AUTOMATA ON INFINITE OBJECTS

Wolfgang Thomas

Lehrstuhl für Informatik II
RWTH Aachen
D-5100 Aachen

Abstract. This paper is a preliminary version of a chapter for the forthcoming "Handbook of Theoretical Computer Science" (Managing Editor: J.v. Leeuwen) to be published by North-Holland. It contains an introduction to the theory of automata on ω-words and on infinite trees, and surveys recent developments and applications.

Aachener Informatik-Berichte 88-17 October 1988

Contents

Introduction

The subject of automata on infinite sequences and infinite trees was established in the sixties by Büchi [1962], McNaughton [1966], and Rabin [1969]. Their work introduced intricate automaton constructions, opened new connections between automata theory and other fields (for example, logic and set theoretic topology), and resulted in a theory which is fundamental for those areas in computer science where nonterminating computations are studied.

A main motivation of the early papers was the investigation of decision problems in mathematical logic. Büchi discovered that automata provide a normal form for certain monadic second-order theories. Rabin's tree theorem (which states that the monadic second-order theory of the infinite binary tree is decidable) turned out to be a powerful result to which a large number of other decision problems could be reduced.

From this core the theory has developed into numerous directions. Today, a major application area is the specification and verification of concurrent programs: many aspects of them are studied adequately in terms of nonterminating computations of automata. Moreover, most of the logical specification formalisms for concurrent programs (such as systems of program logic or temporal logic) are embeddable in the monadic theories studied by Büchi and Rabin; in some sense these theories can be considered as "universal process logics" for linear, resp. branching computations.

Besides applications in program logics and in the development of concurrent systems (see Emerson [1988]), the following lines of research should be mentioned:

- the investigation of other (usually more general) models of computation than finite automata: grammars, pushdown automata, Turing machines, Petri nets, etc.,
- the classification theory of sequence (or tree) properties, e.g. by different acceptance modes of automata or by topological conditions,
- algebraic aspects, e.g. the semigroup theoretical analysis of automata over ω-sequences,
- the connection with fixed point calculi,
- the study of more general structures than ω-sequences and ω-trees in connection with automata (e.g. sequences over the integers or certain graphs), and the extension of Rabin's decidability result to stronger theories.

The aim of the paper is to provide the reader with a more detailed overview of these and related developments of the field, based on a self-contained exposition of the central results.

Some topics are only treated in brief remarks or had to be skipped, among them combinatorics on infinite words, the connections with semantics of program schemes, and the discussion of related term based calculi (such as CCS).

Also the references listed at the end of the paper do not cover the subject completely. However, we hope that very few papers will be missed when the reader traces the articles which are cited within the given references.

Acknowledgement

I would like to thank all those who contributed comments and corrections to a draft version of this paper; in particular, A. Arnold, H.D. Ebbinghaus, J. Flum, Y. Gurevich, A.W. Mostowski, D. Niwinski, D. Perrin, J.E. Pin, D. Seese, D. Siefkes, L. Staiger and P. Wolper, as well as students and colleagues from the RWTH Aachen. I also thank Carla Meckler for her endless work in typing and arranging the manuscript.

I. Automata on infinite words

Notation

Usually A denotes a finite alphabet, and A^*, resp. A^ω, stands for the set of finite words, resp. the set of ω-sequences (or: ω-words) over A. Let $A^\infty = A^* \cup A^\omega$. Finite words are indicated by u,v,w,... , the empty word by ε, and sets of finite words by U,V,W,... . Letters α,β,... are used for ω-words and L,L',... for sets of ω-words (i.e., ω-languages). Notations for segments of ω-words are

$$\alpha(m,n) := \alpha(m)\ldots\alpha(n\text{-}1) \quad (\text{for } m \leq n), \text{ and}$$
$$\alpha(m,\omega) := \alpha(m)\alpha(m+1)\ldots .$$

The logical connectives are written ¬, ∧, ∨, →, ∃, ∀. As a shorthand for the quantifiers "there exist infinitely many n" and "there are only finitely many n" we use "$\exists^\omega n$", resp. "$\exists^{<\omega} n$".

The following operations on sets of finite words are basic: For $W \subseteq A^*$ let

$$W^\omega := \{\alpha \in A^\omega \mid \alpha = w_0 w_1 \ldots \text{ with } w_i \in W \text{ for } i \geq 0\},$$
$$\lim W := \{\alpha \in A^\omega \mid \exists^\omega n\ \alpha(0,n) \in W\},$$
$$\text{pref } W := \{u \in A^* \mid \exists v\ uv \in W\}.$$

Finally, for an ω-sequence $\sigma = \sigma(0)\sigma(1)\ldots$ from S^ω the "infinity set" of σ is

$$\text{In}(\sigma) := \{s \in S \mid \exists^\omega n\ \sigma(n) = s\}.$$

1. Büchi automata

Büchi automata are nondeterministic finite automata equipped with an acceptance condition which is appropriate for ω-words: An ω-word is accepted if the automaton can read it from left to right while assuming a sequence of states in which some final state occurs infinitely often ("Büchi acceptance"). More precisely, a Büchi automaton over the finite alphabet A is of the form $\mathcal{A} = (Q, q_0, \Delta, F)$ with finite state set Q, initial state $q_0 \in Q$, transition relation $\Delta \subseteq Q \times A \times Q$, and a set $F \subseteq Q$ of final states. A run of $\mathcal{A}$ on an ω-word $\alpha = \alpha(0)\alpha(1)\ldots$ from A^ω is a sequence $\sigma = \sigma(0)\sigma(1)\ldots$ such that $\sigma(0) = q_0$, $(\sigma(i),\alpha(i),\sigma(i+1)) \in \Delta$ for $i \geq 0$; the run is called successful if some state of F occurs infinitely often in it, i.e. if $In(\sigma) \cap F \neq \emptyset$. $\mathcal{A}$ accepts α if there is a successful run of $\mathcal{A}$ on α. Let

$$L(\mathcal{A}) = \{\alpha \in A^{\omega} \mid \mathcal{A} \text{ accepts } \alpha\}$$

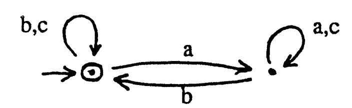
be the ω-language recognized by $\mathcal{A}$. If $L = L(\mathcal{A})$ for some Büchi automaton $\mathcal{A}$, L is said to be Büchi recognizable.

We consider some easy examples over the alphabet $A = \{a,b,c\}$. Define $L_1 \subseteq A^{\omega}$ by

$\alpha \in L_1$ iff after any occurrence of letter a there is some occurrence of letter b in α.

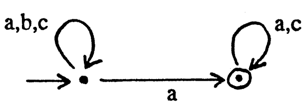
A Büchi automaton recognizing L_1 is (in state graph representation)

$\mathcal{A}_1$:



The complement $\bar{L}_1 = A^{\omega} - L_1$ is recognized by the Büchi automaton

$\bar{\mathcal{A}}_1$:

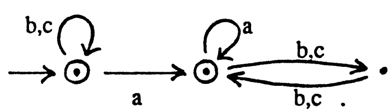


Finally, the ω-language $L_2 \subseteq A^{\omega}$ with

$\alpha \in L_2$ iff between any two occurrences of letter a in α there is an even number of letters b,c

is recognized by

$\mathcal{A}_2$:



For a closer analysis of Büchi recognizable ω-languages we use the following notations, given some fixed Büchi automaton $\mathcal{A} = (Q,q_0,\Delta,F)$: If $w = a_0 \dots a_{n-1}$ is a finite word over A , write $s \xrightarrow[w]{} s'$ if there is a state sequence $s_0,\dots,s_n$ such that $s_0 = s$, $(s_i,a_i,s_{i+1}) \in \Delta$ for $i < n$, and $s_n = s'$. Let

$$W_{ss'} = \{w \in A^* \mid s \xrightarrow[w]{} s'\} .$$

Each of the finitely many languages $W_{ss'}$ is regular. By definition of Büchi acceptance, the ω-language recognized by $\mathcal{A}$ is

(+) $\quad L(\mathcal{A}) = \bigcup_{s \in F} W_{q_0 s} \cdot (W_{ss})^{\omega}$.

This leads to the following basic characterization of Büchi recognizable sets:

1.1. Theorem (Büchi [1962])
An ω-language $L \subseteq A^{\omega}$ is Büchi recognizable iff L is a finite union of sets $U \cdot V^{\omega}$ where $U,V \subseteq A^*$ are regular sets of finite words (and where moreover one may assume $V \cdot V \subseteq V$).

The direction from left to right is clear from equation (+); note that $W_{ss} \cdot W_{ss} \subseteq W_{ss}$. For the converse, we verify the following closure properties of Büchi recognizable sets:

1.2. Lemma

(a) If $V \subseteq A^*$ is regular, then V^{ω} is Büchi recognizable.

(b) If $U \subseteq A^*$ is regular and $L \subseteq A^{\omega}$ is Büchi recognizable, then $U \cdot L$ is Büchi recognizable.

(c) If $L_1, L_2 \subseteq A^{\omega}$ are Büchi recognizable, then $L_1 \cup L_2$ and $L_1 \cap L_2$ are Büchi recognizable.

Proof. (a) Since $V^{\omega} = (V - \{\varepsilon\})^{\omega}$, assume that V does not contain the empty word; further suppose that there is no transition into the initial state q_0 of the finite automaton which recognizes V. A Büchi automaton $\mathcal{A}'$ recognizing V^{ω} is obtained from the given automaton $\mathcal{A}$ by adding a transition (s,a,q_0) for any transition (s,a,s') with $s' \in F$, and by declaring q_0 as single final state of $\mathcal{A}'$.

The claims in (b) and (c) concerning concatenation and union are proved in the same way as for regular sets of finite words. For later use we show closure of Büchi recognizable sets under intersection. Suppose L_1 is recognized by $\mathcal{A}_1 = (Q_1,q_1,\Delta_1,F_1)$ and L_2 by $\mathcal{A}_2 = (Q_2,q_2,\Delta_2,F_2)$. A Büchi automaton recognizing $L_1 \cap L_2$ is of the form $\mathcal{A} = (Q_1 \times Q_2 \times \{0,1,2\}, (q_1,q_2,0),\Delta,F)$, where the transition relation Δ copies Δ_1 and Δ_2 in the first two components of states, and changes the third component from 0 to 1 when an F_1-state occurs in the first component, from 1 to 2 when subsequently an F_2-state occurs in the second component and back to 0 immediately afterwards. Then 2 occurs infinitely often as third component in a run iff some F_1-state and some F_2-state occur infinitely often in the first two components. Hence with $F := Q_1 \times Q_2 \times \{2\}$ we obtain a Büchi automaton as desired. ❑

A representation of an ω-language in the form $L = \bigcup_{i=1}^{n} U_i \cdot V_i^{\omega}$, where the U_i, V_i are given by regular expressions, is called an ω-regular expression. Since the constructions in 1.2 are effective, the conversion of ω-regular expressions into Büchi automata and vice versa can be carried out effectively. Hence Büchi recognizable ω-languages are called regular ω-languages; other terms used in the literature are ω-regular, rational, ω-rational.

As is clear from equation (+), a Büchi automaton $\mathcal{A}$ accepts some word iff $\mathcal{A}$ reaches some final state (say via the word u) which can then be revisited by a loop (say via the word v). The existence of a reachable final state which is located in a loop of $\mathcal{A}$ can be checked by an effective procedure. Hence

1.3. Theorem

(a) Any nonempty regular ω-language contains an ultimately periodic ω-word (i.e., an ω-word of the form uvvv...).

(b) The emptiness problem for Büchi automata is decidable. ❑

Vardi, Wolper [1988] show that the nonemptiness-problem for Büchi automata ("$L(\mathcal{A}) \neq \emptyset$?") is logspace complete for NLOGSPACE, and Sistla, Vardi, Wolper [1987] prove that the nonuniversality problem for Büchi automata ("$L(\mathcal{A}) \neq A^{\omega}$?") is logspace complete for PSPACE.

2. Congruences and Complementation

Closure of Büchi recognizable ω-languages under complement is nontrivial and involves an interesting combinatorial argument. As will be seen, it is not possible to work with a reduction to deterministic Büchi automata.

2.1. Theorem (Büchi [1962])

If $L \subseteq A^{\omega}$ is Büchi recognizable, so is A^{ω}-L. Moreover, from a Büchi automaton recognizing L one can construct one recognizing A^{ω} - L.

For the proof we shall represent both L and A^{ω}-L as finite unions of sets $U \cdot V^{\omega}$ where U and V are regular sets of a special kind, namely classes of a certain congruence relation over A* of finite index. (A congruence is an equivalence relation compatible with concatenation.) Let $L = L(\mathcal{A})$ where $\mathcal{A} = (Q, q_0, \Delta, F)$ is a Büchi automaton. Write $s \overset{F}{\underset{w}{-\!\!->}} s'$ if there is a run of $\mathcal{A}$ on w from state s to state s' such that at least one of the states in the run (including s and s') belongs to F. The set

$$W^F_{ss'} := \{w \in A^* \mid s \overset{F}{\underset{w}{-\!\!->}} s'\}$$

is regular. Now define the equivalence relation $\sim_{\mathcal{A}}$ over A* as follows:

$$u \sim_{\mathcal{A}} v \text{ iff } \forall s,s' \in Q \ (s \underset{u}{-\!\!->} s' \iff s \underset{v}{-\!\!->} s' \text{ and } s \overset{F}{\underset{u}{-\!\!->}} s' \iff s \overset{F}{\underset{v}{-\!\!->}} s')\ .$$

The relation $\sim_{\mathcal{A}}$ is a congruence over A*, which is of finite index by finiteness of Q. Hence each $\sim_{\mathcal{A}}$-class is regular. (The $\sim_{\mathcal{A}}$-class [w] containing the word w is the intersection of the sets $W_{ss'}$ and $W^F_{ss'}$, and A^*-$W_{ss'}$ and A^*-$W^F_{ss'}$, containing w.)

