognizable sets closed under complement? When is the nonemptiness problem decidable?

The paper is structured as follows: In the subsequent two sections we introduce the necessary terminology concerning partial orders and acyclic graphs, as well as the logical systems of first-order logic and monadic second-order logic. Some easy propositions are listed which illustrate the expressive power of these logics. In a section on first-order logic we present the key theorem which supplies a bridge to automata theory. It is a classical result of first-order model theory, due to Hanf [**Hnf65**], but not well-known in the community of theoretical computer science. Automata over acyclic graphs are introduced in Section 5. Some special forms are presented, and classes of partial orders are singled out over which these special forms are no restriction (i.e., normal forms of automata). In Section 6 we analyze the possibility of showing complementation lemmas and study the nonemptiness problem. The concluding section offers some directions for further research.

The approach adopted in this paper is based on ideas of [**Th91**]. Further results which serve as background have been shown in [**GRST96**] (mostly concerning labelled rectangular grids) and [**PST94**] (concerning general acyclic graphs). We cannot provide full proofs in this short communication, but try to give enough information to enable the reader to supply the details.

2. Partial Orders and Acyclic Graphs

As indicated in the introduction, we consider partial orders in the form of acyclic vertex-labelled and edge-labelled directed graphs. Usually we take A as label alphabet for vertices and B as label alphabet for edges (both alphabets are finite). As a relational structure, a graph is thus presented in the form

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

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 $E = \bigcup_{b\in B} E_b$. Thus, 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 always assumed to be acyclic (which means that no nonempty path exists from a vertex v back to v). Hence one obtains a partial order when forming the reflexive transitive closure E^* of the edge set E. We shall also assume that E is given as minimal edge relation generating a partial order; this means we exclude the existence of an edge (u, v) in the presence of a vertex w with nonempty paths from u to w and from w to v. A vertex u is called *root* of a partial order $\leq$ if $u \leq v$ for all vertices v; in the dual case (when $v \leq u$ holds for all vertices v) we speak of a *co-root*.

A special case of edge labelling is called *indexing*, namely when the label alphabet is a set $\{1, \ldots , k\}$ and either the out-edges of each vertex are numbered by $1, \ldots , i$ for some $i \leq k$, or the corresponding holds for the ingoing edges of each vertex. (We shall speak of out-edge indexing, respectively in-edge indexing.)

Let us consider the possibility of accepting such graphs by finite-state devices. We follow the intuitive idea that acceptance is based on a scanning process which checks all "local neighbourhoods" in the graph G under consideration. This scanning process should associate (generally in a nondeterministic way) states from a finite state-set Q to the vertices of G. Here, a minimal version of neighbourhood is given by a vertex together with its incoming and outgoing edges and their source vertices, respectively target vertices. If the acceptor (or graph automaton) is honestly finite, it can distinguish only a fixed number of different local neighbourhoods. In order to match this assumption on finite-state acceptors, we allow only graphs of bounded degree in a recognizable or definable set, i.e., graphs where for each vertex v the number of vertices u with $(u, v) \in E$ or $(v, u) \in E$ is bounded by a predefined constant d. If such a bound is dropped, non-isomorphic neighbourhoods will be confused. This more general case could also be handled in the framework to be developed below, but it adds complications and distracts from the essential points.

Let us list some basic classes of graphs and associated partial orders which fall under these conventions.

- *Words over an alphabet* A: These are (in our case nonempty) structures $(\{1, \ldots , n\}, (P_a)_{a\in A}, E)$ where n is the length of the word, $1, \ldots , n$ are the letter positions, P_a collects the positions carrying letter a, and E is the successor relation on $\{1, \ldots , n\}$.
- *Ordered labelled trees*: Taking the case of binary trees as a typical example, these are graphs of the form $(V, (P_a)_{a\in A}, E_1, E_2)$, where V is the set of tree nodes, the sets P_a are used as for words, and E_1, E_2 are the two relations of "first successor" and "second successor", respectively. In the usual way, this numbering of the successors induces a "left-to-right ordering" on the set of leaves.
- *Dependency graphs of Mazurkiewicz traces* (cf. [**DR95**]): Here the alphabet A is given together with a reflexive and symmetric dependency relation $D \subseteq A \times A$. The format of dependency graphs is the same as for words,

however E does not necessarily generate a linear order but just a partial one: The edge relation E respects D in the sense that edges connect only vertices with dependent letters and that any two vertices labelled by dependent letters are connected by a path. By reflexivity of D, the size of antichains in dependency graphs (subsets consisting of pairwise unrelated vertices in the associated partial order) is bounded by the size of the alphabet; we say that dependency graphs have *bounded antichains*.

