Theoretical Computer Science 103 (1992) 143–159
Elsevier

Infinite trees and automaton-definable relations over ω-words

Wolfgang Thomas*

Institut für Informatik und Praktische Mathematik, Universität Kiel, Olshausenstr. 40, W-2300 Kiel, Germany

Abstract

Thomas, W., Infinite trees and automaton definable relations over ω-words, Theoretical Computer Science 103 (1992) 143–159.

We study relations over ω-words using a representation by tree languages. An ω-word over an alphabet with k letters is considered as a path through the k-ary tree, an n-tuple of ω-words as an n-tuple of paths (coded by an appropriate valuation of the k-ary tree using values in $\{0,1\}^n$), and a relation over ω-words as a tree language. In the first part of the paper we give a logical characterization of the "Rabin-recognizable relations" (whose associated tree languages are recognized by Rabin tree automata) in terms of "weak chain logic", a restriction of monadic second-order logic over trees. In the second part of the paper an extended logic is considered, obtained by adjoining the "equal-level predicate" over trees. We describe the class of relations over ω-words which (in the tree language representation) are definable in this logic, and show that the theory of the k-ary tree in this logic is decidable. It covers tree properties which are not expressible in the monadic second-order logic SkS.

1. Introduction

In this paper we consider sets of valued infinite trees where the valuation codes a tuple of infinite paths. Such "path-valued" trees are useful in two respects: first, a set of path-valued trees corresponds to a relation over ω-words, or ω-*relation* for short, and tree automata can be applied in the investigation of these relations. Secondly, the path-valued trees arise in the study of those logics over infinite trees where *quantifiers over paths* are used, for instance in branching time logics. We study the ω-relations whose associated tree sets are recognized by Rabin tree automata, characterize these ω-relations in terms of several logics, compare them with other relation classes defined

* Research partially supported by ESPRIT BRA No. 3166 ASMICS.

by sequential automata, and obtain new decidability results for logics with path quantifiers.

We give the basic definitions and a more detailed summary, assuming that the reader is familiar with automaton models over infinite words (nondeterministic Büchi automaton, deterministic Muller automaton) and over infinite trees (Rabin tree automaton). A *B-valued k-ary tree* is a map $t:\{0,\ldots,k-1\}^* \to B$; the node represented by $w \in \{0,\ldots,k-1\}^*$ carries value $t(w)$. Denote the set of B-valued k-ary trees by $T_k(B)$. Let A be an alphabet with k letters; without loss of generality, $A=\{0,\ldots,k-1\}$. We associate with any tuple $\bar{\alpha}=(\alpha_1,\ldots,\alpha_n)\in(A^\omega)^n$ a $\{0,1\}^n$-valued tree $t_{\bar{\alpha}}$ (which is a *path-valued tree*). Define $t_{\bar{\alpha}}:\{0,\ldots,k-1\}^* \to \{0,1\}^n$ by setting the ith component $(t_{\bar{\alpha}}(w))_i$ of $t_{\bar{\alpha}}(w)$ to be 1 iff w is a finite prefix of α_i. For an ω-relation $R\subseteq(A^\omega)^n$ let $T_R=\{t_{\bar{\alpha}}\in T_k(\{0,1\}^n)\,|\,\bar{\alpha}\in R\}$. An ω-relation $R\subseteq(A^\omega)^n$ is *Rabin-definable* iff the set T_R is recognized by a Rabin tree automaton (in the sense of [14], see also Section 2).

As shown in [14], a set of k-ary trees is recognized by a Rabin tree automaton iff it is defined by a formula of the monadic second-order theory SkS. SkS has function constants for the k successor functions in the k-ary tree, and variables and quantifiers for nodes and for sets of nodes.

Läuchli and Savioz [12] have investigated the Rabin-definable (or SkS-definable) relations over *finite* words (over A^*). The associated tree sets contain trees where the valuation codes a tuple of nodes instead of a tuple of paths. The main result of [12] states that such a tree set is definable in SkS iff it is definable in the *weak* monadic theory WSkS (where all set quantifiers are restricted to range only over finite sets). Läuchli and Savioz [12] attribute to Rabin the question whether a corresponding result holds for sets of path-valued trees.

A positive answer is given in Section 2 of this paper. Moreover, it is shown that for the description of a Rabin-recognizable set of path-valued trees it suffices to use quantifiers over finite *chains*, i.e. finite subsets of paths through the infinite k-ary tree. The proof combines a consideration of Rabin tree automata on trees $t_{\bar{\alpha}}$ with a reduction of the one-successor theory S1S to the weak theory WS1S. So, for definability of tree sets T_R (coding ω-relations R) the full monadic second-order theory SkS is equivalent to a small fragment, *weak chain logic*, where set quantification is restricted to finite sets which are totally ordered by the partial tree ordering. It follows that the "intermediate" system which allows quantification over arbitrary chains, called *chain logic*, is also equivalent to SkS for definition of sets T_R.

In Section 3 the Rabin-definable ω-relations are compared with other types of ω-relations, defined by different versions of sequential Büchi automata that work on tuples of ω-words. We consider the componentwise recognizable, the Büchi-definable, and the rational ω-relations. They form a strictly increasing hierarchy, and the class of Rabin-recognizable ω-relations is located properly between the first two levels. The unary relations in each of these classes yield precisely the class of regular ω-languages.

For the second half of the paper, the class of Büchi-definable ω-relations is of special interest. It is defined using the natural identification of an ω-relation $R\subseteq(A^\omega)^n$

with a sequence set $L_R \subseteq (A^n)^\omega$ given by

$$L_R = \{\beta \in (A^n)^\omega \mid \beta = (\alpha_1(0), \ldots, \alpha_n(0))(\alpha_1(1), \ldots, \alpha_n(1)) \ldots \text{ such that } (\alpha_1, \ldots, \alpha_n) \in R\}.$$

An ω-relation $R \subseteq (A^\omega)^n$ is *Büchi-definable* iff $L_R \subseteq (A^n)^\omega$ is regular, i.e. recognized by a Büchi automaton over the alphabet A^n. (In [17] the Büchi-definable ω-relations were called "sequential", reminding one of the "sequential calculus" of [2]. We use a different term here in order to avoid confusion with the sequential functions and transducers in the sense of [1]. In recent works of Frougny and Sakarovitch [6, 7] one finds also the term *letter-to-letter relation.*)

The subject of Sections 4 and 5 is an extension of chain logic over trees, obtained by adjoining the *equal-level predicate* E over A^*, *with* $(u, v) \in E$ iff $|u| = |v|$ for two words $u, v \in A^*$. We call this system *chain logic*$+E$. Two results are proved: first, the ω-relations which (in the tree language representation) are definable in this logic are shown to coincide with the Büchi-definable ω-relations. Second, the theory of the k-ary tree in the language of chain logic $+ E$ is shown to be decidable. In contrast, it is known that from the decidable theory SkS, or even the weak theory WSkS, one obtains (for $k \geqslant 2$) an undecidable theory when the predicate E is added. Thus, chain logic $+ E$ gives a decidable theory which allows one to treat tree properties that are not expressible in SkS.