A representation of $L(\mathcal{A})$ and A^{ω}-$L(\mathcal{A})$ in terms of the $\sim_{\mathcal{A}}$-classes will be provided by the following lemma:

2.2. Lemma.

(a) Let $\mathcal{A}$ be a Büchi automaton. For any $\sim_{\mathcal{A}}$-classes U,V: If $U \cdot V^{\omega} \cap L(\mathcal{A}) \neq \emptyset$, then $U \cdot V^{\omega} \subseteq L(\mathcal{A})$. (Hence: If $U \cdot V^{\omega} \cap (A^{\omega}\text{-}L(\mathcal{A})) \neq \emptyset$, then $U \cdot V^{\omega} \subseteq A^{\omega}\text{-}L(\mathcal{A})$.)

(b) Let ~ be a congruence over A* of finite index. For any ω-word $\alpha \in A^{\omega}$ there are ~-classes U,V (even with $V \cdot V \subseteq V$) such that $\alpha \in U \cdot V^{\omega}$.

Part (a) states a "saturation" property of $\sim_{\mathcal{A}}$ with respect to $L(\mathcal{A})$ and A^{ω}-$L(\mathcal{A})$. By definition, a congruence ~ over A* saturates an ω-language $L \subseteq A^{\omega}$ if

$$U \cdot V^{\omega} \cap L \neq \emptyset \text{ implies } U \cdot V^{\omega} \subseteq L, \text{ for all ~-classes } U,V.$$

If ~ saturates L and also is of finite index, then

$$L = \bigcup \{U \cdot V^{\omega} \mid U,V \text{ ~-classes}, U \cdot V^{\omega} \cap L \neq \emptyset\};$$

the inclusion "⊇" holds by saturation and "⊆" follows from part (b) of 2.2. Moreover, since ~ has finite index, the ~-classes are regular and the union is finite; so L is a regular ω-language. For the congruence $\sim_{\mathcal{A}}$, which saturates A^{ω}-$L(\mathcal{A})$ by (a) and is of finite index, we obtain that A^{ω}-$L(\mathcal{A})$ is regular. Note that emptiness of $U \cdot V^{\omega} \cap L(\mathcal{A})$ (and hence also nonemptiness of $U \cdot V^{\omega} \cap (A^{\omega}\text{-}L(\mathcal{A}))$) can be decided effectively by 1.2(c) and 1.3. Thus a Büchi automaton recognizing A^{ω} - $L(\mathcal{A})$ can be constructed effectively from the given automaton $\mathcal{A}$. So Lemma 2.2 suffices to show 2.1.

Proof of 2.2. (a) Suppose $\alpha = uv_1v_2...$ where $u \in U$ and $v_i \in V$ for $i > 0$, and assume further that there is a successful run of $\mathcal{A}$ on α. From this run we obtain states

$s_1, s_2, \ldots$ such that

$$q_0 \xrightarrow[u]{} s_1 \xrightarrow[v_1]{} s_2 \xrightarrow[v_2]{} s_3 \longrightarrow \ldots$$

where we even have

$$s_i \xrightarrow[v_i]{F} s_{i+1} \text{ for infinitely many } i.$$

Let $\beta \in U \cdot V^\omega$ be arbitrary. We show that $\beta \in L(\mathcal{A})$. We have $\beta = u' v'_1 v'_2 \ldots$ where $u' \in U$ and $v'_i \in V$ for $i > 0$. Since U,V are $\sim_{\mathcal{A}}$-classes and hence $u \sim_{\mathcal{A}} u'$, $v_i \sim_{\mathcal{A}} v'_i$, we obtain

$$q_0 \xrightarrow[u']{} s_1 \xrightarrow[v'_1]{} s_2 \xrightarrow[v'_2]{} s_3 \longrightarrow \ldots$$

and

$$s_i \xrightarrow[v'_i]{F} s_{i+1} \text{ for infinitely many } i.$$

This yields a run of $\mathcal{A}$ on β in which some F-state occurs infinitely often. Hence $\beta \in L(\mathcal{A})$.

(b) Let $\sim$ be a congruence of finite index over A^*. Given $\alpha \in A^\omega$, two positions k,k' are said to <u>merge at position m</u> (where $m > k,k'$), if $\alpha(k,m) \sim \alpha(k',m)$. In this case write $k \equiv_\alpha k'$ (m). Note that then also $k \equiv_\alpha k'$ (m') for any $m' > m$ (because $\alpha(k,m) \sim \alpha(k',m)$ implies $\alpha(k,m)\alpha(m,m') \sim \alpha(k',m)\alpha(m,m')$). Write $k \equiv_\alpha k'$ if $k \equiv_\alpha k'$ (m) for some m. The relation $\equiv_\alpha$ is an equivalence relation of finite index over ω (because $\sim$ is of finite index). Hence there is an infinite sequence $k_0, k_1, \ldots$ of positions which all belong to the same $\equiv_\alpha$-class. By passing to a subsequence (if necessary), we can assume $k_0 > 0$ and that for $i > 0$ the segments $\alpha(k_0,k_i)$ all belong to the same $\sim$-class V. Let U be the $\sim$-class of $\alpha(0,k_0)$. We obtain

(*) $\exists k_0\, (\alpha(0,k_0) \in U \wedge \exists^\omega k\, (\alpha(k_0,k) \in V \wedge \exists m\, k_0 \equiv_\alpha k\, (m)))$.

We shall show that (*) implies $\alpha \in U \cdot V^\omega$ and $V \cdot V \subseteq V$ (which completes the proof of (b)). Suppose that k_0 and a sequence $k_1, k_2, \ldots$ are given as guaranteed by (*). Again by passing to an infinite subsequence we may assume that for all $i \geq 0$, the positions $k_0, \ldots, k_i$ merge at some $m < k_{i+1}$ and hence at k_{i+1}. We show $\alpha(k_i,k_{i+1}) \in V$ for $i \geq 0$. From (*) it is clear that $\alpha(k_0,k_1) \in V$. By induction assume that $\alpha(k_j,k_{j+1}) \in V$ for $j < i$. We know $\alpha(k_0,k_{i+1}) \in V$ and that k_0,k_i merge at k_{i+1}. Thus $\alpha(k_i,k_{i+1}) \in V$, as was to be shown. - Finally, in order to verify the claim $V \cdot V \subseteq V$, it suffices to show $V \cdot V \cap V \neq \emptyset$ (since V is a class of a congruence). But this is clear since $\alpha(k_0,k_i)$, $\alpha(k_i,k_{i+1})$ and $\alpha(k_0,k_{i+1})$ belong to V for any $i > 0$. ❑

The use of the merging relation $\equiv_\alpha$ in the preceding proof can be avoided if Ramsey's Theorem is invoked (as done in the original proof by Büchi [1962]): One notes that $\sim$

induces a finite partition of the set $\{(i,j) \mid i < j\}$, by defining that (i,j) and (i',j') belong to the same class iff $\alpha(i,j) \sim \alpha(i',j')$. Now Ramsey's Theorem states that there is an infinite homogeneous set, i.e. a set $\{i_0, i_1, \ldots\}$ such that all pairs (i_k, i_l) with $k < l$ are in one $\sim$-class, in particular all pairs (i_k, i_{k+1}) are in this class. Define V to be this $\sim$-class and let U be the $\sim$-class of $\alpha(0,i_0)$. Then $\alpha \in U \cdot V^\omega$. - We gave the above self-contained proof because condition (*) will be used again in section 4.

By 1.2 and 2.1, the regular ω-languages are effectively closed under boolean operations. As for regular sets of finite words, this implies

2.3. Theorem

The equivalence problem for Büchi automata is decidable.

<u>Proof.</u> Given Büchi automata $\mathcal{A}_1$, $\mathcal{A}_2$, we have $L(\mathcal{A}_1) = L(\mathcal{A}_2)$ iff $(L(\mathcal{A}_1)-L(\mathcal{A}_2)) \cup (L(\mathcal{A}_2)-L(\mathcal{A}_1)) = \emptyset$. A Büchi automaton recognizing the latter language can be constructed from $\mathcal{A}_1, \mathcal{A}_2$ using 1.2 and 2.1, and be tested for emptiness by 1.3. ❑

Let us consider the complexity of the complementation process and the equivalence test. Given a Büchi automaton with n states, there are n^2 different pairs (s,s') and hence $O(2^{2n^2})$ different $\sim_{\mathcal{A}}$-classes. This leads to a size bound of $O(2^{cn^2})$ states for the complement automaton (Pécuchet [1986], Sistla, Vardi, Wolper [1987]). An improved bound of $O(2^{n \cdot \log(n)})$ is given by Safra [1988]; Michel [1988] shows that this bound is optimal. The equivalence problem is considered in Alaiwan [1984] and Sistla, Vardi, Wolper [1987]; they obtain an exponential time, resp. polynomial space bound for its solution. Kurshan [1987] investigates the containment problem for regular ω-languages using a decomposition technique for Büchi automata.

The equivalence problem has also been studied in terms of equations between ω-regular expressions, building on work of Salomaa for classical regular expressions. A sound and complete axiom system, consisting of 8 axioms and 4 rules, is given in Wagner [1976]; other (independent) approaches are found in Izumi, Inagaki, Honda [1984], Darondeau, Kott [1984,1985]. See also Milner [1984]. Representations of regular ω-languages by further operations are studied in Mostowski [1977]. Litovski, Timmerman [1987] consider the generation of ω-powers (sets of the form W^ω); they investigate the structure of the corresponding classes of generators (sets V such that $V^\omega = W^\omega$).

Lemma 2.2 above not only shows complementation for regular ω-languages but also serves as a starting point for an investigation of these ω-languages in terms of finite semigroups. Recall that a language $W \subseteq A^*$ is regular iff there is a finite monoid M and a monoid homomorphism $f: A^* \to M$ such that W is a union of sets $f^{-1}(m)$ where $m \in M$. Since $A^*/\sim_{\mathcal{A}}$ is a finite monoid (for any Büchi automaton $\mathcal{A}$), we obtain from 1.1 and 2.2:

2.4. Theorem

An ω-language $L \subseteq A^\omega$ is regular iff there is a finite monoid M and a monoid homomorphism $f: A^* \to M$ such that L is a union of sets $f^{-1}(m)\cdot(f^{-1}(e))^\omega$ with $m,e \in M$ and where e can be assumed to be idempotent (i.e. satisfying $e\cdot e = e$). ❑

As will be shown in 2.6 below, there is a canonical minimal monoid with this property. In other words, there is a coarsest congruence $\approx_L$ over A^* which saturates L. We introduce $\approx_L$ here together with two other natural congruences associated with an ω-language L. Given $L \subseteq A^\omega$, define for $u,v \in A^*$

$u \sim_L v$ iff $\forall \alpha \in A^\omega (u\alpha \in L \iff v\alpha \in L)$
(a right congruence; Trachtenbrot [1962], Staiger [1983])

$u \equiv_L v$ iff $\forall x \in A^* \forall \alpha \in A^\omega (xu\alpha \in L \iff xv\alpha \in L)$
(Jürgensen, Thierrin [1983])

$u \approx_L v$ iff $\forall x,y,z \in A^* (xuyz^\omega \in L \iff xvyz^\omega \in L$ and $x(yuz)^\omega \in L \iff x(yvz)^\omega \in L)$
(Arnold [1985]).

The first two relations are direct analogues of the congruences associated with sets W of finite words (which are of finite index iff W is regular and represent the minimal automaton for W, resp. its transformation semigroup). The third congruence $\approx_L$ views two finite words as equivalent iff they cannot be distinguished by L as corresponding segments of ultimately periodic ω-words. Regularity of L implies that $\sim_L$, $\equiv_L$, $\approx_L$ are of finite index (since they are all refined by the finite congruence $\sim_{\mathcal{A}}$ if $\mathcal{A}$ is a Büchi automaton which recognizes L). We note that the converse fails:

2.5. Remark (Trachtenbrot [1962])

There are nonregular sets $L \subseteq A^\omega$ such that $\sim_L$, $\equiv_L$, and $\approx_L$ are of finite index.

Proof. For given $\beta \in A^\omega$ let $L(\beta)$ contain all ω-words that have a common suffix with β. Then any two words u,v are $\sim_{L(\beta)}$-equivalent, since for two ω-words uα, vα membership in L(β) does not depend on u,v. So there is only one $\sim_{L(\beta)}$-class; clearly the same is true for $\equiv_{L(\beta)}$. If we choose β to be not ultimately periodic, L(β) is not regular (by 1.3). Furthermore, in this latter case also $\approx_{L(\beta)}$ has only one congruence class. ❑

Staiger [1983] analyses under which additional conditions an ω-language L is regular if $\sim_L$ is of finite index (called "finite state" ω-language there). It is shown that this is true if L belongs to the class $G_\delta \cap F_\sigma$ of the Borel hierarchy (for definition of G_δ, F_σ cf. section 5 below).

We now show the mentioned maximality property of $\approx_L$.

2.6. Theorem (Arnold [1985])

An ω-language L is regular iff $\approx_L$ is of finite index and saturates L; moreover, $\approx_L$ is the coarsest congruence saturating L.