- *Rectangular grids* ("two-dimensional words", "pictures", cf. [**GRST96**]): In this case, the vertices are arranged in a two-dimensional array, connected by a horizontal successor relation E_1 ("to the right") and a vertical successor relation E_2 ("downwards"). Thus the signature coincides with that of binary trees.
- *Mirror tree concatenations*: These are obtained by concatenating tree structures $t_1, s_1, t_2, s_2, \ldots, t_k, s_k$ in the following way (we just consider the case of binary trees): Each t_i is a binary tree as above, each s_i is obtained from a binary tree (with the same number of leaves as in t_i) by inverting the edge directions (which makes leaves into "sources" and the root into a "target"), and concatenation is carried out by identifying the leaves of t_i (left to right) with the sources of s_i (right to left), and identifying the target of s_i with the root of t_{i+1}.
- *(Acyclic) graphs of bounded tree-width* k (cf. e.g. [**Cou89**], [**See92**]): These graphs are associated to trees by the following condition: There is a covering of the vertex set by a collection of vertex sets (called "clusters" here), on which an undirected edge relation R exists such that
  1. for each graph edge (u, v) there is a cluster containing u and v,
  2. the clusters together with R define an undirected tree t,
  3. each cluster C contains at most k vertices,
  4. the clusters in which a given vertex v occurs form a connected subset of the tree t.

In the order of the list above, we denote the respective classes of acyclic graphs by *Words*, *Trees*, *Traces*, *Grids*, *MTreeC*, *BTWGraphs*.

3. Basic Logics

In the sequel, words, trees, traces, grids, and, in general, acyclic graphs, are considered as relational structures of the forms above. This allows to introduce logical definability notions in a uniform way. Here we do this in the framework of monadic second-order logic. Over graphs with the label alphabets A (for vertices) and B (for edges), 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. 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 will be omitted (as done also above) when no confusion arises.

Let $\mathcal{K}$ be a class of (acyclic) 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 (short: MSO-definable) if some sentence φ with $L = L(\varphi)$ exists.

The significance of monadic second-order logic (MSO-logic) for automata theory rests on the following classical result for the class *Words*:

THEOREM 3.1. (Büchi [**Bü60**], Elgot [**Elg61**])
A language $L \subset A^+$ is recognizable by a finite automaton iff it is MSO-definable.

PROOF. The idea for the step from automata to MSO-formulas is to introduce, for any state q_i of the given automaton, a set variable X_i for the set of those positions in a word where state q_i is assumed in a run. One formalizes the existence of an accepting run of an automaton with n states $q_0, \ldots, q_{n-1}$ over a word w by saying that there are sets $X_0, \ldots, X_{n-1}$ such that the first letter position belongs to X_0 (assuming q_0 is the initial state), each successor step is compatible with the transition relation of the automaton, and from the state on the last position, one reaches by the last letter a final state. Note that the first and last position are definable by the formulas $\neg\exists y E(y,x)$ and $\neg\exists y E(x,y)$, respectively. The resulting formula is an existential monadic second-order formula, short an EMSO-formula.

The converse direction, from MSO-formulas to automata, is based on standard closure properties of automaton recognizable languages, namely closure under union and complement (which captures propositional logic) and projection (which captures the existential quantifier). For a more detailed proof see e.g. [**Th96**]. $\square$

By applying the second and the first part of the proof in succession, one obtains that an MSO-formula (over word graphs) can be rewritten as an EMSO-formula.

The basis of the proof above is the equivalence between nondeterministic and deterministic finite automata: Nondeterminism serves to show closure of recognizable sets under projection, determinism shows closure under complement. The reduction to deterministic automata was shown also for finite automata over trees (using the "frontier-to-root mode" in tree automata, cf.[**GS84**]), whence an analogue of the theorem above holds also for the class *Trees*, including the reduction of MSO-logic to EMSO-logic. Without treating definitions in detail, let us also mention that over *Traces* a similar development is possible, now invoking Zielonka's construction of deterministic asynchronous automata ([**Zi87**]).

Let us introduce further subsystems of MSO-logic, including first-order logic with different signatures.

In the traditional classification of second-order formulas, the EMSO-formulas are also called monadic Σ_1^1-formulas. The dual formulas, where a prefix of universal set quantifiers precedes a first-order kernel, are called monadic Π_1^1-formulas. The corresponding properties (defined by such formulas) are called monadic Σ_1^1-properties, respectively monadic Π_1^1-properties. A property which is both monadic-Σ_1^1 and monadic-Π_1^1 is called a monadic Δ_1^1-property. In short we speak of monΣ_1^1-, monΠ^1_1-, and monΔ^1_1-properties. By $(\mathrm{mon}\Delta^1_1)_{Words}$ we denote the class of word properties (or: word languages) which are monΔ^1_1-definable; similarly for the other definability notions.