Quantification "along paths" in trees, as provided by chain logic, suffices for many applications in logics of programs, since most systems of temporal or modal logic can be embedded in chain logic. The predicate E adds a feature which allows one to treat certain "uniformity conditions". We discuss this aspect in Section 5, concerning the model checking problem for finite-state programs.

At the end of the paper some directions for further work are indicated.

2. Rabin-definable ω-relations and chain logic

If $t: \{0, \ldots, k-1\}^* \to B$ is a B-valued $k-$ary tree, a *chain* through t is a subset of the domain $\{0, \ldots, k-1\}^*$ which is totally ordered by the prefix relation $<$. A *path* is a chain which is maximal w.r.t. set inclusion. If π is a path, $t|\pi$ denotes the restriction of the map t to π. Given a set $X \subseteq \{0, \ldots, k-1\}^*$, define its *hull* by

$$\text{hull}(X) = X \cup \{u \in \{0, \ldots, k-1\}^* \mid u = wi \text{ for some } w \in X,\ i < k\}.$$

A *Rabin tree automaton* (in the sense of [14]) over B-valued k-ary trees is of the form $\mathscr{A} = (Q, q_0, \Delta, \mathscr{F})$ with finite-state set Q, initial state $q_0 \in Q$, transition set $\Delta \subseteq Q \times B \times Q^k$ and a system $\mathscr{F} \subseteq 2^Q$ of final-state sets. A *run* of $\mathscr{A}$ on $t \in T_k(B)$ is a tree $r: \{0, \ldots, k-1\}^* \to Q$ such that $r(\varepsilon) = q_0$ and $(r(w), t(w), r(w0), \ldots, r(w(k-1))) \in \Delta$ for $w \in \{0, \ldots, k-1\}^*$. The run r is *successful* if for all paths π the set $\text{In}(r|\pi)$ of states which occur infinitely often in $r|\pi$ belongs to $\mathscr{F}$. A set $T \subseteq T_k(B)$ is *Rabin-recognizable* if for some Rabin tree automaton $\mathscr{A}$ the set T consists of the trees on which there is a successful run of $\mathscr{A}$. As in Section 1, we call an ω-relation R *Rabin-definable* iff the associated tree language T_R is Rabin-recognizable in this sense. The class of Rabin-definable ω-relations is denoted by ω-RBN.

We introduce the necessary logical terminology. A $\{0,1\}^n$-valued k-ary tree t will be identified with the model theoretic structure

$$\underline{t} = (\{0,\dots,k-1\}^*, \varepsilon, \cdot 0, \dots, \cdot (k-1), <, P_1, \dots, P_n),$$

where the first items define the unvalued k-ary tree

$$t_k := (\{0,\dots,k-1\}^*, \varepsilon, \cdot 0, \dots, \cdot (k-1), <),$$

with root ε (empty word), the k successor functions $\cdot 0, \dots, \cdot (k-1)$ on $\{0,\dots,k-1\}^*$ and the prefix relation $<$, and where the predicates P_i are given by

$$P_i = \{w \in \{0,\dots,k-1\}^* \mid (t(w))_i = 1\}.$$

The corresponding monadic second-order formalism *SkS* ("second-order theory of k successors") has variables $x, y, \dots$ and $X, Y, \dots$ for elements ("nodes") and subsets of $\{0,\dots,k-1\}^*$, respectively. The atomic formulas are written as $\tau = \tau'$, $\tau < \tau'$, $\tau \in X$, where τ, τ' stand for terms built up from ε and variables $x, y, \dots$ by means of the k successor functions. Formulas are built up from the atomic formulas using boolean connectives and the quantifiers $\exists$, $\forall$ applied to either kind of variables. If $\varphi(X_1, \dots, X_n)$ is a formula of this language with the free set variables $X_1, \dots, X_n$, we write $(t_k, P_1, \dots, P_n) \models \varphi(X_1, \dots, X_n)$ to indicate that the k-ary tree t_k satisfies φ with P_i as interpretation for X_i. Let, for $\varphi = \varphi(X_1, \dots, X_n)$,

$$T(\varphi) = \{t \in T_k(\{0,1\}^n) \mid \underline{t} \models \varphi(X_1, \dots, X_n)\}.$$

T is called *SkS-definable* if $T = T(\varphi)$ for some SkS-formula φ. Rabin showed in [14] that a set T of k-ary trees is SkS-definable iff it is Rabin-recognizable. The system *WSkS* (weak second-order theory of k successors) is obtained by restricting the range of the set quantifiers to finite sets only.

The corresponding notions for the case of *one* successor (Büchi and Muller automata on ω-words which characterize the *regular ω-languages*, and the theories S1S and WS1S) will be used without introducing the technical details (see e.g. [18]).

We call *chain logic* (resp. *weak chain logic*) the formalism which results from SkS by restricting the set quantifiers to chains (resp. finite chains). If $T = T(\varphi)$ for a formula φ with this restricted interpretation, we say that T is *definable in chain logic* (resp. *definable in weak chain logic*). Since the property of being a chain is definable in SkS, and being a finite chain is definable in chain logic, we have for $T \subseteq T_k(\{0,1\}^n)$:

- If T is definable in weak chain logic, then T is definable in chain logic.
- If T is definable in chain logic, then T is definable in SkS.

We now show the converse for tree sets T_R which code ω-relations R.

Theorem 2.1. *An ω-relation R is Rabin-definable iff T_R is definable in weak chain logic.*

Proof. It suffices to show the direction from left to right. Let $A=\{0,\ldots,k-1\}$ and $R\subseteq(A^{\omega})^n$. Suppose $\mathscr{A}=(Q,q_0,\Delta,\mathscr{F})$ is a Rabin tree automaton which recognizes T_R. The desired formula $\varphi(X_1,\ldots,X_n)$ defining T_R will be a conjunction of the (first-order expressible) formula

(0) "for $i=1,\ldots,n$: X_i forms a path"

with a formula which expresses in weak chain logic that

(1) "there is a successful run of $\mathscr{A}$ on the $\{0,1\}^n$-valued tree given by $X_1,\ldots,X_n$".