Proof. If $\approx_L$ is of finite index and saturates L, then $L = \bigcup \{U\cdot V^\omega \mid U,V$ are $\approx_L$-classes, $U\cdot V^\omega \cap L \neq \emptyset\}$ and L hence is regular (see the remark following 2.2). Conversely, suppose L is regular; then (as seen before 2.5) $\approx_L$ is of finite index. We show that $\approx_L$ saturates L, i.e. $U\cdot V^\omega \cap L \neq \emptyset$ implies $U\cdot V^\omega \subseteq L$ for any $\approx_L$-classes U,V. Since $U\cdot V^\omega \cap L$ is regular, we can assume that there is an ultimately periodic ω-word xy^ω in $U\cdot V^\omega \cap L$. In a decomposition of xy^ω into a U-segment and a sequence of V-segments, we find two V-segments which start after the same prefix y_1 of the period y; so we obtain $w := xy^m y_1 \in U\cdot V^r$ and $z := y_2 y^n y_1 \in V^s$ for some m,n, r,s and $y_1 y_2 = y$, so that $xy^\omega = wz^\omega$. Denote by [w] and [z] the $\approx_L$-classes of w and z. Since $[w] \cap U\cdot V^r \neq \emptyset$ we have $U\cdot V^r \subseteq [w]$, similarly $V^s \subseteq [z]$, and hence $U\cdot V^\omega \subseteq [w]\cdot[z]^\omega$. It remains to prove $[w]\cdot[z]^\omega \subseteq L$. For contradiction assume there is $\alpha \in [w]\cdot[z]^\omega - L$, say $\alpha = w_0 z_1 z_2 \ldots$ where $w_0 \approx_L w$, $z_i \approx_L z$. Since α may be assumed again to be ultimately periodic, we obtain p,q with $\alpha = w_0 z_1 \ldots z_p (z_{p+1} \ldots z_{p+q})^\omega$. But then from $wz^\omega = xy^\omega \in L$ we know $wz^p(z^q)^\omega \in L$, so $w_0 z_1 \ldots z_p (z_{p+1} \ldots z_{p+q})^\omega \in L$ by definition of $\approx_L$ and thus $\alpha \in L$, a contradiction.

It remains to show that $\approx_L$ is the coarsest among the congruences ~ saturating L. So assume ~ is such a congruence and suppose u ~ v (or: <u> = <v> for the ~-classes of u and v). We verify $u \approx_L v$. We have: $xuyz^\omega \in L$ iff $\langle xuy\rangle\langle z\rangle^\omega \subseteq L$ (since ~ saturates L) iff $\langle xvy\rangle\langle z\rangle^\omega \subseteq L$ (since u ~ v) iff $xvyz^\omega \in L$. Similarly one obtains $x(yuz)^\omega \in L$ iff $x(yvz)^\omega \in L$; thus $u \approx_L v$. ❑

The preceding result justifies calling $A^*/\approx_L$ "the syntactic monoid of L", with concatenation of classes as the product. It allows to classify the regular ω-languages by reference to selected varieties of monoids, extending the classification theory for regular sets of finite words (Eilenberg [1976]). Examples will be mentioned in section 6.

3. The Sequential Calculus

One motivation for considering automata on infinite sequences was the analysis of the "sequential calculus", a system of monadic second-order logic for the formalization of properties of sequences. Büchi [1962] showed the surprising fact that any condition on sequences that is written in this calculus can be reformulated as a statement about acceptance of sequences by an automaton.

For questions of logical definability, an ω-word $\alpha \in A^\omega$ is represented as a model theoretic structure of the form $\underline{\alpha} = (\omega, 0, +1, <, (Q_a)_{a \in A})$, where $(\omega, 0, +1, <)$ is the structure of the natural numbers with zero, successor function, and the usual ordering, and where $Q_a = \{i \in \omega \mid \alpha(i) = a\}$ (for $a \in A$). The corresponding first-order language contains variables x,y,... for natural numbers, i.e. for the positions in ω-words. Typical atomic formulas are "x+1 < y" ("the position following x comes before y") or "$x \in Q_a$" ("position x carries letter a"). In this framework, the example set $L_1 \subseteq \{a,b,c\}^\omega$ of section 1 (containing the ω-words where after any letter a there is eventually a letter b) can be defined by the sentence

$$\varphi_1: \forall x\, (x \in Q_a \rightarrow \exists y\, (x < y \wedge y \in Q_b)).$$

We shall also allow variables X,Y,... for <u>sets</u> of natural numbers and quantifiers ranging over them. For example, they occur in a definition of the ω-language L_2 of section 1 (containing the ω-words where between any two succeeding occurrences of letter a there is an even number of letters b,c):

$$\varphi_2: \forall x \forall y\, (x \in Q_a \wedge y \in Q_a \wedge x < y \wedge \neg \exists z\, (x < z \wedge z < y \wedge z \in Q_a) \rightarrow \\ \exists X (x \in X \wedge \forall z\, (z \in X \leftrightarrow \neg\, z+1 \in X) \wedge \neg\, y \in X)).$$

Note that the set quantifier postulates a set containing every second position starting with position x ; this ensures that the number of letters between positions x and y is even. - The sequential calculus consists of all the conditions on ω-words which can be written in this logical language.

One calls this framework also <u>monadic second-order logic</u> over the signature with 0,+1,<, and the predicates Q_a, due to the quantification over sets, which are unary relations and hence "monadic second-order objects". We shall denote it $S1S_A$ for "<u>s</u>econd-order theory of <u>one</u> <u>s</u>uccessor" with the predicates Q_a for $a \in A$. (Below in 3.1 it will be seen that < is second-order definable in terms of successor and hence inessential).

Formally, $S1S_A$ is built up as follows: <u>Terms</u> are constructed from 0 and the variables x,y,... by applications of "+1", <u>atomic formulas</u> are of the form t = t', t < t', $t \in X$, $t \in Q_a$ (for $a \in A$) where t, t' are terms and X is a set variable, and $S1S_A$-<u>formulas</u> are constructed from atomic formulas using the connectives $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$ and the quantifiers $\exists, \forall$ acting on either kind of variables. We write $\varphi(X_1,\ldots,X_n)$ to indicate that at most the variables $X_1,\ldots,X_n$ occur free in φ (i.e., are not in the scope of a quantifier). Formulas without free variables are called sentences. If φ is a sentence, we write $\underline{\alpha} \models \varphi$ to indicate that φ is satisfied in $\underline{\alpha}$ under the canonical interpretation described above. For instance, if α = abaabaaabaaaab... and φ_1 is as before, we have $\underline{\alpha} \models \varphi_1$. The ω-language defined by an $S1S_A$-sentence φ is

$$L(\varphi) = \{\alpha \in A^\omega \mid \underline{\alpha} \models \varphi\}.$$

In the analysis of monadic second-order definability it is sometimes convenient to cancel the predicate symbols Q_a and use free set variables X_k in their place. The resulting formalism will be called S1S. We then use formulas $\varphi(X_1,\ldots,X_n)$ without the symbols Q_a and interpret them in ω-words over the special alphabet $\{0,1\}^n$. In $\alpha \in (\{0,1\}^n)^\omega$, the formula $x \in X_k$ says that the x-th letter of α has 1 in its k-th component. As an example, consider the following sequence $\alpha \in (\{0,1\}^2)^\omega$, where the letters from $\{0,1\}^2$ are written as columns:

$$\alpha: \begin{matrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \quad \ldots$$

Since in α there are infinitely many letters with first component 1 and second component 0, we have

$$\underline{\alpha} \models \forall x \exists y\, (x < y \wedge y \in X_1 \wedge \neg\, y \in X_2).$$

Formally, we represent $\alpha \in (\{0,1\}^n)^\omega$ by the structure $\underline{\alpha} = (\omega, 0, +1, <, P_1, \ldots, P_n)$ where $P_k = \{i \mid (\alpha(i))_k = 1\}$, writing $\underline{\alpha} \models \varphi(X_1,\ldots,X_n)$ iff φ holds in $\underline{\alpha}$ with P_k as interpretation of X_k. Furthermore, for an S1S-formula $\varphi = \varphi(X_1,\ldots,X_n)$ define $L(\varphi) = \{\alpha \in (\{0,1\}^n)^\omega \mid \underline{\alpha} \models \varphi(X_1,\ldots,X_n)\}$.

By embedding a given alphabet A into a set $\{0,1\}^n$ for suitable n, $S1S_A$-sentences can be reformulated as S1S-formulas $\varphi(X_1,\ldots,X_n)$. (Instead of "$x \in Q_a$" write the corresponding conjunction consisting of formulas "$x \in X_k$" and "$\neg\, x \in X_k$".) Depending on the alphabet under consideration we shall henceforth call an ω-language L simply <u>definable in S1S</u> if for some sentence φ with the symbols Q_a, resp. for some formula $\varphi = \varphi(X_1,\ldots,X_n)$ without the Q_a, we have $L = L(\varphi)$.

3.1. Büchi's Theorem [1962]

An ω-language is definable in S1S iff it is regular.

<u>Proof.</u> The direction from right to left is easy: Let $\mathcal{A} = (Q,q_0,\Delta,F)$ be a Büchi automaton over the alphabet A, and assume $Q = \{0,...,m\}$ and $q_0 = 0$. The existence of a successful run on an ω-word $\alpha \in A^\omega$ can be expressed by the existence of a suitable (m+1)-tuple of sets $Y_0,...,Y_m$; Y_i contains the positions where the run assumes state i. $L(\mathcal{A})$ is defined by the sentence

$$\exists Y_0...\exists Y_m (\bigwedge_{i\neq j} \neg\exists y(y \in Y_i \wedge y \in Y_j) \ \wedge \ 0 \in Y_0$$
$$\wedge \forall x \bigvee_{(i,a,j)\in\Delta} (x \in Y_i \wedge x \in Q_a \wedge x+1 \in Y_j)$$
$$\wedge \bigvee_{i\in F} \forall x \exists y (x < y \wedge y \in Y_i)) .$$

We shall prove the converse for S1S-formulas $\varphi(X_1,...,X_n)$ interpreted in ω-words over $\{0,1\}^n$ as explained above. It is to be shown that the above type of formula represents a normal form for S1S-formulas (Büchi's "automata normal form"). We proceed in two steps: First S1S is reduced to a simpler formalism $S1S_0$ where <u>only</u> second-order variables X_i occur and the atomic formulas are of the form $X_i \subseteq X_j$ ("X_i is a subset of X_j") and $Succ(X_i) = X_j$ ("X_i,X_j are singletons {x}, {y} where x+1 = y"). In a second step an induction over $S1S_0$-formulas $\varphi(X_1,...,X_n)$ shows that $L(\varphi)$ is Büchi recognizable.

<u>(1) Reduction of S1S to $S1S_0$.</u> Carry out the following steps, starting with a given S1S-formula:

(i) Eliminate superpositions of "+1" by rewriting, e.g.,

"$(x+1)+1 \in X$" as "$\exists y\exists z(x+1 = y \wedge y+1 = z \wedge z \in X)$".

(ii) Eliminate the symbol 0 and then the symbol < by rewriting, e.g.,

"$0 \in X$" as "$\exists x(x \in X \wedge \neg\exists y(y < x))$",
"$x < y$" as "$\forall X(x+1 \in X \wedge \forall z(z \in X \rightarrow z+1 \in X) \rightarrow y \in X)$".

We arrive at a formula with atomic formulas of type x = y, x+1 = y, and $x \in X$ only. For the remaining step we use the shortwritings

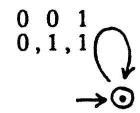
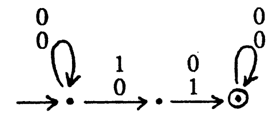
"X = Y" for "$X \subseteq Y \wedge Y \subseteq X$", "$X \neq Y$" for "$\neg X = Y$",
"Sing X" ("X is a singleton") for
"$\exists Y(Y \subseteq X \wedge Y \neq X \wedge \neg\exists Z(Z \subseteq X \wedge Z \neq X \wedge Z \neq Y))$"
("there is exactly one proper subset of X").

(iii) Eliminate first-order variables, by rewriting, e.g.,

"$\forall x\exists y(x+1 = y \wedge y \in Z)$" as
"$\forall X(\text{Sing } X \rightarrow \exists Y(\text{Sing } Y \wedge Succ(X) = Y \wedge Y \subseteq Z))$".

We obtain a $S1S_0$-formula equivalent to the given S1S-formula.

<u>(2) Büchi recognizability of $S1S_0$-definable sets.</u> By induction over $S1S_0$-formulas we show that for any $S1S_0$-formula $\varphi(X_1,...,X_n)$ there is a Büchi automaton $\mathcal{A}$ over $\{0,1\}^n$ with $L(\mathcal{A}) = L(\varphi)$. As typical examples of atomic formulas consider $X_1 \subseteq X_2$ and $Succ(X_1) = X_2$. Corresponding Büchi automata are

 and 

For the induction step it suffices to treat ¬, ∨, and ∃ (since ∧, →, ↔, ∀ are expressible in terms of ¬, ∨, ∃). Cases ¬ and ∨ are clear, by closure of the regular ω-languages under complement and union. Concerning ∃, we have to show closure of the regular ω-languages under <u>projection</u>: Assume, for example, that for the $S1S_0$-formula $\varphi(X_1,X_2)$ a Büchi automaton $\mathcal{A}$ over $\{0,1\}^2$ exists with $L(\mathcal{A}) = L(\varphi)$; we have to find an automaton $\mathcal{A}'$ over {0,1} for the formula $\varphi'(X_1) = \exists X_2\ \varphi(X_1,X_2)$. We obtain $\mathcal{A}'$ by changing the letters of the transitions of $\mathcal{A}$ from $\binom{1}{0}, \binom{1}{1}$ to 1, resp. from $\binom{0}{0}, \binom{0}{1}$ to 0. A successful run of $\mathcal{A}'$ thus "guesses" a second component for the given input $\alpha \in \{0,1\}^\omega$, and for the resulting sequence from $(\{0,1\}^2)^\omega$ it is a successful run of $\mathcal{A}$. Thus $\mathcal{A}'$ recognizes $L(\varphi')$.□

Büchi's Theorem shows that "global" properties of sequences, as formulated in S1S-conditions by means of quantifiers over arbitrary elements and sets, can be given a strictly "operational" meaning, represented by the stepwise working of a Büchi automaton. From a dual point of view, Büchi automata are, despite their conceptual simplicity, a very powerful formalism for the specification of sequence properties.