As an example, consider a monadic Σ^1_1-sentence which says that a successful run of a finite automaton over a word exists (see the proof above). For a *deterministic* finite automaton this sentence can also be written as a monadic Π^1_1-sentence, namely as saying: "All state sequences which start in the initial state and which for any two succeeding positions are compatible with the transition relation, have a state on the last letter position from which (by the last letter) a final state is reached." Since finite automata on words can be made deterministic, we thus have the following equalities:

PROPOSITION 3.2.

$$(\mathrm{mon}\Delta^1_1)_{Words} = (\mathrm{mon}\Sigma^1_1)_{Words} = (\mathrm{mon}\Pi^1_1)_{Words} = \mathrm{MSO}_{Words}.$$

The same is true over *Trees*.

First-order logic, short FO-logic (over acyclic graphs) is obtained from MSO-logic as above by dropping set quantifications. Typical quantifications in this logic are of the form $\exists y(E_b(x,y) \wedge \varphi(y))$ and $\forall y(E_b(x,y) \rightarrow \varphi(y))$, which express "there is a b-successor of x satisfying φ", respectively "all b-successors of x satisfy φ". Thus FO-logic includes standard process logics, such as the finitary version of "Hennessy-Milner-logic" (cf. [**Mil90**]).

It is well-known that in first-order logic the transitive closure of a given relation is (in general) not definable: In particular, in acyclic graphs the associated partial order is not definable. (A proof will be given in the next section.) Thus we obtain a stronger system of "first-order logic with $\leq$" when to FO-logic as above a symbol $\leq$ for the reflexive transitive closure of the edge relation E is added. We denote this system by FO[$\leq$]-logic. Typically, it allows to express properties of linear or partial orders which are formalizable in systems of propositional temporal logic. Over grids, we obtain an expressively equivalent variant of FO[$\leq$]-logic when for the two edge relations E_1 and E_2 the respective reflexive transitive closures $\leq_1$ and $\leq_2$ are introduced instead of $\leq$. Note that we have $x \leq y$ iff $x \leq_1 z$ and $z \leq_2 y$ for some z. Conversely, each relation $\leq_i$ is first-order definable in terms of the relation E_i and $\leq$: We have $x \leq_i y$ iff $x \leq y$ and (in case x and y are distinct) any z with $x < z \leq y$ is E_i-successor of some z' with $x \leq z' \leq y$.

For a class $\mathcal{K}$ of acyclic graphs, any of the above notions of definability induces a corresponding class of definable graph sets. We denote this class by the logical system with an index for the class $\mathcal{K}$, in the form $\mathrm{FO}_{\mathcal{K}}$, $\mathrm{FO}[\leq]_{\mathcal{K}}$, $(\mathrm{mon}\Sigma^1_1)_{\mathcal{K}}$ ($=$ $\mathrm{EMSO}_{\mathcal{K}}$), etc.

The following statement is trivial.

PROPOSITION 3.3. *For any class $\mathcal{K}$ of acyclic graphs, we have*

$$\mathrm{FO}_{\mathcal{K}} \ \subseteq \ (\mathrm{mon}\Delta^1_1)_{\mathcal{K}} \ \subseteq \ (\mathrm{mon}\Sigma^1_1)_{\mathcal{K}} \ \subseteq \ \mathrm{MSO}_{\mathcal{K}}.$$

Over *Words* and *Trees*, FO[$\leq$]-logic can be placed between FO-logic and EMSO-logic: One notes that $x \leq y$ is defined by the MSO-formula

$$\forall X(X(x) \wedge \forall z \forall z'((X(z) \wedge E(z,z')) \rightarrow X(z')) \rightarrow X(y)),$$

whence the claim follows by the expressive equivalence of EMSO-logic and MSO-logic over *Words*, respectively *Trees*. In fact, we have a sharper result, establishing the following proper inclusions (indicated by "$\subset$") and equalities:

PROPOSITION 3.4.

$$\mathrm{FO}_{Words} \subset \mathrm{FO}[\leq]_{Words} \subset (\mathrm{mon}\Delta^1_1)_{Words} = (\mathrm{mon}\Sigma^1_1)_{Words} = \mathrm{MSO}_{Words}.$$

PROOF. (Hint.) The language $a^*ba^*ca^*$ is an example of a word set which is definable in FO[$\leq$]-logic by the sentence

$$\exists x \exists y (P_b(x) \wedge x < y \wedge P_c(y) \wedge \forall z (\neg(z = x \vee z = y) \rightarrow P_a(z)))$$

but not in FO-logic (see next section). The next proper inclusion is exemplified by the set of words of even length. It is definable by a monadic Σ^1_1-sentence requiring a set X of positions which contains the first letter position, then every second position (i.e. satisfying $\forall z \forall z' (E(z,z') \rightarrow (X(z) \leftrightarrow \neg X(z')))$, and does not contain the last position. An equivalent Π^1_1-sentence says that all sets which contain the first position and then every second position do not contain the last position. An application of the Ehrenfeucht-Fraissé method shows that the word property of having even length is not expressible in first-order logic with linear ordering (cf. e.g. [**EF95**], [**Th96**]). The last two equalities are clear from Proposition 3.2. □

In Section 6 we shall see that over *Grids*, FO[$\leq$]-logic and EMSO-logic (or $(\mathrm{mon}\Sigma^1_1)$-logic) are incompatible in expressive power, and that the last two equalities of Proposition 3.4 turn into strict inclusions.