Call a state q of $\mathscr{A}$ *zero-accepting* if the automaton $(Q,q,\Delta,\mathscr{F})$ accepts the trivial k-ary tree t_0 with $t_0(w)=(0,\ldots,0)$ for all $w\in\{0,\ldots,k-1\}^*$. Condition (1) is equivalent to

(2) "there is a partial run r: $\mathrm{hull}(X_1\cup\cdots\cup X_n)\to Q$ of $\mathscr{A}$ which is successful on the paths $X_1,\ldots,X_n$ and which reaches zero-accepting states on the difference set $\mathrm{hull}(X_1\cup\cdots\cup X_n)-(X_1\cup\cdots\cup X_n)$."

A partial run as specified in (2) can be described by an assignment r from $X_1\cup\cdots\cup X_n$ to the set Δ of transitions of $\mathscr{A}$. Since Δ is finite, the transitions can be coded by 0–1-vectors of an appropriate length m, and the existence of the assignment r can be expressed by claiming the existence of the corresponding subsets $Y_1,\ldots,Y_m$ of $X_1\cup\cdots\cup X_n$. We obtain a condition of the form

(3) "there are subsets $Y_1,\ldots,Y_m$ of $X_1\cup\cdots\cup X_n$ which code a partial run as in (2)".

We have to reduce the Y_i to finite chains. Call $w\in X_1\cup\cdots\cup X_n$ a *branching point* of $X_1\cup\cdots\cup X_n$ if w is the empty word ε or not all paths X_i which pass through w also pass through a single successor of w. By a *last branching point* of $X_1\cup\cdots\cup X_n$ we mean a branching point which is maximal w.r.t. $<$. The property of being a branching point (last branching point) is first-order-definable in terms of $X_1\cup\cdots\cup X_n$. The branching points define a decomposition of $X_1\cup\cdots\cup X_n$ into segments given by the pairs of $<$-consecutive branching points (which are finite chains) and into infinite chains starting at the last branching points. Assuming that there are p such finite and infinite segments, each of the sets Y_i can be split into subsets $Z_{i1},\ldots,Z_{ip}$ such that the Z_{ij} code partial runs delimited by (and including) branching points. Since the Z_{ij} are finite chains when contained in segments between consecutive branching points, it suffices to treat the case of sets $X_i[w]:=X_i\cap\{u\mid w\leqslant u\}$ for last branching points w. So it remains to reduce to weak chain logic a condition of the form

(4) "there are subsets $Z_1,\ldots,Z_m$ of $X_i[w]$ which code a partial run of $\mathscr{A}$ on $X_i[w]$ as in (2)".

This condition is expressible in monadic second-order logic S1S as a statement on ω-words over the alphabet $\{0, \ldots, k-1\}$ (representing $X_i[w]$). By [16] (or [18, Theorem 4.6]), an S1S-formula is equivalent to a formula of the weak monadic theory WS1S. The retranslation into chain logic over the k-ary tree is a formula of weak chain logic which expresses condition (4). $\square$

In what follows, we give two consequences of Theorem 2.1. The first presents a normal form for Rabin-definable ω-relations in terms of regular languages and regular ω-languages, which is a natural extension of the "special relations" over A^* as introduced in [12]. The second shows that deterministic Rabin tree automata suffice for recognizing tree sets T_R.

The description of Rabin-definable ω-relations in terms of languages is based on the above-mentioned decomposition of a tuple of paths into segments delimited by the branching points. The representation is technically cumbersome due to the necessary distinction of the possibilities by which the individual paths may diverge. Given an n-tuple $(\alpha_1, \ldots, \alpha_n) \in (A^\omega)^n$, the segments between consecutive branching points, starting at last branching points (as introduced in the above proof), form a finite tree in which each node represents a segment. Label each node by those indices i such that the corresponding segment is a segment of α_i. Call the resulting finite tree labelled by nonempty subsets of $\{1, \ldots, n\}$ the *branching pattern* of $(\alpha_1, \ldots, \alpha_n)$. (Its root is labelled $\{1, \ldots, n\}$, and the labels of the sons of a node x form a proper partition of the label of x.)

Example. Figure 1 shows a 5-tuple of ω-words, along with its branching pattern.

Now suppose that with each inner node of a given branching pattern p a language $W \subseteq A^*$ and with each leaf an ω-language $L \subseteq A^\omega$ is associated. Let these languages and ω-languages be indexed as $W_1, \ldots, W_r$ and $L_{r+1}, \ldots, L_{r+s}$, respectively. Given an n-tuple $(\alpha_1, \ldots, \alpha_n)$ with branching pattern p, the *index sequence* of the ith component α_i is the finite sequence of indices of the W_j, L_j which are associated with the path through p whose nodes contain i.

Example (*continued*). Given $(\alpha_1, \ldots, \alpha_5)$ as in Fig. 1 and (ω-) languages associated with its branching pattern in the form shown in Fig. 2, the index sequence of α_1 is (1, 5), and the index sequence of α_2 is (1, 2, 3).

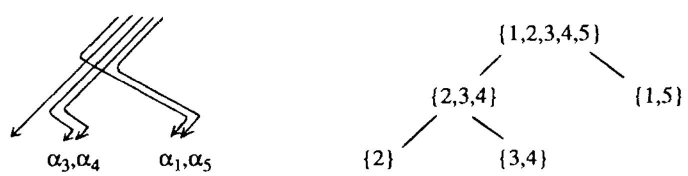

Fig. 1.

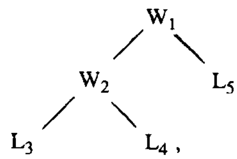

Fig. 2.

Assume that with the nodes of a branching pattern p the languages $W_1, \ldots, W_r$, $L_{r+1}, \ldots, L_{r+s}$ are associated. An ω-relation $R \subseteq (A^{\omega})^n$ is *generated by* $W_1, \ldots, W_r$ and $L_{r+1}, \ldots, L_{r+s}$ *via the branching pattern* p if R contains those n-tuples $(\alpha_1, \ldots, \alpha_n)$ which have a branching pattern p, such that there are words $w_1 \in W_1, \ldots, w_r \in W_r$ and ω-words $\beta_{r+1} \in L_{r+1}, \ldots, \beta_{r+s} \in L_{r+s}$ with

$$\alpha_i = w_{i(1)} \ldots w_{i(m-1)} \beta_{i(m)},$$

where $(i(1), \ldots, i(m))$ is the index sequence for α_i.

Definition 2.2. An ω-relation $R \subseteq (A^{\omega})^n$ is *special* if there are a branching pattern p, regular languages $W_1, \ldots, W_r \subseteq A^*$, and regular ω-languages $L_{r+1}, \ldots, L_{r+s} \subseteq A^{\omega}$ such that R is generated by $W_1, \ldots, W_r, L_{r+1}, \ldots, L_{r+s}$ via p.