With minor modifications the above result also holds for sets of <u>finite</u> words. Thus "regular" is equivalent to "monadic second-order definable" for languages as well as for ω-languages. In a finite word w of length k, the variables x,y,... refer to the positions of w (from 1 to k) and the variables X,Y,... to subsets of {1,...,k}. Moreover, the successor function has to be redefined for the maximal element k; we may set, for instance, k+1 = k. Also a suitable convention for the empty word is adopted (it should satisfy universal sentences but not existential sentences). Then with the resulting notion of monadic second-order definability for sets $W \subseteq A^*$ one obtains

3.2. Theorem (Büchi [1960], Elgot [1961])

A set $W \subseteq A^*$ is regular iff it is definable in S1S (interpreted over finite word models).

The proof involves some obvious modifications of 3.1. For example, the formula describing a successful run of an automaton on a finite word now states that at the last position of the word a combination (state, letter) is entered which leads to some final state. ❑

Looking back at the projection step of 3.1, one notes that it works also for a formula $\varphi' = \exists X_1\ \varphi(X_1)$, which is a sentence of S1S, i.e. a formula without free variables. The resulting automaton $\mathcal{A}'$ has unlabelled transitions and in this sense works "input-free". $\mathcal{A}'$ admits a successful run iff the existential sentence φ' is true over the domain ω of the natural numbers. Since the existence of a successful run is effective (see 1.3), this yields:

3.3. Theorem (Büchi [1962])
Truth of sentences of S1S is decidable. ❑

The decision problem for S1S was a main motivation in Büchi's investigations. Automata represented a manageable normal form of S1S-formulas which was simple enough to be decided effectively.

If set quantification refers to finite sets only, one speaks of the weak monadic theory of successor, denoted WS1S. Decidability was shown earlier for WS1S than for S1S (in connection with the characterization of regular sets of finite words, by Büchi [1960], Elgot [1961]); it also follows from 3.3 by an interpretation of WS1S in S1S. (One considers S1S-sentences in which set quantifiers are relativized to finite sets, by writing, e.g., "$\exists X\ \varphi(X)$" as "$\exists X(\exists y \forall x(x \in X \rightarrow x < y) \wedge \varphi(X))$", and noting that $<$ can be eliminated as before when using only finite-set quantifiers.) WS1S was the first decidable theory shown to be intractable in the sense that there is no polynomially time bounded (even elementary time bounded) decision procedure for it (Meyer [1975]).

Theorem 3.3 can be used to derive further interesting decidability results. For example, one concludes that Presburger arithmetic (the first-order theory of additon over ω) is decidable. The idea is to represent numbers in binary notation and interpret the resulting (finite) 0-1-sequences by (finite) sets. It is easy to define the relation $x+y = z$ in terms of this interpretation; one simply describes the process of digit by digit calculation of the sum (using successor to proceed to the next digit, and some auxiliary set variable to represent the carry). In this way sentences in the language of Presburger arithmetic can be translated into the language of WS1S where of course number quantifiers become set quantifiers. Now the decision procedure for WS1S can be applied. As noted already by Büchi [1962], this idea extends to the case where infinite binary expansions of real numbers are considered. Thus also the first-order theory of (R,+), the theory of addition of the reals, is decidable. Moreover, the fact that any S1S-definable set of 0-1-sequences contains (by 1.3) an ultimately periodic sequence (which is the expansion of a rational number) can be used to show that the first-order theories of (R,+) and (Q,+) coincide.

The decidability result on S1S has been extended in many directions. One of them, Rabin's "tree theorem" (stating decidability of the monadic theory S2S of two successor functions, i.e. of the binary tree) will be discussed in section 11 below. Another kind of extension is concerned with addition of further number theoretic relations or functions to the theory S1S (besides successor and order). Elgot, Rabin [1966] showed that one may add the unary predicates "is a factorial" or "is a power of k" (for $k \geq 2$) without destroying decidability. However, by Robinson [1958], S1S enriched by the function $x \mapsto 2x$ already allows to interpret full first-order arithmetic and thus is undecidable. More recent results are discussed in Semenov [1984].

Büchi [1973] extended his proof of 3.1 to transfinite ordinals and showed that the monadic second-order theory of $(\omega_1,<)$ is decidable. Automata working on transfinite sequences were further considered in Choueka [1978], Wojciechowski [1985]. Siefkes [1970] and Büchi, Siefkes [1973] presented axiomatizations of the monadic theories of $(\omega,<)$ and $(\omega_1,<)$. In subsequent work of Gurevich, Magidor and Shelah [1983] it became clear that from ω_2 onwards the monadic second-order theory of an ordinal depends on set theoretic hypotheses.

A fundamental progress in the study of monadic theories was made by Shelah [1975]. He developed a model theoretic technique for obtaining decidability results, which does not refer to automata and is applicable to a larger class of structures. A central idea in this approach is to "compose" a finite fragment of the theory of an ordering (given say by the formulas up to some quantifier-depth) from the corresponding theory-fragments of suborderings and the way these suborderings are arranged. In a series of intricate papers Gurevich and Shelah applied the method to show strong decidability results, in particular concerning dense orderings and trees. However, again based on Shelah [1975], they also showed that the monadic second-order theory of (R,<), the ordering of the real numbers, is undecidable. For an exhaustive presentation of the subject we recommend the survey Gurevich [1985].

Recently the automata theoretic aspects of the monadic theory of the integers, i.e. of the ordering (Z,<), have attracted increasing attention. This theory is closely related to problems in ergodic theory and symbolic dynamics (cf. Blanchard, Perrin [1980]). Besides this, interesting phenomena have to be dealt with which are not present in the theory of ω-languages. In particular, words over the ordering of the integers ("biinfinite words") have no distinguished position, like the first position of an ω-word. Thus two Z-words are identified if they can be transformed into each other by a finite shift; and a natural automaton model does not refer to some "start position" but works on Z-words from left to right, "coming from infinity" and "going to infinity". A detailed development of the theory of regular Z-languages, establishing results analogous to the case of ω-languages, is given in Nivat, Perrin [1986], Beauquier [1984], and Perrin, Schupp [1986].

4. Determinism and McNaughton's Theorem

A simple argument, given in 4.2 below, shows that deterministic Büchi automata are not closed under complement and hence strictly weaker than Büchi automata in general. Nevertheless, by refining the notion of acceptance, it is possible to define a type of deterministic automaton which characterizes the regular ω-languages. In the present section we discuss this fundamental determinization theorem, due to McNaughton [1966].

If a deterministic finite automaton $\mathcal{A}$, which recognizes the set $W \subseteq A^*$, is used as a Büchi automaton, it accepts an ω-word α iff infinitely many prefixes of α lead $\mathcal{A}$ to a final state, i.e. belong to W. Collecting these α, we obtain the "limit" set

$$\lim W := \{\alpha \in A^\omega \mid \exists^\omega n\ \alpha(0,n) \in W\}.$$

By the mentioned conversion of finite automata into deterministic Büchi automata and vice versa we have immediately:

4.1. Remark

An ω-language $L \subseteq A^\omega$ is recognized by a deterministic Büchi automaton iff $L = \lim W$ for some regular set $W \subseteq A^*$. ❑

The following example shows that not every regular ω-language is of this form; at the same time we see that closure under complement fails for deterministic Büchi automata:

4.2. Example (Landweber [1969])

Let $A = \{a,b\}$ and $L := \{\alpha \in A^\omega \mid \alpha$ contains only finitely many letters a$\}$ (i.e., $L = A^\omega - \lim(b^*a)^*$). Then L is not of the form lim W with $W \subseteq A^*$.

Proof. Assuming $L = \lim W$ one obtains a contradiction as follows: For some n_1,

$b^{n1} \in W$ (because $b^\omega \in L = \lim W$). For this n_1 there is some n_2 such that $b^{n1}ab^{n2} \in W$ (because $b^{n1}ab^\omega \in L = \lim W$). Proceeding in this way, one obtains a sequence of words $b^{n1}ab^{n2}...ab^{nk} \in W$ $(k = 1,2,...)$. Hence the ω-word $b^{n1}ab^{n2}...$ is in lim W and thus in L, contradicting the definition of L. ❑

A suitably generalized acceptance condition for deterministic automata on ω-words which captures the power of Büchi automata was defined by Muller [1963] (in connection with a problem in asynchronous switching theory). A Muller automaton over the alphabet A is of the form $\mathcal{A} = (Q,q_0,\delta,\mathcal{F})$ where Q is finite, $q_0 \in Q$, $\delta: Q \times A \to Q$ is the (deterministic) transition function, and $\mathcal{F} \subseteq 2^Q$ a collection of final state-sets. $\mathcal{A}$ accepts an ω-word α if those states which $\mathcal{A}$ assumes infinitely often in its unique run σ on α form a set occurring in $\mathcal{F}$, i.e. $\mathrm{In}(\sigma) \in \mathcal{F}$. An ω-language $L \subseteq A^\omega$ is called Muller recognizable if it consists of all ω-words over A accepted by some Muller automaton. (There is also a nondeterministic version of Muller automaton, where a transition relation $\Delta \subseteq Q \times A \times Q$ replaces δ, and acceptance means existence of a run σ as described above. The ω-languages recognized by nondeterministic Muller automata are definable in S1S - by the same idea as in Theorem 3.1 - and hence obviously coincide with the regular ω-languages. In the sequel we consider only the deterministic version.)

Each deterministic Büchi automaton $\mathcal{A} = (Q,q_0,\delta,F)$ is equivalent to a Muller automaton, namely to the automaton $\mathcal{A}' = (Q,q_0,\delta,\mathcal{F})$ where $\mathcal{F}$ consists of all subsets of Q having a nonempty intersection with F. Furthermore, the Muller recognizable ω-languages are closed under boolean operations: If $\mathcal{A} = (Q,q_0,\delta,\mathcal{F})$ recognizes L, then $(Q,q_0,\delta,2^Q-\mathcal{F})$ recognizes A^ω-L. Given $\mathcal{A} = (Q,q_0,\delta,\mathcal{F})$ and $\mathcal{A}' = (Q',q'_0,\delta',\mathcal{F}')$ recognizing L and L', respectively, $L \cup L'$ is recognized by the product automaton of $\mathcal{A}$ and $\mathcal{A}'$ where the collection of final state sets contains $\{(q_1,q'_1),...,(q_n,q'_n)\}$ iff $\{(q_1,...,q_n\} \in \mathcal{F}$ or $\{q'_1,...,q'_n\} \in \mathcal{F}'$. These observations, together with 4.1, yield the direction from right to left of the following lemma:

4.3. Lemma

An ω-language $L \subseteq A^\omega$ is Muller recognizable iff L is a boolean combination (over A^ω) of sets lim W where $W \subseteq A^*$ is regular.

Proof. For the direction from left to right, consider a Muller automaton $\mathcal{A} = (Q,q_0,\delta,\mathcal{F})$ recognizing L; write W_q for the set of finite words recognized by the finite automaton $(Q,q_0,\delta,\{q\})$. By definition, $\mathcal{A}$ accepts α iff for some $F \in \mathcal{F}$, α belongs to $\lim W_q$ for all $q \in F$ and α does not belong to $\lim W_q$ for all $q \in Q$-F. Thus we get the desired representation

$$L = \bigcup_{F\in \mathcal{F}} \left(\bigcap_{q\in F} \lim W_q \cap \bigcap_{q\in Q-F} \sim\lim W_q \right). \quad \square$$

We now show that Muller automata and Büchi automata are equivalent in recognition power. Note that this reproves closure of Büchi recognizable sets under complement. The main difficulty lies in the fact that a membership test "$\alpha \in U\cdot V^{\omega}$?" as performed by a Büchi automaton, i.e. the test whether α can be split into segments $uv_1v_2v_3...$ with $u \in U$ and $v_i \in V$, involves "unlimited guessing" in choosing the segments v_i; this has to be reduced to a deterministic procedure which should depend only on finite information about finite prefixes of α.

4.4. McNaughton's Theorem [1966]

An ω-language is regular (i.e., Büchi recognizable) iff it is Muller recognizable.

The direction from right to left follows from 4.3 and the closure properties for Büchi automata. For the converse, it will suffice, again by 4.3, to show:

4.5. Theorem

Every Büchi recognizable ω-language is a finite union of sets of the form $\lim W \cap \sim\lim W'$ where $W,W' \subseteq A^*$ are regular.

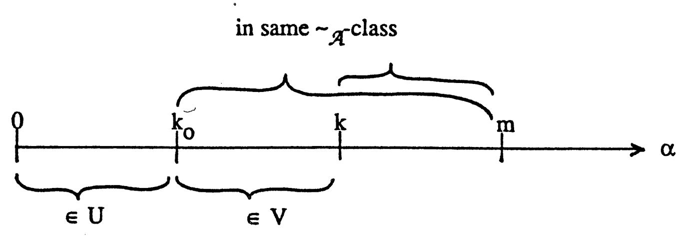
Before turning to the proof, we mention that 4.5 motivates also a modified version of Muller automaton, the sequential <u>Rabin automaton</u> introduced by Rabin [1972]. Here the collection $\mathcal{F}$ of final sets (in a Muller automaton $(Q,q_0,\delta,\mathcal{F})$) is replaced by a collection $\Omega = \{(L_1,U_1),...,(L_n,U_n)\}$ of "accepting pairs" (L_i,U_i) where $L_i,U_i \subseteq Q$. The acceptance condition for a run of the resulting automaton states that for some $i \in \{1,...,n\}$ the L_i-states occur only finitely often but some U_i-state occurs infinitely often in the run. If W_i (resp. W_i') is the regular language recognized by the finite automaton (Q,q_0,δ,U_i) (resp. (Q,q_0,δ,L_i)), then the automaton (Q,q_0,δ,Ω) recognizes the ω-language

$$\bigcup_{i=1}^{n} (\lim W_i \cap \sim\lim W_i'),$$

i.e. a language as required in 4.5. From 4.3 and 4.5 it follows that the Rabin automata (Q,q_0,δ,Ω) are equivalent to Muller automata.