4. Hanf's Theorem

In [**Hnf65**], Hanf showed that in the first-order language of graphs only "local properties" can be specified. A property is local if it depends only on the occurrence (or non-occurrence) of certain local neighbourhoods around vertices. More precisely, 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.

It is easy to write down a sentence $\varphi_{\tau,\geq n}$ which says that there are at least n different occurrences of spheres of a given isomorphism type τ. Using conjunctions of such sentences and negations of such sentences, one can specify for finitely many types $\tau_1, \ldots, \tau_m$ that the occurrence number of τ_i is $\leq n_i$, or $< n_i$, or $= n_i$. A graph language L defined by a disjunction of such conditions (or equivalently: by a boolean combination of sentences $\varphi_{\tau,\geq n}$) is called *locally threshold testable*.

Equivalently, L is representable in terms of a certain equivalence relation $\sim_{r,t}$ between graphs. Define $G \sim_{r,t} G'$ to hold if for all types τ of r-spheres, the occurrence numbers of τ in G and G' are both $\geq t$ or else coincide. Over graphs of bounded degree, $\sim_{r,t}$ is an equivalence relation of finite index. An easy exercise shows that a set L is locally threshold testable iff L is a union of $\sim_{r,t}$-classes for some radius r and threshold number t.

The main result in the first-order model theory of graphs says that the above mentioned conditions on occurrence numbers already exhaust the expressive power of first-order logic:

THEOREM 4.1. (essentially Hanf [**Hnf65**])
A first-order definable set of graphs (of bounded degree) is locally threshold testable.

In particular, a first-order sentence is equivalent to a boolean combination of sentences of the form "there are $\geq n$ occurrences of r-spheres of type τ".

The proof rests on an application of the Ehrenfeucht-Fraissé-game. We refer the reader to [**EF95**], [**FSV95**], or [**Th96**] for details.

Let us sketch three applications. First, we verify that the language $L = a^*ba^*ca^*$ is not in FO_{Words}. Otherwise, we would obtain a contradiction: From an assumed FO-sentence defining L we would obtain r and t such that two words (word models) which are $\sim_{r,t}$-equivalent are both in L or both not in L. But it is easily seen that for sufficiently large n the words $a^n ba^n ca^n (\in L)$ and $a^n ca^n ba^n (\notin L)$ have the the same occurrence numbers of r-spheres counted up to threshold t and thus are $\sim_{r,t}$-equivalent.

In a similar way, it is shown in the domain *Grids* that the set of all square grids (of size $n \times n$ for $n \geq 1$, whose vertices are all labelled with a) is not first-order definable.

Finally, as a preparation to the next section, we note the following consequence of Hanf's Theorem:

PROPOSITION 4.2. *The class* $\mathrm{EMSO}_{\mathcal{K}}$ *coincides with the class of projections of locally threshold testable languages* $L \subseteq \mathcal{K}$.

PROOF. As a preparation, consider a graph G with vertex labels in A. An expansion $(G, V_1, \ldots, V_m)$ by designated vertex sets V_i, which allows to interpret a formula $\varphi(X_1, \ldots, X_m)$, can be represented as a graph H with vertex labels in $A \times \{0,1\}^m$: The i-th additional component has value 1 for vertex v iff $v \in V_i$.

Now a graph G satisfies a sentence $\exists X_1 \ldots \exists X_m \varphi(X_1, \ldots X_m)$ (with first-order formula φ) iff some graph H, which arises from G by expanding the vertex labels from A to $A \times \{0,1\}^m$, satisfies $\varphi(X_1, \ldots, X_m)$. But this is equivalent to the existence of a graph H in $L(\varphi)$ (which by Hanf's Theorem is a locally threshold testable language) such that $h(H) = G$ for the projection $h : A \times \{0,1\}^m \to A$. $\square$

Hanf's Theorem connects first-order logic to local properties and is thus a good starting point for a logically motivated automata theory over graphs.

5. Finite-State Acceptors and Special Forms

We introduce graph acceptors which capture projections of locally threshold testable sets:

A *graph acceptor* over the 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,
- Occ is a boolean combination of conditions "there are $\geq n$ occurrences of spheres of type τ" (where τ is an r-sphere type over the label alphabets $A \times Q$ and B).