Theorem 2.3. *An ω-relation is Rabin-definable iff it is a finite union of special ω-relations.*

Proof. If the ω-relation R is generated from regular languages $W_1, \ldots, W_r$ and regular ω-languages $L_{r+1}, \ldots, L_{r+s}$ as in the above definition, a Rabin tree automaton recognizing T_R can be constructed from sequential automata recognizing the W_i and L_i. By closure of Rabin-recognizable sets under finite union, finite unions of special ω-relations are Rabin-definable. Conversely, consider a Rabin tree automaton $\mathscr{A}$ which recognizes a set T_R. Given the arity of R there are only finitely many possible branching patterns. Each branching pattern p induces a subset Δ_p of the transition set Δ of $\mathscr{A}$, containing the transitions which can be used for acceptance of trees $t_{\bar{\alpha}}$, where $\bar{\alpha}$ is of branching pattern p. The resulting tree automaton $\mathscr{A}_p$ defines a special ω-relation R_p. (We do not describe in detail the straightforward but tedious task of extracting from $\mathscr{A}_p$ the sequential automata for the (ω-) languages W_i, L_i which generate R_p.) Since R is the union of the ω-relations R_p (where p ranges over all branching patterns for the arity of R), it is a union of special ω-relations. $\square$

Applied to "finitary" relations $R \subseteq (A^*)^n$, Theorems 2.1 and 2.3 re-prove (in simplified form) the main result of [12].

Let R be a special ω-relation. A Rabin tree automaton that recognizes T_R has to check the designated paths of an input tree for a certain branching pattern, and along each designated path has to check whether the segments delimited by the branching points belong to given regular languages (or ω-languages). Both tests can be performed deterministically (for the membership in regular ω-languages, due to McNaughton's theorem on determinizing ω-automata). So, any special ω-relation is defined by a deterministic Rabin tree automaton. Hence, by products of deterministic Rabin tree automata (each factor for a fixed branching pattern), finite unions of special ω-relations are recognized. Since these products are again deterministic Rabin tree automata, we obtain the following corollary.

Corollary 2.4. *An ω-relation is Rabin-definable iff T_R is recognized by a deterministic Rabin tree automaton.*

3. Classes of automaton-definable ω-relations

In this section we describe some possibilities for specifying ω-relations in terms of sequential finite automata. The different versions of acceptance for tuples of ω-words are obtained by allowing different kinds of dependency between the (one-way) scanning processes on the components of a tuple. The scanning may proceed completely independently on the components, or "strictly in parallel" (i.e. letter by letter on all components), or by means of reading heads which may move at different speeds on the components and communicate via the finite control. These three possibilities lead to the classes of componentwise recognizable, Büchi-definable, and rational ω-relations, respectively.

We call an ω-relation $R \subseteq (A^{\omega})^n$ *componentwise recognizable* iff it is a finite union of sets $L_1 \times \cdots \times L_n$, where each ω-language L_i is regular (i.e. recognized by a Büchi automaton over A). Thus, membership of tuples $(\alpha_1, \ldots, \alpha_n)$ of ω-words in R is determined by membership of the individual components α_i in given regular ω-languages. Let ω-RCG be the class of componentwise recognizable ω-relations.

If one uses Büchi automata over A^n, which scan all components of an n-tuple $(\alpha_1, \ldots, \alpha_n)$ letter by letter from left to right, one views an n-tuple from $(A^{\omega})^n$ as an ω-word from $(A^n)^{\omega}$. The ω-relations recognized in this way by Büchi automata are the *Büchi-definable* ω-relations (or *letter-to-letter* relations of [6, 7]) as defined in Section 1. We denote by ω-BÜC the class of Büchi-definable ω-relations.

It is straightforward (but technically cumbersome) to define "special Büchi automata" over A^n which characterize the special n-ary ω-relations as introduced in Section 2. Since we do not need this automaton model here, we just give an informal description. The idea is to specialize the model of a Büchi automaton on A^n (as used for the Büchi-definable relations) in the following way: The automaton starts scanning the components of $(\alpha_1, \ldots, \alpha_n)$ letter to letter in parallel until different letters occur in two components, after which the scanning continues independently in all blocks of mutually identical components. In general, components $\alpha_{i_1}, \ldots, \alpha_{i_r}$ of $(\alpha_1, \ldots, \alpha_n)$ are scanned letter by letter in parallel as long as they coincide, and when a difference occurs (say between $\alpha_{i_1}, \ldots, \alpha_{i_s}$ and $\alpha_{i_{s+1}}, \ldots, \alpha_{i_r}$) the automaton branches into separate scanning processes for the two indicated blocks of components, which by themselves are still processed letter by letter in parallel. The model is, thus, located between the automaton characterization of ω-RCG, where all components are scanned individually from the start, and that of ω-BÜC, where the parallel scanning of the components continues through the whole ω-computation. Since these automata are nondeterministic and, thus, are closed under union, this can serve as a description of the Rabin-definable ω-relations.

The last automaton model employs n reading heads which are moved forward separately, but coordinated by the finite control, on the n components of an n-tuple $(\alpha_1, \ldots, \alpha_n)$ given to the automaton on n right-infinite input tapes. For the case of finite words, this model goes back to Rabin and Scott [15]; a version for ω-relations was studied by Gire and Nivat [9]. The nondeterministic transition table specifies for any state and any n-tuple of scanned letters which possibilities are admitted to move one or more reading heads forward by one tape cell and reach a new state. An ω-relation $R \subseteq (A^\omega)^n$ is called *rational* iff it is recognized by an acceptor of this type (where an n-tuple $(\alpha_1, \ldots, \alpha_n)$ is accepted if it admits an infinite run such that each component α_i is scanned completely and some final state is reached infinitely often). Let ω-RAT be the class of rational ω-relations. In [9] these ω-relations are characterized by a calculus of rational expressions. Frougny and Sakarovitch [6, 7] studied the restriction of *bounded delay* for successful computations of multihead automata (where the reading heads may move apart only by a fixed amount of cells); they showed that in this way exactly the Büchi-definable (or "letter-to-letter") ω-relations are recognized.

A subclass of ω-RAT is obtained by considering *deterministic* finite multihead automata with transitions as explained above, equipped with a system $\mathscr{F}$ of final-state sets and accepting in Muller's sense (where a run is successful if the states visited infinitely often form a set in $\mathscr{F}$). The ω-relations recognized by such *deterministic multihead Muller automata* are called *deterministic rational*, and the class of these ω-relations is denoted by ω-DRAT.

Proposition 3.1. *We have*

$$\omega\text{-RCG} \subset \omega\text{-RBN} \subset \omega\text{-BÜC} \subset \omega\text{-DRAT} \subset \omega\text{-RAT},$$

and all inclusions in this chain are strict.