<u>Proof of 4.5.</u> Let $\mathcal{A}$ be a Büchi automaton. By Lemma 2.2 it suffices to consider the case $L(\mathcal{A}) = U\cdot V^{\omega}$ with $\sim_{\mathcal{A}}$-classes U,V and $V\cdot V \subseteq V$. We use the notation introduced in the proof of 2.2, in particular the merging relation $\equiv_\alpha$. (By $k \equiv_\alpha k'$ (m) we mean here that $\alpha(k,m) \sim_{\mathcal{A}} \alpha(k',m)$.) The condition $\alpha \in U\cdot V^{\omega}$ was shown to be equivalent to

(*) $\exists k_0 (\alpha(0,k_0) \in U \wedge \exists^{\omega} k (\alpha(k_0,k) \in V \wedge \exists m\, k_0 \equiv_\alpha k\, (m)))$.



Call a segment $\alpha(k_0,m)$ a <u>V-witness</u> if for some k with $k_0 < k < m$, m is the <u>smallest</u> position such that $\alpha(k_0,k) \in V$ and $k_0 \equiv_\alpha k$ (m). It is not difficult to verify that the set $W_V \subseteq A^*$ of V-witnesses is regular. Since for any k as described in (*) there is a unique V-witness, we obtain that (*) is equivalent to

(**) $\exists k_0 (\alpha(0,k_0) \in U \wedge \exists^{\omega} m\, \alpha(k_0,m)$ is a V-witness),

in other words:

$$\alpha \in U\cdot \lim W_V.$$

Our aim is to rewrite (**) as a boolean combination of conditions "$\exists^{\omega} m\, \alpha(0,m) \in W$" (with regular W); so we want to exchange the quantifiers $\exists k_0$ and $\exists^{\omega} m$ in (**). In the desired condition of the form "$\exists^{\omega} m \exists k_0$ ..." we have to ensure that k_0 may be chosen fixed (independent of the m). The idea is to postulate infinitely many prefixes $\alpha(0,m)$ which admit a decomposition $\alpha(0,m) = \alpha(0,k_0)\alpha(k_0,m)$ with $\alpha(0,k_0) \in U$ and $\alpha(k_0,m) \in W_V$, and to guarantee that only finitely many choices of k_0 occur while m increases. A simple approach would be to refer always to the smallest $k_0 < m$ with $\alpha(0,k_0) \in U$. However, this k_0 might not have the property that there are infinitely many m with $\alpha(k_0,m) \in W_V$: It may happen that only some greater k_1 with $\alpha(0,k_1) \in U$, which does not merge with k_0, has this property. Suppose there are exactly r $\equiv_\alpha$-classes with elements k such that $\alpha(0,k) \in U$, i.e. there are r positions, say $k_1,...,k_r$, in α which pairwise do not merge and satisfy $\alpha(0,k_i) \in U$ ("case r"). Then k_0 may be chosen as one of these positions. So we require in "case r" for infinitely many m the existence of $k_1,...,k_r < m$ which pairwise do not merge at m and satisfy $\alpha(0,k_1) \in U$. We express that for increasing m these k_i remain below a fixed finite bound (so that only finitely many choices of k_0 are possible) by saying that for only finitely many m a new choice of $k_1,...,k_r$ (or just of the maximal position k_r) is necessary.

To give a precise formulation, let us call k <u>r-appropriate for</u> m iff $k = k_r$ for some r-tuple $(k_1,...,k_r)$ with $0 < k_1 < ... < k_r < m$ such that $\alpha(0,k_i) \in U$ for all k_i and not

$k_i \equiv_\alpha k_j$ (m) for $k_i \neq k_j$. If in addition $\alpha(k_i,m) \in W_V$ for some k_i, we say that k is r-appropriate for m by W_V. Finally, a new k r-appropriate for m is a number which is r-appropriate for m and fails to be r-appropriate for any $m' < m$. Then, assuming "case r", (**) amounts to the following:

$(***)_r$ $\quad \exists^\omega m$ [there exists k r-appropriate for m by W_V]
$\wedge \exists^{<\omega} m$ {there exists a new k r-appropriate for m}.

Both conditions [...] and {...} above depend only on the segment $\alpha(0,m)$; indeed, by a tedious programming exercise one can design two finite automata accepting exactly those words $\alpha(0,m)$ satisfying these requirements. (As an alternative, one may note their definability in WS1S and apply 3.2.) Denote by W_r and W_r' the (regular) sets of words $\alpha(0,m)$ satisfying [...], resp. {...}. Then $(***)_r$ says that

$$\alpha \in \lim W_r \cap \sim \lim W_r'.$$

The disjunction of the conditions $(***)_r$ over all possible r is equivalent to (**). Since r cannot exceed the finite number n of different $\sim_{\mathcal{A}}$-classes, we obtain

$$\bigcup_{r=1}^{n} (\lim W_r \cap \sim \lim W_r')$$

as the desired representation of $U \cdot V^\omega$. ❑

McNaughton's Theorem and its proof have interesting consequences, among them further characterizations of the regular ω-languages. For example, parts (c) and (d) of the following result offer a simple inductive construction of the regular sequence sets and show the surprising fact that S1S and the weak theory WS1S have the same expressive power:

4.6. Theorem

For an ω-language $L \subseteq A^\omega$ the following conditions are equivalent:

(a) L is regular.

(b) L is a finite union of sets $U \cdot \lim W$ where $U,W \subseteq A^*$ are regular.

(c) L can be obtained from A^ω by finitely many applications of union, complement (w.r.t. A^ω) and concatenation with a regular set $W \subseteq A^*$ on the left.

(d) L is definable in WS1S.

Proof. Implication (a) → (b) is clear from the proof of 4.5 (cf. condition (**)). For (b) → (c) it suffices to represent lim W as stated in (c), supposing that W is regular. Let (Q,q_0,δ,F) be a finite automaton recognizing W; set $W_{pq} := \{w \in A^* \mid \delta(p,w) = q\}$. We have $\lim W = \sim L$ where L contains all ω-words which have only finitely many prefixes in W, i.e. no prefix in W or a last prefix in W. Note that $\alpha(0,m)$ is a last

prefix of α in W iff for some state $p \in F$, $\delta(q_0,\alpha(0,m)) = p$ and there is no $n > m$ and no $q \in F$ with $\delta(p,\alpha(m,n)) = q$. This yields the equation

$$L = \sim(W \cdot A^\omega) \cup \bigcup_{p \in F}(W_{q_0 p} \cdot \sim(\bigcap_{q \in F}(W_{pq} \cap A^+) \cdot A^\omega)).$$

So L and hence lim W are represented as desired.

Implications (c) → (a) and (d) → (a) are obvious; so it remains to show (a) → (d). By McNaughton's Theorem, a regular ω-language L is a boolean combination of sets lim W with regular W. From a WS1S-formula $\varphi(\bar{X})$ which defines W by 3.2 we obtain a WS1S-formula $\psi(\bar{X},y)$ which expresses over ω-words that the prefix up to position y satisfies $\varphi(\bar{X})$ (use relativization to the positions < y). Now L is definable in WS1S by a boolean combination of formulas $\forall x \exists y (x < y \wedge \psi(\bar{X},y))$. ❑

There are several fully worked-out expositions of McNaughton's Theorem; we mention Rabin [1972], Schützenberger [1973], Büchi [1973], Trachtenbrot, Barzdin [1973], Choueka [1974], Eilenberg [1974]. The above proof of 4.5 follows Thomas [1981]. A very elegant automaton construction appears in Safra [1988]; it yields for a Büchi automaton with n states a deterministic automaton, accepting by Rabin's condition, with $O(2^{n \cdot \log(n)})$ states and n accepting pairs. Characterization 4.6(b) of the regular ω-languages is implicit in McNaughton [1966] and was given by Choueka [1974] and Eilenberg [1974]; it shows that Büchi automata with very restricted nondeterminism suffice to recognize the regular ω-languages (cf. also Karpinski [1975]). Arnold [1983a] proves a related result on non-ambiguous automata: Any regular ω-language is recognized by a Büchi automaton which admits on any given input at most one successful run. Condition 4.6(c), noted in Thomas [1979] in a logical setting, was given a simplified proof by Choueka, Peleg [1983]. Characterization 4.6(d) appears in Thomas [1980], where also a generalization to arbitrary limit ordinals is shown.

Finally, we mention that the question of determinism has also been studied for finite automata working in a reverse direction (from right to left) on ω-words: Mostowski [1982] and Beauquier, Perrin [1985] show that "codeterministic automata" are as powerful as Büchi automata. The equivalence of (nondeterministic) two-way automata on ω-words to Büchi automata is shown in Pécuchet [1985].

5. Acceptance conditions and Borel classes

Definability of sequence sets is a traditional subject in set theoretic topology and descriptive set theory. Automata on ω-words constitute a special type of "finitary" definability of sequence sets and lead to combinatorial problems not considered in the classical mathematical literature. (McNaughton's Theorem is an example.) Conversely, several basic topological notions, in particular in connection with the Borel hierarchy, have a natural meaning in the context of automata and help to systematize the acceptance conditions. In the present section we introduce the classification of acceptance modes for automata in a topological setting, and indicate applications to sequence properties which arise in distributed systems. For a more complete discussion see the surveys Hoogeboom, Rozenberg [1986] and Staiger [1987a].

We refer to the Cantor topology on the set A^ω, where A is a finite alphabet. This topology may be characterized in several equivalent ways:

- as the product topology of the discrete topology on A,
- by declaring as open sets all ω-languages $W \cdot A^\omega$ with $W \subseteq A^*$, or the sets $\{w\} \cdot A^\omega$, as a basis,
- by the metric $d: A^\omega \times A^\omega \to R$, given by

$$d(\alpha,\beta) = \begin{cases} 0 & \text{if } \alpha = \beta \\ 1/2^n \text{ with } n = \min\{i \mid \alpha(i) \neq \beta(i)\} & \text{else}. \end{cases}$$

Taking the example $A = \{0,1\}$, the elements of the space A^ω may be considered as paths through the infinite binary tree (where "0" means "branch left" and "1" means "branch right"). Two paths α,β are close to each other if they have a long common initial segment. The so-called Cantor discontinuum is a linearly ordered representation of the space, corresponding to the left to right ordering of the paths in the binary tree.

Following classical terminology we denote by G the class of open ω-languages and by F ("fermé") the class of closed ω-languages. The Borel hierarchy is obtained by taking alternately countable intersections and unions starting with ω-languages in G or F. One denotes by G_δ (F_σ) the countable intersections (resp. unions) of sets in G (resp. F), by $G_{\delta\sigma}$ ($F_{\sigma\delta}$) the countable unions (resp. intersections) of sets in G_δ (resp. F_σ), etc. Then a hierarchy of the form

$$\begin{array}{lllll} \text{open} = & G & G_\delta & G_{\delta\sigma} & \\ & \times & \times & \times & \ldots \\ \text{closed} = & F & F_\sigma & F_{\sigma\delta} & \end{array}$$

results, where each line indicates a proper inclusion. A set L belongs to some class

in the hierarchy (say $L \in G_\delta$) iff its complement is in the dual class (in the example, $\sim L \in F_\sigma$).

Our aim is to locate the regular ω-languages in the Borel hierarchy. As a preparation note two useful facts about open, resp. closed sets. First, an open set $W \cdot A^\omega$ is equal to $W_o \cdot A^\omega$ where W_o contains all words from W with no proper prefix in W. Any two words in W_o are then incomparable w.r.t. the prefix relation; if this holds and $W \cdot A^\omega = W_o \cdot A^\omega$ we say that W_o is a minimal basis for $W \cdot A^\omega$. The property of being a minimal basis will not be destroyed if we replace, for some n, a word $w \in W_o$ by all words in $\{w\} \cdot A^n$. Using thus longer and longer words in minimal bases, it is possible to represent a sequence $W_1 \cdot A^\omega$, $W_2 \cdot A^\omega$, ... of open sets by a sequence of pairwise disjoint minimal bases W'_i.

Secondly, note that the closed sets in the Cantor topology do not coincide with the sets $\lim W \subseteq A^\omega$. The topological closure of a set L consists of all sequences that have arbitrary long common prefixes with ω-words in L. Thus the set $L = \lim a^*bb^*$ is not closed since a^ω is in the closure of L but does not belong to L itself. The sets $\lim W$ are more general than closed sets in the following sense:

5.1. Remark

$L = \lim W$ for some $W \subseteq A^*$ iff L is in G_δ.

Proof. Assume first $L \in G_\delta$, i.e. $L = \bigcap_{n \geq o} W_n \cdot A^\omega$. W.l.o.g. we have $W_o \supseteq W_1 \supseteq \ldots$ (if not, replace W_n by $\bigcap_{i \leq n} W_i$). Choose a sequence W'_o, W'_1,... of disjoint minimal bases (as explained above) of $W_o \cdot A^\omega$, $W_1 \cdot A^\omega$,... and set $W := \bigcup_{n \geq o} W'_n$. Then $\alpha \in \lim W$ iff α has a prefix in W'_n for infinitely many n iff α has a prefix in W_n for infinitely many n iff (by monotonicity) $\alpha \in \bigcap_{n \geq o} W_n \cdot A^\omega$, i.e. $\alpha \in L$. Conversely, suppose that $L = \lim W$. Denote by $A^{\geq n}$ the set of words of length $\geq n$ over A. We have

$$\lim W = \bigcap_{n \geq o} (W \cap A^{\geq n}) \cdot A^\omega$$

and hence $L \in G_\delta$. ❑

For a discussion of the topology induced by the lim-operation as topological closure see Redziejowski [1985]. - By 5.1 and McNaugthon's Theorem, the regular ω-languages occur very low in the Borel hierarchy:

5.2. Theorem

Every regular ω-language L is a boolean combination of G_δ-(and/or F_σ-)sets; in particular, $L \in G_{\delta\sigma} \cap F_{\sigma\delta}$. ❑

We now discuss some interesting refinements of this result due to Landweber [1969], which yield a characterization of the regular sets $L \subseteq A^\omega$ occurring on the levels G, F, G_δ, F_σ of the Borel hierarchy, and establish effective procedures for deciding whether a regular set is of one of these types.