We call Δ the set of *transitions* and Occ 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 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 occurrences of these spheres are compatible with the constraint Occ. We call this covering of G an "accepting tiling" of G and sometimes speak of transitions as "tiles" and graph acceptors as "tiling systems" (cf. [**Th91**]).

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 $\mathcal{A}$.

By Proposition 4.2, graph acceptors characterize existential monadic second-order logic:

PROPOSITION 5.1. *For any class $\mathcal{K}$ of graphs of bounded degree, a language $L \subseteq \mathcal{K}$ is recognizable iff $L \in \mathrm{EMSO}_{\mathcal{K}}$.*

Similarly, a language L is recognizable by a graph acceptor with only one state iff L is first-order definable.

Usual finite automata over words or trees are simulated by special graph acceptors, in which only 1-spheres are used as transitions and the occurrence constraints are cancelled. The use of initial and final states in the classical model is captured by the use of 1-spheres whose designated center has no predecessor, respectively no successor; such transitions can only be used at the beginning, respectively at the end of a word.

In comparison with classical automata, two features of graph automata seem complicated: the use of r-spheres for $r > 1$, and the use of occurrence constraints. We shall see that both features can be eliminated only with extra restrictions on the input graphs.

In order to see that over acyclic graphs in general the use of r-spheres in transitions can not be eliminated by resorting to 1-spheres only, we consider the following example, suggested by S. Seibert.

PROPOSITION 5.2. *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. It is easy to verify recognizability of L_n 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, there will be two occurrences of the same 1-sphere transition at corresponding positions on two superedges, not touching the ends of the superedges and unrelated in the partial order of the supergrid. (One may choose two occurrences of the same 1-sphere transition at the central positions of two superedges in the same row or in the same column of a large enough n-supergrid.) Obtain a new graph by exchanging the targets of the outgoing edges in the two 1-spheres covered by these transitions. The new graph is still acyclic, accepted by $\mathcal{A}$, but not in L_n. $\square$

A similar idea appears in Example 3.2 of [**Th91**]; there it is shown that our graph acceptors are properly more expressive than the dag automata of Kamimura and Slutzki [**KS81**].

In contrast to the proposition above, one verifies that over the classes *Words*, *Trees*, *Traces*, and *Grids*, the use of 1-spheres is sufficient. (The reduction from

r-sphere transitions to 1-sphere transitions involves a blow-up in the number of states.) Moreover, in the domain *Grids* there is a variant of 1-spheres which may seem more natural: In the approach developed in [**GRST96**] over *Grids*, the transitions are just (2×2)-squares of four vertices and edges. In this model, where transitions have no designated center, the corners and borders of grids are no more detectable (i.e., tilable by special transitions only), and thus grids are presented with extra rows and columns of border markers #, also to be covered by transitions.

A precise description of the class of acyclic graphs where in graph acceptors the use of 1-sphere transitions suffices is not known.

Let us turn to the occurrence constraints. In general they can also not be eliminated: We consider the set of acyclic graphs G_n 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 and one to $v_{(i+1) \bmod n}$. One may 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. Now consider the graph language L consisting of such graphs where at least one u_i is labelled a and the remaining vertices (not labelled a) are labelled b. It is clear that by an occurrence constraint the existence of a vertex with label a can be guaranteed. 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 a, say at u_1. For sufficiently large n, there will be an accepting 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 a, 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 the classes *Words*, *Trees*, and *Grids*. 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. This motivates the following claim:

PROPOSITION 5.3. *Let $\mathcal{K}$ be a class of acyclic graphs which have indexed out-edges and a co-root (and hence are connected). Then a language $L \subseteq \mathcal{K}$ is recognizable iff it is recognizable by a graph acceptor without occurrence constraints. The same holds if the graphs in $\mathcal{K}$ have indexed in-edges and a root.*

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 indices of the out-edges serve to avoid double-counting: The accumulated occurrence numbers are transferred only along the outgoing edges with index 1. 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 carry index 1, and where furthermore 1 is added to n_i (to capture that the present type is τ_i). Finally, r-sphere transitions for the co-root (the 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 indexed in-edges and the existence of a root is analogous. □

It is clear that words, trees, and grids are subsumed by the preceding proposition. Formally, in the case of grids one has to modify the edge labels in order to have indexed out-edges: The vertices of the last column of a grid, which have no (horizontal) E_1-successors in the original convention, should now have vertical out-edges in E_1 (instead of E_2). The elimination of occurrence constraints over grids is treated in detail in [**GRST96**].