Proof. The inclusions $\omega\text{-RCG} \subseteq \omega\text{-RBN} \subseteq \omega\text{-BÜC}$ are easy. The inclusion ω-BÜC $\subseteq \omega$-DRAT is clear from McNaughton's theorem (see e.g. [18] for details), by which a nondeterministic Büchi automaton (here over (A^n)) is equivalent to a deterministic Muller automaton. For the last inclusion one observes that deterministic Muller automata with separately moving heads are generalized by nondeterministic automata of this type, and that the latter can be simulated by nondeterministic multihead automata with the Büchi acceptance condition (as in [9]). The required automaton has to guess one of the final-state sets (say the set F) at the start of a run, and it has to guess a position in the run from which onwards exactly the F-states are repeated. This fact can be recorded by assuming infinitely often a state which signals that a cycle through the set F has been completed.

Let $A=\{0,1\}$. We present relations $R_1,\dots,R_4$ which can serve as examples showing the strictness of the above inclusions (proceeding from left to right). All proofs work by simple pumping arguments.

Let $R_1=\{(\alpha,\beta)\in(A^\omega)^2 \mid \text{for some } i,\ \alpha\in 0^i1A^\omega \text{ and } \beta\in 0^i1A^\omega\}$. Clearly, R_1 is Rabin-definable. Assume that R_1 is a finite union of sets $L\times L'$ with regular ω-languages L, L'. Choose n such that these ω-languages are recognized by Büchi automata with at most n states. Consider the pair $(0^n1^\omega, 0^n1^\omega)\in R_1$, which belongs, say, to $L\times L'\subseteq R_1$. Applying a pumping argument to the automaton recognizing L, we obtain a pair of form $(0^m1^\omega, 0^n1^\omega)$, with $m<n$, which is also in $L\times L'$, which is a contradiction.

Let $R_2=\{(\alpha,\beta)\in(A^\omega)^2 \mid \text{for some } i,\ \alpha\in 0^i1A^\omega \text{ and } \beta\in 1^i0A^\omega\}$, which is Büchi-definable. If $\mathscr{A}$ is a Rabin tree automaton with n states, assumed to recognize the tree set associated with R_2, $\mathscr{A}$ accepts the tree coding $(0^n1^\omega, 1^n0^\omega)$, and one can construct (by state repetition on the leftmost branch) a pair $(0^m1^\omega, 1^n0^\omega)$, with $m<n$, whose tree is also accepted by $\mathscr{A}$, which is a contradiction.

Let $R_3=\{(\alpha,\beta)\in(A^\omega)^2 \mid \text{for some } i,\ \alpha\in 1^*0^i1A^\omega \text{ and } \beta\in 1^*0^i1A^\omega\}$. A deterministic automaton with separately moving reading heads which recognizes R_3 (even with Büchi acceptance) waits on the first scanned letter 0 until in the other component also letter 0 is reached; then by moving forward in both components letter by letter it is checked whether letter 0 occurs the same number of times in both components before a letter 1 is encountered again (after which a final state is repeated). Suppose R_3 were Büchi-definable, say by a Büchi automaton over A^2 with n states, which scans an input pair (α,β) letter by letter from the start. From an accepting run on $(0^n1^\omega, 1^n0^n1^\omega)$ an accepting run on $(0^m1^\omega, 1^m0^n1^\omega)$ with $m<n$ can be obtained.

Let $R_4=\{(\alpha,\beta)\in(A^\omega)^2 \mid \text{for some } i,\ \alpha\in A^*10^i1A^\omega \text{ and } \beta\in A^*10^i1A^\omega\}$. A nondeterministic multihead automaton which recognizes R_4 simply guesses a pair of positions on an input pair (α,β) where two segments of letter 0 of equal length start, and checks this by further proceeding letter by letter on both components. Assume that a deterministic automaton with separately moving heads (and with Büchi or Muller acceptance) recognizes R_4, and that there are n states. Consider on the input $(10^n1^\omega, 10^{n+1}10^n1^\omega)$ the point in a successful computation where the second 1 is reached on one of the two components for the first time. If this happens with the first component, a state repetition has occurred there such that, say, r letters 0 (where $r>0$) have been scanned in between, while on the second component, say, s letters 0 (and possibly the leading 1) have been scanned. If $r\geqslant s$, one obtains by deletion of 0^r and $(1)0^s$ a pair of ω-words which is not in R_4 but nevertheless accepted; if $s>r$, an insertion of 0^r and $(1)0^s$ will cause the same. The case that the second 1 is first reached in the second component is handled similarly. □

Remark 3.2. (a) From the definitions it is immediate that the *unary* ω-relations in any of the classes of Proposition 3.1 coincide with the regular ω-languages.

(b) The statement of Proposition 3.1 is true also for the corresponding classes of relations over A^*. (In the definition of the example relations R_i, replace A^ω everywhere by A^*.)

4. Büchi-definable ω-relations and chain logic with the equal-level predicate

Elgot and Rabin studied in [4] the "theory of generalized successor", denoted as GS, the first-order theory of the structure $(\{0,1\}^*, \varepsilon, \cdot 0, \cdot 1, E)$ with the empty word ε, the two successor functions and the equal-length predicate E on $\{0,1\}^*$. Since we view $\{0,1\}^*$ as the full binary tree, we refer to E as the "equal-level predicate". Elgot and Rabin showed that GS is decidable, and they raised the question of decidable extensions of GS. They proved that many predicates and functions cause undecidability when added to GS. While GS with predicates like $P = \{0^i \mid i \text{ is a factorial}\}$ remains decidable, no "interesting" binary predicate P on $\{0,1\}^*$ is known such that the extension of GS by P is decidable.

In this paper we introduce a different proper extension of GS which is decidable, by allowing quantifiers ranging over chains in the tree of finite words. It will be shown that this theory, called *chain logic* + *E*, and the monadic theory S1S of one successor can be interpreted in each other. This yields a characterization of Büchi-definable ω-relations in terms of chain logic + E, as well as decidability of the theory of the infinite k-ary tree in the language of chain logic + E. For simplicity of exposition we treat only binary trees; the generalization to k-ary trees needs some additional coding.

S1S-formulas $\varphi(X_1, \ldots, X_n)$ are satisfied by n-tuples of subsets of 0^* or, equivalently, of the set ω of natural numbers. Each subset of ω can be represented by its characteristic function (an ω-word over $\{0,1\}$). Hence, the possible models for S1S-formulas $\varphi(X_1, \ldots, X_n)$ may be considered as n-tuples $\bar{\alpha} = (\alpha_1, \ldots, \alpha_n) \in (\{0,1\}^\omega)^n$.