Let $\mathcal{A} = (Q,q_0,\delta,F)$ a deterministic finite automaton over A and $\alpha \in A^\omega$. $\mathcal{A}$ <u>1-accepts</u> α iff $\exists n\ \delta(q_0, \alpha(0,n)) \in F$, and $\mathcal{A}$ <u>2-accepts</u> α iff $\exists^\omega n\ \delta(q_0,\alpha(0,n)) \in F$. Thus 2-acceptance is Büchi acceptance for deterministic automata. The dual notions, called 1'- and 2'-acceptance, arise by changing $\exists n$ to $\forall n$ and $\exists^\omega n$ to "for almost all n"; by 3-acceptance the recognition by Muller automata is meant. Now $L \subseteq A^\omega$ is called 1- (resp. 2-) recognizable if L consists of the ω-words 1- (resp. 2-) accepted by some deterministic automaton. Since L is 1- (resp. 2-) recognizable iff $L = W \cdot A^\omega$ (resp. $L = \lim W$) for some regular set W, one calls such ω-languages regular-open (resp. regular-G_δ). The following result shows in particular that regular ω-languages which are open are in fact regular-open, similarly for G_δ:

5.3. Theorem (Landweber [1969])

(a) A regular ω-language is in G iff it is 1-recognizable.

(b) A regular ω-language is in G_δ iff it is 2-recognizable.

(c) It is decidable whether a regular ω-language (represented say by a Muller automaton) is 1-recognizable, resp. 2-recognizable.

Analogously, F and F_σ can be characterized in terms of 1'- and 2'-recognizability. Moreover, as Staiger, Wagner [1974] showed, $G_\delta \cap F_\sigma$ contains exactly the boolean combinations of 1-recognizable ω-languages. - 5.3 will follow immediately from

5.4. Lemma

There are effective procedures transforming any Muller automaton $\mathcal{A}$ into a Muller automaton $\mathcal{A}_1$, resp. $\mathcal{A}_2$, such that $L(\mathcal{A}_1)$ is 1-recognizable, $L(\mathcal{A}_2)$ is 2-recognizable, and

$$L(\mathcal{A}) \in G \text{ iff } \mathcal{A} = \mathcal{A}_1 ; \quad L(\mathcal{A}) \in G_\delta \text{ iff } \mathcal{A} = \mathcal{A}_2.$$

<u>Proof.</u> Assume $\mathcal{A} = (Q,q_0,\delta,\mathcal{F})$ is a Muller automaton, w.l.o.g. such that each $q \in Q$ is reachable from q_0 (i.e., there is $w \in A^*$ with $\delta(q_0,w) = q$). By a "loop of $\mathcal{A}$" we shall mean a strongly connected subset of $\mathcal{A}$ (considering $\mathcal{A}$ as a directed graph).

(a) Call a state q of $\mathcal{A}$ an $\mathcal{F}$-state iff q is located in a loop of $\mathcal{A}$ which forms a set from $\mathcal{F}$. Define $\mathcal{A}_1 := (Q,q_0,\delta,\mathcal{F}_1)$ by extending $\mathcal{F}$ to $\mathcal{F}_1$ such that $\mathcal{F}_1$ contains all loops of $\mathcal{A}$ which are reachable from some $\mathcal{F}$-state of $\mathcal{A}$. Let F_1 be the union of the sets $F \in \mathcal{F}_1$. $\mathcal{A}_1$ is equivalent to (Q,q_0,δ,F_1) with 1-acceptance. Namely, some state $q \in F_1$ is reached by (Q,q_0,δ,F_1) on input α iff $\mathcal{A}_1$ assumes on α ultimately the states of a loop from $\mathcal{F}_1$. It remains to show that $L(\mathcal{A}) \in G$ implies $\mathcal{A} = \mathcal{A}_1$. For this we have to verify $\mathcal{F}_1 \subseteq \mathcal{F}$. Let $F \in \mathcal{F}_1$; so there is an $\mathcal{F}$-state q of $\mathcal{A}$ from which the loop F can be reached. Since q is an $\mathcal{F}$-state, we can choose a sequence α inducing $\mathcal{A}$ to assume a loop from $\mathcal{F}$ in which q is located. By assumption, $L(\mathcal{A}) = W \cdot A^\omega$ for suitable W; so some prefix w of α is in W. Since <u>all</u> sequences $w\beta$ are in $L(\mathcal{A})$, any loop reachable from q, and hence F, is realized via some ω-word in $L(\mathcal{A})$. Thus $F \in \mathcal{F}$.

(b) $\mathcal{A}_2$ is constructed from $\mathcal{A}$ by enlarging $\mathcal{F}$ to $\mathcal{F}_2$ as follows: For any state q in a loop forming a set $F \in \mathcal{F}$, and any set E forming some loop containing q, add $F \cup E$ to $\mathcal{F}_2$. $L(\mathcal{A}_2)$ is 2-recognizable: For this we construct a Büchi automaton working as $\mathcal{A}$ does, but further equipped with a "state memory" S which collects the visited states and is set back to $\emptyset$ whenever S includes some $\mathcal{F}$-set. So the states are of form $(q,S) \in Q \times 2^Q$, and from (q,S) the Büchi automaton passes to state $(\delta(q,a),S \cup \{\delta(q,a)\})$ by letter a if $S \cup \{\delta(q,a)\}$ does not include a set from $\mathcal{F}$, otherwise the transition goes to $(\delta(q,a), \emptyset)$. The initial state is $(q_0,\emptyset)$, and the states $(q,\emptyset)$ are the final ones. 2-acceptance for this automaton means that some loop extending a loop forming an $\mathcal{F}$-set is ultimately assumed on the given input, i.e. that $\mathcal{A}_2$ accepts. Suppose now $L(\mathcal{A}) \in G_\delta$, say $L(\mathcal{A}) = \lim W$, and consider two loops F and E of $\mathcal{A}$ as above with common state q. We show $F \cup E \in \mathcal{F}$ (and hence $\mathcal{F} = \mathcal{F}_2$). Pick w such that $\delta(q_0,w) = q$. Via the loop F we can extend w to a sequence α of $L(\mathcal{A})$ (= lim W) and hence reach some finite prefix of α in W, say $wu_1 \in W$. From wu_1 we may complete the loop F back to state q (say via v_1) and continue further through the loop E, again back to q (say via w_1). Repeating the process, we obtain a sequence $wu_1v_1w_1u_2v_2w_2$ ... which is in lim W and causes $\mathcal{A}$ to assume ultimately the states in $F \cup E$. Hence $F \cup E \in \mathcal{F}$.
❑

The notions of 1-, 2-, and 3-acceptance have been investigated for various other machine models (besides finite automata), for example pushdown automata (Cohen, Gold [1977]), deterministic pushdown automata (Cohen, Gold [1978], Linna [1977]), Petri nets (Valk [1983]), and Turing machines (Cohen, Gold [1978a,1980], Lindsay [1986]). Staiger [1986] contains a comprehensive study of recursive (i.e. Turing machine recognizable) ω-languages and their classification in the Borel hierarchy as well as in the arithmetical hierarchy of recursion theory.

In the domain of finite automata theory, Landweber's Theorem has been extended by consideration of further modes of acceptance, also using nondeterministic automata (Staiger, Wagner [1974]), by including the case of nonfinite transition systems (Arnold

[1983]) and alternating finite automata (Miyano, Hayashi [1984]), and by comparison with the quantifier complexity of the logical formulas defining ω-languages (Takahashi, Yamasaki [1983], Yamasaki, Takahashi, Kobayashi [1986]). A very refined system of structural invariants for regular ω-languages is given in Wagner [1979]. It allows, for example, to estimate the length of the boolean expressions arising in McNaughton's Theorem: Consider representations of a regular set $L \subseteq A^{\omega}$ by boolean combinations of G_{δ}-sets:

$$\bigcup_{i=1}^{n} (\lim W_i \cap \sim\lim W_i')$$

and define the Rabin index of L to be the smallest n such that L is obtained in this form with regular W_i, W_i'. Wagner [1979] (and independently Kaminski [1985]) show that this index induces a strict hierarchy; moreover the Rabin index of a regular ω-language is effectively computable. More general acceptance conditions, allowing an unbounded number of quantifier alternations and leading beyond the class of regular ω-languages, are studied in Wisniewski [1987].

We can only briefly mention here related work on topological aspects of functions or relations over A^{ω} (or over the extended space A^{∞}): Boasson, Nivat [1980] and Staiger [1987] investigate sequential mappings, in particular the topological and language theoretical properties of ω-languages they preserve, and Gire, Nivat [1984] introduce a generalization of the notion of rational transduction to ω-languages.

Recently, the topological approach has also been applied in the classification of sequence properties that arise in distributed systems. Intuitively one calls a property of state sequences of a system a safety property if it ensures that nothing "bad" (like deadlock) happens at any time instance. Similarly, a liveness property guarantees that given any time instance something "good" (like entering a critical section) will eventually happen. For a systematic treatment of such sequence properties (leading to specific verification strategies), several exact definitions for describing these intuitive notions have been proposed, e.g. by Lichtenstein, Pnueli, Zuck [1985], Sistla [1985], Alpern, Schneider [1985, 1987]. These proposals agree in identifying safety properties with closed sets (in the Cantor topology). Liveness properties, however, are defined in different ways, for example as (boolean combinations of) G_{δ}-sets by Lichtenstein, Pnueli, Zuck [1985], or as dense sets in Alpern, Schneider [1985]. (A set $L \subseteq A^{\omega}$ is dense iff every $w \in A^*$ is extendible to a sequence $w\alpha \in L$.) In the latter case, a simple topological fact ("every set of the space is the intersection of a closed set with a dense set") can be applied to obtain a decomposition of correctness proofs into two parts, one establishing a safety property and the other showing a liveness property (Alpern, Schneider [1987]).

An acceptance mode which transcends the previous variants is motivated by so-called fairness conditions. These conditions require that actions (or transitions), which could be carried out again and again, happen in fact again and again. In this way it is possible to specify the absence of effects like "starvation" (where a process is "enabled" infinitely often but does not continue). A corresponding language theoretical operation ("fair merge") has been considered by Park [1980,1981]; "fair" acceptance conditions for Büchi automata appear in Priese, Rehrmann, Willecke-Klemme [1987]. For example, a Büchi automaton $\mathcal{A}$ accepts a sequence α by "path fairness" iff there is a run $s_0 s_1 \ldots$ of $\mathcal{A}$ on α which contains any finite sequence of states infinitely often which could have been started infinitely often in states of the run (on the respective rest-input). With this kind of acceptance nonregular ω-languages are recognizable; on the other hand, there are simple regular ω-languages that do not fall in this class.- The study of fairness assumptions remains an important topic of current research. For a general exposition see the monograph Francez [1987]; as a recent contribution which characterizes regular ω-languages by fairness conditions in the calculus SCCS (synchronous CCS) we mention Guessarian, Niar [1988].

6. Star-free ω-languages

An interesting class of regular ω-languages is obtained when the monadic second-order formalism S1S is restricted to first-order logic; in this case quantification is allowed only over elements (i.e., positions in sequences). The first-order definable ω-languages are closely related to the class of star-free languages and to the propositional temporal logic of linear time. In this section we discuss both aspects.

Recall that a language $W \subseteq A^*$ is star-free if it can be generated from finite sets of words by repeated application of the boolean operations and concatenation. The resulting star-free expressions are very similar to first-order formulas, by the close correspondence between $\sim$, $\cup$, $\cdot$ and the connectives $\neg$, $\vee$, $\exists$. For example, the star-free language $b^* \cdot a \cdot A^*$ (with $A = \{a,b,c\}$, $b^* = \sim(A^* \cdot (a \cup c) \cdot A^*)$) is defined by the first-order sentence

$$\exists x (x \in Q_a \wedge \neg \exists y (y < x \wedge (y \in Q_a \vee y \in Q_c))).$$

An easy induction shows that each star-free language is first-order definable. In particular, if U, V are defined by the first-order formulas φ, ψ, then $U \cdot V$ is defined by $\exists x(\varphi'(x) \wedge \psi'(x))$ where $\varphi'(x)$ and $\psi'(x)$ are the relativizations of φ, ψ to the elements $\le x$ and $> x$, respectively (assuming here for simplicity that $\varepsilon \notin U$). More difficult is the converse translation, which yields

6.1. Theorem (McNaughton, Papert [1971])

A language $W \subseteq A^*$ is star-free iff it is first-order definable (in the signature with $<$ and the unary predicates Q_a for $a \in A$).

Proof. For the translation from first-order logic into star-free expressions we use induction over quantifier depth of formulas (i.e., the maximum number of nested quantifiers). In the induction step, we consider the case of the existential quantifier, here for a formula $\exists x\varphi(x)$ of quantifier depth n+1, assuming that sentences of quantifier depth n define star-free sets. (The general situation $\exists x\varphi(x,\bar{y})$, with free variables $\bar{y}$ as parameters, is a little more technical.) We shall reduce the statement $\exists x\varphi(x)$ to statements of the form $\exists x(\varphi_{<x} \wedge x \in Q_a \wedge \varphi_{>x})$ where $\varphi_{<x}$, $\varphi_{>x}$ speak only about the elements $< x$, resp. $> x$. If $\varphi_{<x}$ and $\varphi_{>x}$ are of quantifier depth n, then such a formula describes a language $U{\cdot}a{\cdot}V$ where U, V are star-free by induction hypothesis.