Finally, we turn to a special form of acceptor on partial orders which represents a proper restriction: *deterministic acceptors*. Partial orders are a useful assumption for introducing deterministic acceptors; there should be a uniqueness in the construction of runs when proceeding from smaller to greater vertices in the partial order. There seems to be no canonical definition of deterministic graph acceptors; and even over simple acyclic graphs like the rectangular grids there are several possibilities. We suggest here a "determinism by states" (rather than "determinism by transitions"). We call an acceptor (say with r-sphere transitions) over partial orders *deterministic* if for any r-sphere around a vertex v and any state assignment to the vertices $u < v$ in this r-sphere, the assignment of a state to v (by the available transitions) is unique. (Note that a certain "lookahead" is built into this definition because a sphere has to match a whole neighbourhood of the input graph.) So, the state assignment is unique *per se* for vertices which have no predecessors in the partial order. This definition is compatible with determinism over words and trees (using frontier-to-root tree automata, i.e., with the reversed partial order in trees). For a class $\mathcal{K}$ of acyclic graphs, denote by $\mathrm{Det}_{\mathcal{K}}$ the class of languages $L \subseteq \mathcal{K}$ which are recognized by deterministic graph acceptors.

An example of a language in Det_{Grids} is the set of square grids (trivially labelled by a throughout). The assignment of states can be arranged such that a special state is associated to the diagonal starting from the unique vertex without incoming edges (which we assume to be on the top left corner). The square property is verified when in transitions for other border positions this special state is allowed only for the vertex without any outgoing edges (at the bottom right corner).

Let us verify that determinism is a proper restriction. A well-known example is provided by the domain *Trees* when scanned in root-to-frontier mode (cf. [**GS84**]). But also over partial orders which have a co-root (where information of a run can be gathered in a single vertex) this phenomenon occurs:

Proposition 5.4. *There is a grid language which is recognizable by a graph acceptor but not by a deterministic graph acceptor.*

Proof. A suitable example is provided in [**PST94**]: Consider the set L of square grids which have label b everywhere except for two vertices labelled a on the right border and bottom border, in the same distance δ to the right-bottom corner. (Call this δ the "a-distance".) An appropriate nondeterministic graph acceptor guesses a point on the diagonal (from the top left to the bottom right corner), and from this point sends two "signals" (in the form of special states), one horizontally to the right, one vertically to the bottom. If at the two border points hit in this way letter a 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 b occurs is easily implemented.

Now suppose that a deterministic graph acceptor recognizing this grid language L exists. Invoking the construction of Proposition 5.3, we can assume that occurrence constraints are eliminated (note that the construction transforms deterministic graph acceptors again into deterministic ones). Suppose the acceptor has r-sphere transitions. Then the states of accepting runs on two grids from L of the same size are identical except for the last r rows and last r columns. The $(r+1)$-st last rows thus coincide except for the last r columns. Because there are only finitely many assignments of transitions to the last r positions of a row, there exist (for sufficiently large size of input squares) two squares $G_1, G_2 \in L$ of same size and with two different a-distances such that in the corresponding accepting runs also the last r transitions on the $(r+1)$-st last row coincide in G_1 and G_2. Then the last r rows from the accepting tiling of G_1 can be exchanged with the last r rows of the accepting tiling of G_2. Hence a grid outside the language L is accepted, a contradiction. $\square$

For deterministic acceptors over *Grids*, the reduction of r-spheres to 1-spheres is no more possible. A simple example is the set of computations of a Turing machine. Such computations are represented in a space-time diagram and hence in grid form. To check a labelled grid for being a computation of a given Turing machine, one can use a deterministic (single-state) acceptor using 2-sphere transitions, but not a deterministic acceptor with 1-sphere transitions.

Determinism corresponds to a restriction of EMSO-logic. As in the case of words (see Proposition 3.2), monadic Σ_1^1-definitions can be put into Π_1^1-form:

PROPOSITION 5.5. *If a language $L \subseteq \mathcal{K}$ of acyclic graphs is recognizable deterministically, then $L \in (\mathrm{mon}\Delta_1^1)_{\mathcal{K}}$.*

6. Some Results on Expressiveness and Decidability

In this section we come back to the question raised in the introduction: Over which classes of acyclic graphs (or generated partial orders) are the recognizable sets closed under complement (i.e., EMSO-logic is as expressive as MSO-logic), and when is the nonemptiness problem decidable? Whereas both questions are solved positively in the domains *Words*, *Trees*, *Traces*, let us see that this fails over *Grids*. In the statement below we also include the relation to deterministic recognizability and Δ_1^1-properties. At the same time, we settle the relation between EMSO-logic and FO[$\leq$]-logic over grids.

THEOREM 6.1. (a) *The following inclusion chain is proper:*

$$\mathrm{Det}_{Grids} \subset (\mathrm{mon}\Delta_1^1)_{Grids} \subset (\mathrm{mon}\Sigma_1^1)_{Grids} \subset \mathrm{MSO}_{Grids}$$

(b) *The classes $\mathrm{FO}[\leq]_{Grids}$ and $(\mathrm{mon}\Sigma_1^1)_{Grids}$ are incompatible with respect to inclusion.*

(c) *The nonemptiness problem of graph acceptors over grids is undecidable.*

PROOF. (a) The inclusions as such are clear from the preceding remarks. To verify that the first inclusion is strict, take the example set L of Proposition 5.4. To show that L is in $(\mathrm{mon}\Delta_1^1)_{Grids}$, it remains to supply a $(\mathrm{mon}\Pi_1^1)$-definition. Such a sentence can be constructed starting from the following condition: "For each set X of vertices consisting of (1) a prefix of the diagonal up to some vertex u, (2) the vertices to the right of u on the same row, ending with v, and (3) the vertices below u on the same column, ending with w, we have: if v is labelled with a, so is w."