We shall represent tree models for formulas $\psi(X_1, \ldots, X_n)$ of chain logic + E in a similar way. These models are given by the binary tree with n designated chains, as an interpretation of the X_i. To describe a model, it suffices to specify these n chains. One chain C can be represented as a pair (δ, β) of 0–1-sequences, where δ (the "direction sequence") codes the leftmost path through the binary tree of which C is a subset, and β indicates which nodes on this path belong to C and which do not. (The leftmost path is taken in order to have a well-defined coding in the case of finite chains.) Given a binary tree model $\underline{t} = (t_2, P_1, \ldots, P_n)$, where the P_i are chains, we say that the sequence tuple $(\delta_1, \beta_1, \ldots, \delta_n, \beta_n)$ *codes* $\underline{t}$ if for $i = 1, \ldots, n$ the pair (δ_i, β_i) represents the chain P_i in this way.

Theorem 4.1. (a) *Chain logic + E over the binary tree can be interpreted in S1S in the following sense: For any formula $\varphi(X_1, \ldots, X_n)$ of chain logic + E there is by effective* *construction an S1S-formula* $\psi(Y_1, Z_1, \ldots, Y_n, Z_n)$ *such that for all tree models* $t=(t_2, P_1, \ldots, P_n)$ *with chains* P_i *we have*

$$\underline{t} \models \varphi(X_1, \ldots, X_n) \text{ iff } \psi(Y_1, Z_1, \ldots, Y_n, Z_n) \text{ is satisfied by the sequence tuple } (\delta_1, \beta_1, \ldots, \delta_n, \beta_n) \text{ which codes } \underline{t}.$$

(b) *S1S can be interpreted in chain logic + E over the binary tree in the following sense: For any S1S-formula* $\varphi(X_1, \ldots, X_n)$ *there is by effective construction a formula* $\psi(X_1, \ldots, X_n)$ *of chain logic + E such that*

$$\varphi(X_1, \ldots, X_n) \text{ is satisfied by } \bar{\alpha}=(\alpha_1, \ldots, \alpha_n) \text{ iff } \underline{t}_{\bar{\alpha}} \models \psi(X_1, \ldots, X_n).$$

Proof. (a) It is convenient to work with a version of chain logic in which only set variables (but no individual variables) occur. The atomic formulas are of the form $X_1 \subseteq X_2$ ("chain X_1 is included in chain X_2"), Sing X ("chain X is a singleton"), X_1 succ$_i$ X_2 ("chains X_1, X_2 are singletons $\{x_1\}$, $\{x_2\}$ such that x_1 has x_2 as the ith successor"), and $X_1 \, E \, X_2$ ("chains X_1, X_2 are singletons with elements on the same level of the tree"). It is easy to see that this version of chain logic + E is expressively equivalent to the original one (cf. [18, Theorem 3.1]).

The proof of part (a) works by induction over the formulas of this modified chain logic. For better readability we allow in S1S-formulas the (definable) $\leqslant$-relation and write $z+1$ for the successor of z. An atomic formula $X_1 \subseteq X_2$ of chain logic + E is translated to the S1S-formula

$$\psi_{\subseteq}(Y_1, Z_1, Y_2, Z_2): \quad \forall y(\exists z(y \leqslant z \wedge z \in Z_1) \Rightarrow (y \in Y_1 \Leftrightarrow y \in Y_2)) \\ \wedge \forall z(z \in Z_1 \Rightarrow z \in Z_2);$$

similarly, Sing X is translated to the formula

$$\psi_S(Y, Z): \quad \exists z(z \in Z \wedge \forall z'(\neg z = z' \Rightarrow \neg z' \in Z)),$$

X_1 succ$_1$ X_2 to

$$\psi_1(Y_1, Z_1, Y_2, Z_2): \quad \psi_S(Y_1, Z_1) \wedge \psi_S(Y_2, Z_2) \\ \wedge \exists z(z \in Z_1 \wedge z+1 \in Z_2 \\ \wedge \forall y(y \leqslant z \Rightarrow (y \in Y_1 \Leftrightarrow y \in Y_2)) \wedge z+1 \in Y_2)$$

(for X_1 succ$_0$ X_2 the last atomic formula is changed to $\neg z+1 \in Y_2$) and, finally, $X_1 \, E \, X_2$ to

$$\psi_E(Y_1, Z_1, Y_2, Z_2): \quad \psi_S(Y_1, Z_1) \wedge \psi_S(Y_2, Z_2) \wedge \exists z(z \in Z_1 \wedge z \in Z_2).$$

The induction steps (for which the cases $\neg$, $\vee$, $\exists$ suffice) are obvious; if the chain logic formula $\varphi(X, \ldots)$ is expressed in S1S by $\psi(Y, Z, \ldots)$ then $\exists X \varphi(X, \ldots)$ is translated to $\exists Y \exists Z \psi(Y, Z, \ldots)$ in S1S.

(b) From [2], an S1S-formula is equivalent to a Büchi automaton $\mathscr{A}$ over $\{0,1\}^n$. So it suffices to find for any such Büchi automaton $\mathscr{A}$ a formula $\psi(X_1,\ldots,X_n)$ of chain logic + E such that

$$\mathscr{A} \text{ accepts } \bar{\alpha}=(\alpha_1,\ldots,\alpha_n) \text{ iff } \underline{t}_{\bar{\alpha}} \models \psi(X_1,\ldots,X_n).$$

The formula ψ has to express the existence of a successful run of $\mathscr{A}$ on α. If there are (without loss of generality) 2^m states in $\mathscr{A}$, given as 0–1-vectors of length m, this can be formulated as the existence of an m-tuple $(Y_1,\ldots,Y_m)$ of subsets of the leftmost path 0^ω of the binary tree. For instance, if the run assumes state $(1,0,1)$ at step $y\in 0^*$, we should have $y\in Y_1, y\notin Y_2, y\in Y_3$. In expressing that the state sequence is compatible with the input $\bar{\alpha}$ and the transition table of $\mathscr{A}$ we use the predicate E: if, for example, we deal with the alphabet $\{0,1\}^2$ and state set $\{0,1\}^3$, and the transition $(q,(1,0),q')$ is applied with $q=(0,0,0)$ and $q'=(1,1,1)$ at step $y\in 0^*$, we have

$y\notin Y_1, y\notin Y_2, y\notin Y_3$,
$x_1 1\in X_1$ for the unique $x_1\in X_1$ with $y E x_1$,
$x_2 0\in X_2$ for the unique $x_2\in X_2$ with $y E x_2$, and
$y0\in Y_1, y0\in Y_2, y0\in Y_3$.