As a preparation we introduce an equivalence relation $\equiv_n$ over A^*: Define for $u, v \in A^*$

$u \equiv_n v$ iff u and v satisfy the same sentences of quantifier-depth n.

Also an extended version of this definition is needed for formulas with free variables, in our case for formulas $\varphi(x)$ with one free variable x. The corresponding models are words with some distinguished position, of the form (u,r) where $1 \le r \le |u|$. We write $(u,r) \equiv_{n,1} (v,s)$ if (u,r) and (v,s) satisfy the same formulas $\varphi(x)$ of quantifier-depth n. An induction over quantifier depth shows the basic fact that there are, for any $n \geq 1$, only finitely many equivalence classes of $\equiv_n$ and $\equiv_{n,1}$, and that any such class W of words u, resp. class $\underline{W}$ of word models (u,r), can be defined by a sentence φ_W, resp. formula $\varphi_{\underline{W}}(x)$, of quantifier depth n. It follows that

(+) any formula $\varphi(x)$ of quantifier depth n is equivalent to a finite disjunction of formulas $\varphi_{\underline{W}}(x)$ (namely those $\varphi_{\underline{W}}(x)$ where $\underline{W}$ contains some (u,r) satisfying $\varphi(x)$).

Next we note a fact which (unlike (+)) depends on the present choice of signature with ordering and unary predicates only: The relations $\equiv_n$ and $\equiv_{n,1}$ are congruences. Namely,

(++) If $u \equiv_n v$ and $u' \equiv_n v'$, then $uu' \equiv_n vv'$,

$(++)_1$ If $u \equiv_n v$, $a \in A$, and $u' \equiv_n v'$, then $(uav', |u|+1) \equiv_{n,1} (u'av', |u'|+1)$.

(A convenient way of showing (++) and $(++)_1$ uses a characterization of $\equiv_n$ and $\equiv_{n,1}$ by the Ehrenfeucht-Fraissé game. For more details Rosenstein [1982] is recommended.) It turns out that (++) and $(++)_1$ do not depend on the fact that u,v are finite; indeed these statements hold for any linear orderings expanded by unary predicates.

We now can find the desired star-free expression for $\exists x\varphi(x)$: By (+) it suffices to treat the case of a formula $\exists x\varphi_{\underline{W}}(x)$ (since $\exists x\varphi(x) \leftrightarrow \exists x \bigvee_{\underline{W}} \varphi_{\underline{W}}(x) \leftrightarrow \bigvee_{\underline{W}} \exists x\varphi_{\underline{W}}(x)$). Consider now a triple (U,a,V) where U,V are $\equiv_n$-classes and $a \in A$. If there are $u_o \in U$, $v_o \in V$ such that $(u_oav_o, |u_o|+1) \in \underline{W}$, then, by $(++)_1$, we have for all words uav with $u \in U$, $v \in V$ that $(uav, |u|+1) \in \underline{W}$. Hence all words from $U{\cdot}a{\cdot}V$ satisfy $\exists x\varphi_{\underline{W}}(x)$. Thus $\exists x\varphi_{\underline{W}}(x)$ defines the union of the sets $U{\cdot}a{\cdot}V$ taken over all triples (U,a,V) where U, V are $\equiv_n$-classes and contain u_o, v_o as above. Since U,V are star-free by induction hypothesis, $\exists x\varphi_{\underline{W}}(x)$ defines a star-free set. ❑

The correspondence between star-free expressions and first-order formulas is even tighter than expressed above: The classification of star-free languages by dot-depth (= number of alternations between concatenation and boolean operations) coincides with the classification of first-order definable languages in terms of quantifier alternation depth (cf. Thomas [1982], Takahashi [1986], Perrin, Pin [1986]).

The proof of 6.1 provides the main prerequisites which are needed for a development of a theory of star-free (or first-order) ω-languages in close analogy to the regular case. It suffices essentially to repeat the proofs in sections 2 and 4 (in particular, 2.2, 4.5, and 4.6), replacing the congruences $\sim_{\mathcal{A}}$ by the congruences $\equiv_n$ and using the fact that $\equiv_n$-classes are star-free. In this way one obtains

6.2. Theorem (Ladner [1977], Thomas [1979,1981])

For an ω-language $L \subseteq A^\omega$, the following conditions are equivalent:

(a) L is first-order definable (in signature $<$, Q_a for $a \in A$),

(b) L is a finite union of sets $U{\cdot}V^\omega$ where $U,V \subseteq A^*$ are star-free and $V{\cdot}V \subseteq V$,

(c) L is a finite union of sets $\lim U \cap \sim \lim V$, where $U,V \subseteq A^*$ are star-free,

(d) L is obtained from A^ω by repeated application of boolean operations and concatenation with star-free sets $W \subseteq A^*$ on the left. ❑

An ω-language satisfying one of the conditions above is called star-free. The notion was proposed by Ladner [1977], who referred to condition (d) and showed that the star-free ω-languages form a proper subclass of the regular ones. A short and self-contained proof of 6.1 and 6.2(a) ↔ (d) is given in Perrin, Pin [1986]. Perrin [1985] shows that the equivalence between (b) and (c) remains true for any class of regular languages which is associated with a variety of semigroups which is closed under Schützenberger product.

Further interesting aspects of the star-free ω-languages are revealed when one considers their syntactic monoid as introduced in Theorem 2.6 above. Referring to this monoid it is possible to extend Schützenberger's Theorem from languages to ω-languages. Recall that this theorem states that a regular language $W \subseteq A^*$ is star-free iff its syntactic monoid is group-free. Via condition (b) of 6.2, this implies

6.3. Theorem (Perrin [1984])
A regular ω-language $L \subseteq A^\omega$ is star-free iff its syntactic monoid $A^*/{\approx_L}$ is group-free. ❑

This result yields an effective test deciding whether a given regular ω-language is star-free. Also one can use this characterization to exhibit regular ω-languages that are not star-free. For this purpose one observes that a nontrivial group exists in $A^*/{\approx_L}$ iff there are words $u,x,y,z \in A^*$ such that for infinitely many n, $xu^n yz^\omega \in L$ and for infinitely many n, $xu^n yz^\omega \notin L$ (or analogously for $x(yu^n z)^\omega$). So the example language L_2 of section 1 ("between any two a there is an even number of b,c") is not star-free, as can be seen by taking x,y,z = a and u = b.

As in the theory of regular languages of finite words, the group-free monoids are just a first example of a variety of semigroups that characterizes an interesting language class. Further cases in the domain of ω-languages have been studied by Pécuchet [1986a,1987].

Much of the recent interest in the star-free ω-languages rests on their connection with <u>propositional temporal logic</u> of linear time, short <u>PTL</u>. PTL-formulas are built up from atomic propositions $p_1,p_2,\ldots$ by means of the boolean connectives, the unary temporal operators O ("next"), ◊ ("eventually"), [] ("always", "henceforth"), and the binary operator U ("until"). A PTL-formula with atomic propositions $p_1,\ldots,p_n$ is interpreted in ω-sequences over $\{0,1\}^n$, where by definition a sequence α satisfies p_i (short: $\alpha \models p_i$) iff α(0) has 1 in its i-th component. The semantics of O, ◊, [], U is defined by

$\alpha \models O\varphi$	iff	$\alpha(1,\omega) \models \varphi$	("φ holds next time")
$\alpha \models \Diamond\varphi$	iff	there is i≥0 s.t. $\alpha(i,\omega) \models \varphi$	("φ holds eventually")
$\alpha \models \Box\varphi$	iff	for all i≥0, $\alpha(i,\omega) \models \varphi$	("φ holds henceforth")
$\alpha \models \varphi U \psi$	iff	there is i≥0 s.t.: $\alpha(i,\omega) \models \psi$ and $\alpha(j,\omega) \models \varphi$ for 0≤j<i	("φ holds until eventually ψ holds").

It is straightforward to translate PTL-formulas (over atomic propositions $p_1,\ldots,p_n$) into formulas of first-order logic (with unary predicate symbols $X_1,\ldots,X_n$ and using the interpretation described in section 3). For example, the PTL-formula

$$\Box(p_1 \rightarrow O\Diamond((\neg p_2)\, U\, p_1))$$

is equivalent (over ω-sequences $\alpha \in (\{0,1\}^2)^\omega$) to the first-order formula

$$\forall x(x \in X_1 \rightarrow \exists y(x < y \wedge y \in X_1 \wedge \forall z(x < z \wedge z < y \rightarrow \neg\, z \in X_2)))$$

Thus PTL may be considered as a system of first-order logic with only implicit use of variables. As seen above, quantification refers to segments that are unbounded to the right, with the only exception of the bounded quantification involved in the until-operator. Hence it tends to be hard to express in PTL statements about finite segments of sequences. Nevertheless, we have

6.4. Theorem (Kamp [1968], Gabbay, Pnueli, Shelah, Stavi [1980])
Propositional temporal logic PTL (with atomic propositions $p_1,\ldots,p_n$) is expressively equivalent to first-order logic over ω-sequences (in the signature with < and n unary predicates). ❑

The translation from first-order logic to PTL is difficult and will not be described here. It involves a non-elementary blow-up in the length of the formulas, as can be seen from the fact that satisfiability of PTL-formulas is PSPACE-complete (Sistla, Clarke [1985]) while satisfiability is nonelementary for star-free expressions or for first-order formulas in the given signature (Stockmeyer, Meyer [1973]). A transparent proof of 6.4 which is based on 6.3 and the wreath product decomposition of group-free monoids has been found by Perrin, Pin [1986a].

An extension of PTL which allows to define exactly the regular sets of ω-sequences was suggested by Wolper [1983], Wolper, Vardi, Sistla [1983] (see also Vardi, Wolper [1988]); it is called ETL ("extended temporal logic"). The idea is to admit an infinity of temporal operators, each of them associated with a regular grammar (or an automaton). The standard operators of PTL are included in this set-up as simple examples.

Temporal logic has attracted attention as a framework for the specification and verification of concurrent programs (Pnueli [1981], Lamport [1983]). Temporal logic formulas are well suited for this purpose since arguing about concurrent programs is primarily concerned with their ongoing behavior in time and not so much with their input-output behavior (to which the formulas of Hoare logic correspond more directly). From the extensive literature on the subject we select a few aspects connected with Büchi automata and ω-language theory. (The reader will find more on the topic in the surveys Pnueli [1986] and Emerson [1988].)

We refer to the model of computation proposed in Manna, Pnueli [1981]. A <u>concurrent program</u> P consists of say n modules $P_1,\ldots,P_n$ where each of the P_i is a sequential program composed of labelled instructions; both shared variables and variables private to the P_i are admitted. The possible computations of P are defined as the interleavings of the computations of the P_i. A state of the program is identified with an n-tuple of instruction labels from $P_1,...,P_n$ and of values for the program variables. Since the desired properties of such a program (like deadlock freedom, fairness etc.) are typically concerned with the flow of control and not so much with an infinity of possible values for the variables, it is often appropriate to assume that the number of essentially different states is finite. In this case one speaks of a finite state program.

Suppose that for a specification of the program P the properties $p_1,...,p_m$ of states are relevant. Then the program can be represented as an annotated directed graph, where nodes are states and arrows represent transitions between states in one step. The state s is annotated by those p_i which are true in s. Formally, the graph is a Kripke structure $\mathcal{M}_P = (S,R,\Phi)$ where S is the set of states, $R \subseteq S \times S$ the transition relation, and $\Phi: S \rightarrow 2^{\{p_1,...,p_m\}}$ a truth valuation. Note that each $\Phi(s)$ can be regarded as an m-bit vector from $\{0,1\}^m$; thus any computation $\sigma = s_0 s_1 ... \in S^\omega$ induces a corresponding sequence $\alpha = \Phi(s_0)\Phi(s_1)...$ from $(\{0,1\}^m)^\omega$, containing the relevant information about σ w.r.t. the properties $p_1,...,p_m$.

In this framework, the correctness problem (whether program P is in accordance with specification φ) is the following question: Do all sequences $\alpha \in (\{0,1\}^m)^\omega$ which are given by paths through $\mathcal{M}_P$ satisfy the PTL-formula φ? One says in this case that "$\mathcal{M}_P$ is a model of φ"; so the correctness problem has been rephrased as a modelchecking problem.

Several approaches have been studied to develop efficient model checking procedures, for instance using tableaux (= extensions of the annotations of $\mathcal{M}_P$ by arbitrary subformulas of the specification), cf. Lichtenstein, Pnueli [1985]. Vardi, Wolper [1986a] suggest to apply the theory of Büchi automata for the programs and for the specifications: First it is possible to view the Kripke structure $\mathcal{M}_P$ as a Büchi automaton $\mathcal{A}_P$ such that $L(\mathcal{A}_P)$ contains the sequences $\alpha \in (\{0,1\}^m)^\omega$ given by $\mathcal{M}_P$. Secondly a PTL-formula φ defines a regular ω-language (by 6.2, 6.3) and thus is also representable by a Büchi automaton $\mathcal{A}_\varphi$. Hence the correctness problem reduces to the containment problem for ω-languages "$L(\mathcal{A}_P) \subseteq L(\mathcal{A}_\varphi)$?" or, in negated form, to the intersection problem "$L(\mathcal{A}_P) \cap L(\mathcal{A}_{\neg\varphi}) \neq \emptyset$?". Alpern, Schneider [1985a,1987] and Manna, Pnueli [1987] suggest to use Büchi automata as a genuine specification formalism, taking advantage of the fact that a pictoral graph representation can be more transparent than logical formulas. In Manna, Pnueli [1987] a variant of the Büchi acceptance condition is used (the "$\forall$-automaton"), which involves a "universal" condition on runs instead of an "existential" one and is hence better suited for the question whether all $\mathcal{A}_P$-runs satisfy the specification.