For the strictness of the second inclusion, we identify a grid with its sequence of columns, regarding as column the associated sequence of vertex labels. Following [**GRST96**], we consider the set N of grids of the form GH where G and H are distinct square grids of the same size over the vertex label alphabet $\{a, b\}$. This set is monadic Σ_1^1, because the existence of a pair (x, y) of vertices (at corresponding positions in G and H) with distinct labels can be formulated using existential set quantifiers. (Namely, there should be a set X_1 containing all points on the same horizontal as x, and furthermore a set X_2 which occupies the diagonal, which starts at the topmost vertex above x, downward to the right. Now y is the unique point above the end of this diagonal which belongs to X_1.) In order to show that N is not monadic Π_1^1, it suffices to show that the set of grids GG, consisting of two identical square grids, is not monadic Σ_1^1. Here we use the characterization of monadic Σ_1^1, i.e. EMSO-logic, by graph acceptors with 1-sphere transitions and without occurrence constraints. Such a graph acceptor can transfer the information from the left square grid to the right square grid only via the two stripes of transitions along the border between the two half grids (of square form). For the given graph acceptor, the number of such stripes is $k^{(r \cdot n)}$ (for some fixed k and r) in the length n of the sides of squares. However the number of possible squares grows by the rate 2^{n^2}. Thus, for sufficiently large n we find distinct squares G and H of side length n such that on accepting tilings over GG and HH the stripes of 1-spheres right and left to the central border are identical. This implies that GH and HG are also accepted, a contradiction.

The set of grids GG where G is square shows that also the last inclusion of the claim is proper.

(b) The set of grids consisting of a single column of even length is (monΣ_1^1)-definable but not FO[$\leq$]-definable (see Proposition 3.4). In order to exhibit a grid language which is FO[$\leq$]-definable but not (monΣ_1^1)-definable (i.e., not recognizable), consider a variant of the set N above: the set M of grids of the form GCH where C is a column labelled by a special letter c and where the sets of different column words occurring in G and H (over the vertex label alphabet $\{a, b\}$) coincide. This set M is definable in FO[$\leq$]-logic, making use of the condition that for all positions x in the first row before the vertex labelled c, there is a position y in the first row after the vertex labelled c such that the columns associated to x and y coincide; similarly for each such y after the c-labelled vertex there is a corresponding x before the c-labelled vertex. The coincidence of the columns below x and y is easily formalizable with the relations $\leq_1$ and $\leq_2$, which in turn are definable in terms of $\leq$ (as shown in Section 3). The proof that M is not (monΣ_1^1)-definable is analogous to part (a) above, using the fact that for any constants k and r, the number of distinct sets of columns of length n exceeds $k^{(r \cdot n)}$ for sufficiently large n.

(c) We show that for any Turing machine $\mathcal{M}$ we can define a graph acceptor $\mathcal{A}_{\mathcal{M}}$ over an appropriate label alphabet which accepts some grid iff $\mathcal{M}$ halts when started on the empty tape. The idea is to let $\mathcal{A}$ accept just those grids which code a halting computation of $\mathcal{M}$ on the empty tape. Such a halting computation is finite in space and time (the two dimensions of the grid). Thus, the first line of such a grid is a sequence of blanks, 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-sphere transitions. That the grid is suffiently large to include all work cells of the computation is guaranteed by excluding transitions for border vertices which code work cells. Finally the last line should include a final state of $\mathcal{M}$. □

It should be noted that over *Words*, *Trees*, and *Traces* all classes of part (a) of the preceding theorem coincide (cf. Proposition 3.2).

An interesting problem is to find classes of partial orders beyond the domains *Words*, *Trees*, and *Traces*, over which EMSO-logic is closed under complement and/or where the nonemptiness problem for recognizable sets (satisfiability of EMSO-logic) is decidable. We discuss three classes: the partial orders with bounded antichains, the mirror tree concatenations, and the acyclic graphs of bounded tree-width.