Let $\psi_1(y,X_1,\ldots,X_n,Y_1,\ldots,Y_m)$ be the disjunction of these formulas over all transitions of $\mathscr{A}$. Then the desired formula ψ can be written in the form

$$\exists Y_1\ldots Y_m(\psi_0(Y_1,\ldots,Y_m)$$
$$\wedge \forall y(y\in 0^* \Rightarrow \psi_1(y,X_1,\ldots,X_n,Y_1,\ldots,Y_m)) \wedge \psi_2(Y_1,\ldots,Y_m)),$$

where ψ_0 expresses the initial-state condition ("state $(0,\ldots,0)$ at step ε") and ψ_2 the acceptance condition ("for each $x\in 0^*$ there is $y\in 0^*$ with $x\leqslant y$ such that at step y a final state is assumed"). It is easy to formalize these conditions in chain logic. $\square$

Part (a) of the above proof extends to the case of k-ary trees (say for $k=2^r$) by replacing the sequences δ_i by r-tuples of 0–1 sequences. The following results are consequences of Theorem 4.1 (in the generalization to k-ary trees).

Corollary 4.2. *An ω-relation R is Büchi-definable iff T_R is definable in chain logic + E.*

For the proof of Corollary 4.2 (from right to left) note that formulas of chain logic + E are interpreted only over trees $t_{\bar{\alpha}}$, i.e. with designated *paths* only (instead of chains). Hence, in the coding of trees $t_{\bar{\alpha}}$ by tuples of 0–1-sequences the β_i-components can be cancelled, and the S1S-formula obtained by Theorem 4.1(a) is of the form $\psi(Y_1,\ldots,Y_n)$, defining an n-ary Büchi-definable ω-relation as desired.

For the case of formulas without free variables, we conclude from Theorem 4.1 by decidability of S1S the following corollary.

Corollary 4.3. *The theory of the k-ary tree in the language of chain logic + E is decidable.*

The predicate E is not definable in full monadic second-order logic SkS, and when adjoined to SkS it yields an undecidable theory (see [12] or [18] for a proof). So, chain logic + E over the k-ary tree gives a decidable theory in which tree properties which are not expressible in SkS can be defined.

Finally, by reduction of S1S to WS1S as mentioned in Section 2, we obtain the following corollary.

Corollary 4.4. *Over tree models $\underline{t}_{\bar{\alpha}}$, chain logic + E is expressively equivalent to weak chain logic + E.*

5. Chain logic + E over regular trees, and finite-state programs

In the preceding sections, the interpretation of logical formulas over trees refers to a rather restricted kind of model, the path-valued (or chain-valued) trees. In this section we discuss certain trees which are not necessarily path-valued and are of interest for several applications: the regular trees. (The case of arbitrary tree models is not considered in the present paper.)

A k-ary B-valued tree t is called *regular* if t contains only finitely many nonisomorphic subtrees. Equivalently, for each letter $b \in B$ the set

$$V_b = \{w \in \{0, \ldots, k-1\}^* \mid t(w) = b\}$$

is regular. The following lemma is shown using definability in WS1S of regular sets of finite words (see e.g. [18]).

Lemma 5.1. *Let t be a regular B-valued k-ary tree. Then for each letter $b \in B$ there is a formula $\varphi_b(x)$ of weak chain logic with one free individual variable x such that the unvalued k-ary tree t_k with designated node w satisfies $\varphi_b(x)$ iff $t(w) = b$.*

Proof. Suppose that the set V_b is recognized by the finite automaton $\mathscr{A}$. The formula $\varphi_b(x)$ expresses that $\mathscr{A}$ has a successful run when reading as input the directions taken on the finite path up to node x. This can be expressed in weak chain logic using auxiliary subsets of the finite path up to x as codes for the states assumed by the automaton. □

Theorem 5.2. *It is decidable whether an effectively given regular $\{0,1\}^n$-valued k-ary tree (i.e. a model $(t_k, P_1, \ldots, P_n)$) satisfies a given formula $\varphi(X_1, \ldots, X_n)$ of chain logic + E.*

Proof. We transform $\varphi(X_1, \ldots, X_n)$ into a *sentence* in the language of chain logic + E (i.e. a formula without free variables) which is true in the unvalued tree t_k iff $(t_k, P_1, \ldots, P_n)$ satisfies $\varphi(X_1, \ldots, X_n)$. Then the result follows from Corollary 4.3. For the transformation, one only has to replace each of the X_i by an explicit definition in chain logic. This explicit definition is provided by Lemma 5.1 since the given tree is regular. More precisely, one substitutes each atomic formula $x \in X_i$ in $\varphi(X_1, \ldots, X_n)$ by the disjunction of all formulas $\varphi_b(x)$ such that the ith component of $b \in \{0, 1\}_b$ is 1. □

An effective test as guaranteed by Theorem 5.2 is needed in the verification of finite-state programs with respect to specifications in a system of branching time logic ("model checking", cf. [3]). For this purpose, several systems of branching time logic have been considered in the literature (such as CTL, CTL*, ECTL*), which can all be considered as fragments of chain logic interpreted over k-ary trees (see e.g. [10]). In the sequel we consider the properly more expressive system of chain logic $+E$ as a specification language for finite-state programs.

In this context, a (possibly concurrent) finite-state program P is considered as a finite directed graph G_P whose nodes represent the program's states and whose edges represent the possible transitions in one step. There is a designated initial state s_0. If for the specification the state properties $q_1, \ldots, q_m$ are relevant, each state s is annotated by those q_i which are true in s. Let t_P be the state tree which results from G_P by unravelling it from the initial state, where again the nodes are labelled by the q_i as prescribed by G_P. A specification for P is a formula φ to be interpreted in t_P; in our case φ is a formula of chain logic $+ E$ where for each state property q_i the atomic formula $x \in Q_i$ ("in state x the property q_i holds") is available. Program P is correct with respect to the specification φ if t_P satisfies φ.

Note that, by the finiteness of G_P, the tree t_P is at most $|G_P|$-ary and regular; also, given P, this tree is effectively presented. Hence, we obtain from Theorem 5.2 the following corollary.

Corollary 5.3. *With respect to specifications in chain logic $+ E$, the correctness of finite-state programs is decidable.*

The predicate E allows one to express over infinite trees that node x has equal distance to nodes y and z (in the subtree at x). As mentioned in the previous section, such "uniformity" conditions [5] transcend the expressive power of SkS and Rabin tree automata. This applies *a fortiori* to chain logic and the above-mentioned systems CTL, CTL*, ECTL* of branching time logic. The additional expressive power provided by the E-predicate may be useful in applications where one wants to express liveness properties combined with time constraints (guaranteeing the "uniform occurrence" of certain events along all computation paths).