Partial orders with bounded antichains constitute a generalization of trace graphs, in which the partial order is no more tied to a dependence structure of the vertex label alphabet. By a small modification of parts (a) and (c) of the preceding theorem, one verifies the following:

Proposition 6.2. *Over acyclic graphs with bounded antichains, EMSO-logic is not closed under complement, and the satisfiability problem for EMSO-sentences (and hence the nonemptiness problem for finite-state graph acceptors) is undecidable.*

Proof. We modify the grids of the preceding theorem (following an idea of I. Schiering): In the definition of the first successor relation (which proceeds horizontally from left to right), add an extra edge from the last vertex of each row (excluding the last two rows) to the first vertex of the second-next row, respectively. The resulting grid structure generates a partial order with antichains of at most two elements. One can now adapt the proofs of claims (a) and (c) above for these modified grids. □

For the class *MTreeC* if mirror tree concatenations we do not know whether a complementation result of EMSO-logic holds. However, it is easy to see that the nonemptiness problem for graph acceptors over the class *MTreeC* is undecidable: We use the undecidability of the nonemptiness problem for intersections of context-free languages. Given two context-free grammars G_1, G_2, one can construct a graph acceptor which accepts a pair (t, s) of mirror-concatenated trees iff t is a derivation tree for G_1, s is an inverted derivation tree for G_2, and the common sequence of leaves for t and s consists of terminal symbols only. Such a pair (t, s) exists iff G_1 and G_2 generate a common terminal word.

A better candidate domain for generalizing the classical closure and decidability results of automata theory seems to be the class of graphs of bounded tree-width. As shown by Courcelle [**Cou89**], the satisfiability of MSO-sentences over *BTWGraphs* is decidable. However, a reduction of MSO-logic to EMSO-logic (or equivalently: a complementation theorem for EMSO-logic) is unknown. In a restricted case, this reduction is possible ([**ST96**]), namely where a tree decomposition exists whose clusters are vertex sets which are connected by the symmetric closure of the graph edge relation.

7. Conclusion

In this paper, some suggestions were developed towards an automata theory over partial orders, and connections to various logical systems were established. We studied EMSO-logic and acceptors over several classes of finite partial orders and investigated the complementation problem and the nonemptiness problem for recognizable sets.

Some open questions have been mentioned already. Let us list some further directions which are unexplored.

(1) A theory of recognizable sets of infinite partial orders. Over which classes of infinite partial orders is it possible to introduce logically meaningful acceptance conditions, and what are these conditions? Over which classes is the nonemptiness problem decidable, possibly such that furthermore nonempty recognizable sets contain "regular" partial orders (where the meaning of "regular" is also open)?

(2) Complexity bounds for transformation algorithms and decision procedures. We did not discuss the complexity issue, e.g. in the conversion of formulas into automata or for the nonemptiness test. Note that already in the domain *Traces*, the available algorithms are of such a high complexity that a practical application seems hard.

(3) Development of other descriptive formalisms. Instead of systems of classical logic, more restrictive systems should be studied, whose expressive power might suffice for interesting applications but with acceptable complexity bounds e.g. for the satisfiability problem. These can be versions of regular expressions (cf. [**BDW95**]), or restrictions of EMSO-logic, or of FO[$\leq$]-logic, over partial orders.

(4) Comparison with the algebraic approach to recognizability. Here we refer to Courcelle's theory of recognizability, which is based on many-sorted and locally finite graph algebras (cf. [**Cou90**]). The class of recognizable graph sets in this setting is closed under boolean operations, and all MSO-definable sets turn out to be recognizable. Over *Grids*, recognizability in the algebraic sense is even strictly stronger than MSO-definability. It is open whether, for instance, the two approaches of recognizability (via tilings and via locally finite algebras) coincide for exactly those classes of partial orders where EMSO-logic is closed under complement.

8. Acknowledgment

I thank Oliver Matz, Ina Schiering, and Sebastian Seibert for their comments and suggestions, which led to corrections and improvements.

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.

[Bü60] J.R. Büchi, Weak second-order arithmetic and finite automata, *Z. Math. Logik Grundl. Math.* **6** (1960), 66-92.

[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.

[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.

[Elg61] C.C. Elgot, Decision problems of finite automata design and related arithmetics, *Trans. Amer. Math. Soc.* **98**, (1961), 21-52.

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

[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.

[Hnf65] W. Hanf, Model-theoretic methods in the study of elementary logic, in: *The Theory of Models* (J. Addison, L. Henkin, P. Suppes, Eds.), North-Holland, Amsterdam 1965, pp. 132-145.

[Mil90] R. Milner, Operational and algebraic semantics of concurrent processes, in: *Handbook of Theoretical Computer Science* (J. v. Leeuwen, Ed.), Vol. B, Elsevier Sci. Publ., Amsterdam 1990.

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

[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.

[Rab69] 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 Science Publishers, 1992, pp. 83-114.

[ST96] I. Schiering, W. Thomas, work in progress.

[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).

[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.

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

Institut für Informatik und Praktische Mathematik, Universität Kiel, D-24098 Kiel
E-mail address: `wt@informatik.uni-kiel.de`