The complexity of the algorithm given by Corollary 5.3 is nonelementary in the length of the specification in chain logic $+E$ (since the theory S1S, used in the decision procedure, is nonelementary). It remains to be investigated how a better complexity bound can be obtained by rebuilding the syntax of chain logic $+E$, e.g. when replacing quantifiers by suitable automaton operators in the sense of [19]. Note that S1S is expressively equivalent to such a system with automaton operators ("extended temporal logic", ETL, cf. [19]), for which the satisfiability problem is in PSPACE.

6. Concluding remarks

We have studied several classes of automaton-definable ω-relations and their description in terms of tree automata, sequential automata, and systems of monadic second-order logic. For technical convenience only relations over ω-words were considered; one can adjust the characterization results to the more general case of relations $R \subseteq (A^{\infty})^n$, where $A^{\infty} = A^* \cup A^{\omega}$.

Compared to the extensive research on regular ω-languages, there are only few papers concerned with automaton-definable ω-relations (or functions over ω-words). As the above results indicate, this subject is of interest not only by the diversity of definability notions (which collapse to one notion in the case of ω-languages, cf. Remark 3.2(a) above), but also by its close connection to path-oriented logics over trees, and in view of other applications. For instance, in the analysis of concurrent systems, it can be useful to extend the study of properties of execution sequences, prevailing in the literature, to the investigation of relations between sequences.

We mention some topics for further investigation. One question concerns a logical characterization of ω-DRAT or ω-RAT. Since neither of these relation classes is closed under the boolean operations, the logic should involve restrictions on the use of negation. Logics with quantifiers for transitive closure or the least-fixed-point operator (cf. [11]) might be appropriate. An alternative is to investigate the boolean closures of ω-DRAT or ω-RAT.

Between ω-BÜC and ω-DRAT there is a gap in expressiveness. This is due to the fact that the concatenation relation is definable by a deterministic multihead automaton (over finite words) and, thus, by a result of Quine [13], the closure of ω-DRAT under projections and boolean operations includes all recursive ω-relations and, hence, is as expressive as first-order arithmetic. On the other hand, ω-BÜC is closed under projections and boolean operations and contains only very special recursive ω-relations. (For a more detailed discussion see [17].) This suggests a study of relation classes between ω-BÜC and ω-DRAT. Recently, Frougny and Sakarovitch [6, 7] have investigated restrictions for the movements of the reading heads of the automata.

The strictness of the inclusions of Proposition 3.1 raises the question whether there are algorithms which decide membership in ω-RCG, ω-RBN, ω-BÜC, ω-DRAT for a given relation in ω-RAT; similarly, for membership in ω-RCG, ω-RBN for a given relation in ω-BÜC. (Within ω-RAT, the property of being a rational function is decidable by [8].)

Finally, concerning the system chain logic $+ E$, it seems interesting to look for other relations R such that chain logic $+ R$ (or chain logic $+ E + R$) is decidable. Also, the case of arbitrary-valued trees as underlying models (not necessarily coding tuples of paths) remains to be investigated. We conjecture that in this general case SkS and chain logic $+ E$ are incompatible in expressive power, i.e. that there are SkS-definable tree sets which cannot be defined in chain logic $+ E$. A proposed example is the set of $\{0, 1\}$-valued trees having finitely many nodes with value 1, such that the number of $<$-maximal nodes with value 1 is even.

Acknowledgment

I thank the referees of STACS 90 for their comments, and C. Frougny and J. Sakarovitch for pointing out an inaccuracy in the preliminary version of Section 3.

References

[1] J. Berstel, *Transductions and Context-Free Languages* (Teubner, Stuttgart, 1979).
[2] J.R. Büchi, On a decision method in restricted second order arithmetic, in: Proc. 1960 Internat. Congress on *Logic, Methodology and Philosophy of Science.* (Stanford University Press, Stanford, 1962) 1–11.
[3] E.M. Clarke, E.A. Emerson and A.P. Sistla, Automatic verification of finite-state concurrent systems using temporal logic specifications, *ACM Trans. Progr. Lang. Systems* **8** (1986) 244–263.
[4] C.C. Elgot and M.O. Rabin, Decidability and undecidability of second (first) order theory of (generalized) successor, *J. Symbolic Logic* **31** (1966) 169–181.
[5] E.A. Emerson, Uniform inevitability is tree automaton ineffable, *Inform. Process. Lett.* **24** (1987) 77–80.
[6] C. Frougny and J. Sakarovitch, Synchronized rational relations, in: *Proc. ASMICS Workshop "Logics and Recognizable Sets"*, Rep. 9104, Inst. für Informatik der Universität Kiel, 1991, 41–44.
[7] C. Frougny and J. Sakarovitch, Rational relations with bounded delay, in: C. Choffrut and M. Jantzen, eds., *Proc. STACS 91*, Lecture Notes in Computer Science, Vol. 480 (Springer, Berlin, 1991) 50–63.
[8] F. Gire, Two decidability problems for infinite words, *Inform. Process. Lett.* **22** (1986) 135–140.
[9] F. Gire and M. Nivat, Relations rationelles infinitaires, *Calcolo* **21** (1984) 91–125.
[10] Th. Hafer and W. Thomas, Computation tree logic CTL* and path quantifiers in the monadic theory of the binary tree, in: T. Ottmann, ed. *Proc. 14th ICALP*, Lecture Notes in Computer Science, Vol. 267 (Springer, Berlin, 1987) 269–279.
[11] N. Immerman, Languages that capture complexity classes, *SIAM J. Comput.* **16** (1987) 760–778.
[12] H. Läuchli and C. Savioz, Monadic second order definable relations on the binary tree, *J. Symbolic Logic* **52** (1987) 219–226.
[13] W.V. Quine, Concatenation as a basis for arithmetic, *J. Symbolic Logic* **11** (1946) 105–114.
[14] M.O. Rabin, Decidability of second-order theories and automata on infinite trees, *Trans. Amer. Math. Soc.* **141** (1969) 1–35.
[15] M.O. Rabin and D. Scott, Finite automata and their decision problems, *IBM J. Res. Develop.* **3** (1959) 114–125.
[16] W. Thomas, On the bounded monadic theory of well-ordered structures, *J. Symbolic Logic* **45** (1980) 334–338.
[17] W. Thomas, Automata and quantifier hierarchies, in: J.E. Pin, ed., *Formal Properties of Finite Automata and Applications*, Lecture Notes in Computer Science, Vol. 386 (Springer, Berlin, 1989) 104–119.
[18] W. Thomas, Automata on infinite objects, in: J.v. Leeuwen, ed., *Handbook of Theoretical Computer Science, Vol. B* (Elsevier, Amsterdam, 1990) 133–191.
[19] M.Y. Vardi and P. Wolper, Reasoning about infinite computation paths, to appear.