Further applications of ω-language theory are based on normal form theorems, in particular the representation of regular ω-languages as unions of sets $\lim U_i \cap \sim\lim V_i$ in McNaughton's Theorem and in 6.2 (see e.g. Lichtenstein, Pnueli, Zuck [1985], Pnueli [1986], Alpern Schneider [1987a]).

7. Context-free ω-languages

In this section we give a short account on the use of grammars for the generation of ω-words. A natural approach is to allow infinite leftmost derivations. As an example, consider the following grammar:

G_0: $x_1 \rightarrow x_2x_1$, $x_2 \rightarrow ax_2b \mid ab$.

The infinite derivation

(1) $x_1 \vdash x_2x_1 \vdash abx_1 \vdash abx_2x_1 \vdash ababx_1 \vdash ababx_2x_1 \vdash ...$

generates the ω-word $(ab)^\omega$ from left to right, and

(2) $x_1 \vdash x_2x_1 \vdash abx_1 \vdash abx_2x_1 \vdash abax_2bx_1 \vdash abaax_2bbx_1 \vdash abaaax_2bbbx_1 \vdash ...$

yields from left to right the ω-word aba^ω.

Derivation (2) will be excluded if we impose the condition that both variables x_1,x_2 be used infinitely often (as left-hand side of applied rules). Thus there are two variants of context-free generation of ω-languages, depending on whether arbitrary (leftmost) derivations are admitted or the variables used infinitely often are also specified.

In the sequel, a context-free grammar G over the alphabet A with variables (nonterminals) $x_1,...,x_n$ is given by an n-tuple $(G_1,...,G_n)$ of finite sets $G_i \subseteq (A \cup \{x_1,...,x_n\})^*$, where $w \in G_i$ means that $x_i \rightarrow w$ is a rule of G. As start symbol the variable x_1 is used. A leftmost derivation

$$y_0 \vdash u_1y_1v_1 \vdash u_2y_2v_2 \vdash ...$$

with $y_0 = x_1$, $u_i \in A^*$, and $y_i \in \{x_1,...,x_n\}$ generates either an ω-word (the unique common extension of the u_i) or a finite word (if for some n, $u_n = u_{n+1} = ...$). Of course, a finite word can also be generated by a terminating derivation. Hence an appropriate domain for the present discussion is A^∞ instead of A^ω. An ∞-language $L \subseteq A^\infty$ is called algebraic (or: ∞-algebraic) if there is a context-free grammar G such that L consists of all words of A^∞ that are generated from x_1 by a leftmost derivation of G. If G is given together with a system $\mathcal{F}$ of sets of variables, such that only those derivations are admitted where the variables used infinitely often form a set in $\mathcal{F}$, then the resulting ∞-language is called context-free (or: ∞-context-free). Analogous definitions apply to ω-languages. Then it can be shown that a ∞-language is algebraic (resp. context-free) iff it is the union of an algebraic (resp. context-free) ω-language with a context-free language of finite words. In this way also the notion of a regular ∞-language is introduced. For the algebraic case the restricion to leftmost derivations is inessential (Nivat [1978]), while for context-free ∞-languages cancellation of this property causes a proper loss of generating power (Cohen, Gold [1977]).

Let us first note that the regular, algebraic and context-free ω-languages (and hence ∞-languages) form a proper hierarchy:

7.1. Theorem (Cohen, Gold [1977])

The class of regular ω-languages is properly contained in the class of algebraic ω-languages which itself is properly contained in the class of context-free ω-languages.

Proof. It is obvious that any algebraic ω-language is context-free. To show that regular ω-languages are algebraic, we refer to the representation of regular ω-languages in the form $L = \bigcup_{i=1}^{n} U_i \cdot V_i^{\omega}$ where U_i, V_i are regular. The proof will be clear when the case $L = V^{\omega}$ (assuming $\varepsilon \notin V$) is settled. Let G_V be a left-linear grammar which generates the regular set V (say with start symbol y_1). Then G_V extended by the rule $x_1 \to y_1 x_1$ generates ω-words by leftmost derivations of the form

$$x_1 \vdash y_1 x_1 \vdash^* v_1 x_1 \vdash v_1 y_1 x_1 \vdash^* v_1 v_2 x_1 \vdash \ldots$$

and hence defines the ω-language V^{ω}. (Note that if G_V were right-linear one would obtain the ω-words in $V^* \cdot (\lim(\text{pref } V))$ which in general is different from V^{ω}.)

We indicate properness of the inclusions only by definiton of suitable example languages: The algebraic ω-language

$$\{a^{n_1}b^{n_1}a^{n_2}b^{n_2}\ldots \mid n_i \geq 1\} \cup \{a^{n_1}b^{n_1}\ldots a^{n_r}b^{n_r}a^{\omega} \mid n_1,\ldots,n_r \geq 1\}$$

is generated by the example grammar G_0 above and easily shown to be non-regular (by an adaptation of the pumping lemma for nondeterministic finite automata). On the other hand, the first part of the above union is context-free since it is generated by G_0 with the system $\mathcal{F} = \{\{x_1,x_2\}\}$ of one designated variable-set (see the above derivation examples (1), (2)). It turns out that this ω-language is indeed not algebraic. ❑

The above proof for regular ω-languages can be extended to a characterization of the

context-free ω-languages. If $\mathcal{L}$ is a class of languages ($\subseteq A^*$), the ω-Kleene closure of $\mathcal{L}$ is the class of all ω-languages which are finite unions of sets $U \cdot V^{\omega}$ with $U,V \in \mathcal{L}$.

7.2. Theorem (Cohen,Gold [1977])

An ω-language is context-free iff it belongs to the ω-Kleene closure of the class of context-free languages. ❑

Some standard results on context-free grammars fail when considered in the domain of ∞-languages. We discuss an interesting example of this kind, the reduction to Greibach normal form (i.e. to grammars $G = (G_1,\ldots,G_n)$ where each G_i is contained in $A \cdot (A \cup \{x_1,\ldots x_n\})^*$). Call an ∞-language Greibach-algebraic if it is given as in the definition of "algebraic" but referring to grammars in Greibach form. We shall show that the Greibach algebraic ∞-languages do not cover the class of regular ∞-languages. For the proof one extends the topology over A^{ω} (as introduced in section 5) to a topology over A^{∞}, by introduction of a new symbol $\Omega \notin A$ and representation of finite words w as sequences $w \cdot \Omega^{\omega}$. We shall verify that Greibach algebraic ∞-languages are closed in this topology. Since there are nonclosed regular ∞-languages (for example, lim a*bb*, as shown before 5.1), the Greibach algebraic ∞-languages form a proper subclass of the algebraic ones.

The topological closure cl(L) of a set $L \subseteq A^{\infty}$ contains the sequences that have arbitrary long common prefixes with sequences from L. In other words, the set

$$\text{adh}(L) := \lim(\text{pref}(L))$$

the adherence of L, is the set of accumulation points of L, and we have $\text{cl}(L) = L \cup \text{adh}(L)$.

In order to show that Greibach algebraic ∞-languages are closed, let $\alpha \in A^{\omega}$ be an accumulation point of the Greibach algebraic set L, i.e. for infinitely many prefixes u of α there is $\beta \in A^{\infty}$ with $u\beta \in L$. We have to verify $\alpha \in L$. Consider all leftmost derivations (by the Greibach grammar G for L) which generate a sequence $u\beta \in A^{\infty}$ where u is prefix of α. We organize the finite initial parts of such derivations in tree form. We obtain a finitely branching tree of finite derivations which has the zero-step derivation x_1 as root and is infinite by assumption on α. By König's Lemma there is an infinite path, i.e. an infinite derivation. This derivation must describe a generation of α (and not only of a finite prefix of α) because G is Greibach.

Boasson, Nivat [1980] proved a sharpened form of this result, including also a converse statement:

7.3. Theorem (Boasson, Nivat [1980])

A context-free ∞-language $L \subseteq A^{\infty}$ is Greibach algebraic iff it is the (topological) closure of a context-free language $W \subseteq A^*$. ❑

A common framework for characterizations of the Greibach algebraic, algebraic and context-free ∞-languages has been developed by Niwinski [1984], continuing previous work of Nivat [1977,1978,1979], Park [1980], and others. Here a grammar $G = (G_1,...,G_n)$ is regarded as a 'fixed point operator. The operator maps n-tuples of ∞-languages to n-tuples of ∞-languages. The first component of its (well-defined) greatest fixed point is by definition the ∞-language defined by G. More precisely, any word w from $(A \cup \{x_1,...,x_n\})^*$ defines a map F_w that sends an n-tuple $(L_1,...,L_n)$, where $L_i \subseteq A^{\infty}$, to the ∞-language which results from w by substituting L_i for x_i. (For cases like $w = x_1x_2$, certain conventions concerning concatenation are used, such as $L_1 \cdot L_2 = L_1$ for $L_1 \subseteq A^{\omega}$, $L_2 \subseteq A^{\infty}$.) Now one associates with a set $G_i \subseteq (A \cup \{x_1,...,x_n\})^*$ the map which yields for $(L_1,...,L_n)$ the union of the sets $F_w(L_1,...,L_n)$ where $w \in G_i$.

In this way G induces an operator $\bar{G}$ mapping n-tuples to n-tuples of ∞-languages. Since $\bar{G}$ is monotone (w.r.t. set inclusion taken componentwise), an application of the Knaster-Tarski Theorem guarantees a greatest fixed point $(K_1,...,K_n)$. (The n-tuple $(K_1,...,K_n)$ is obtained from $(A^{\infty},...,A^{\infty})$ by a β-fold iteration of $\bar{G}$ for some ordinal β, possibly greater than ω.) Note that in this set-up the sets G_i need not be finite. For a class $\mathcal{L}$ of languages $L \subseteq A^*$ denote by $GFP(\mathcal{L})$ the class of ∞-languages that are obtained from an operator $\bar{G}$ as first component of its greatest fixed point, where the components G_i of G are in $\mathcal{L}$. For $\mathcal{L}$ = FIN (the finite languages), resp. REG (the regular languages), and CF (the context-free languages) we arrive at the mentioned characterization:

7.4. Theorem (Niwinski [1984])

For any ∞-language L :

(a) L is Greibach algebraic iff $L \in GFP(FIN)$,

(b) L is algebraic iff $L \in GFP(REG)$,

(c) L is context-free iff $L \in GFP(CF)$. ❑

A much more complicated classification of context-free ω- (or ∞-) languages arises when we consider definitions by deterministic or nondeterministic pushdown automata with various modes of acceptance. Cohen, Gold [1977] characterize the algebraic ω-languages by pushdown automata with 1'-acceptance (meaning that a run exists such

that all its states are in one of the designated state sets), and the context-free ω-languages by pushdown automata with 3-acceptance (analogous to Muller acceptance for nondeterministic automata). A detailed analysis of ∞-language classes induced by deterministic pushdown automata is given in Linna [1976,1977] and Cohen, Gold [1978].

In the above treatment of grammars a certain asymmetry is manifested in the convention that only left to right generation of ω-words is considered and terminal symbols are ignored when not reached from the left within ω steps. Dropping this restriction, the derivation example (2) of the beginning of this section would be regarded as producing the generalized word abaaa... ...bbb. Formally, generalized words over the alphabet A are identified with A-labelled countable orderings (where the ordering $(\omega,<)$ occurs as a special case). In Courcelle [1978], Heilbrunner [1980], Thomas [1986], Dauchet, Timmerman [1986] some results on generalized words, their derivation trees, and finite expressions for their description are presented.

There are further devices of computation that have been investigated in connection with ω-languages but cannot be treated in detail here. We mention two such models: Turing machines (studied e.g. in Cohen, Gold [1980], Lindsay [1986], see also Staiger [1986]), and Petri nets. From a Petri net, a language and an ω-language can be extracted essentially by collecting all sequences of transitions that describe an admissible firing sequence. Jantzen [1986] presents a survey on Petri net languages of finite words. In Valk [1983] a language theoretical characterization of Petri net ω-languages is established close to the representation of regular ω-sets as unions of sets $U \cdot V^{\omega}$ with regular U,V. Parigot, Pelz [1985] and Pelz [1987] describe a logical formalism (extending Büchi's theory S1S) which characterizes Petri net (ω-)languages; they refer to existential formulas in a signature where a primitive for comparison of finite cardinalities has been added.

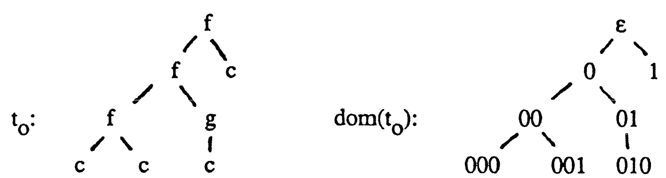
II. Automata on infinite trees

Notation

If A is an alphabet, an A-valued tree t is specified by its set of nodes (the "domain" $dom(t)$) and a valuation of the nodes in the alphabet A. Formally, a k-ary A-valued tree is a map $t : dom(t) \to A$ where $dom(t) \subseteq \{0,...,k-1\}^*$ is a nonempty set, closed under prefixes, which satisfies

$$wj \in dom(t),\ i<j \Rightarrow wi \in dom(t).$$

As an example over $A = \{f,g,c\}$ consider a finite tree:



The frontier of t is the set

$$fr(t) = \{w \in dom(t) \mid \neg \exists i\ wi \in dom(t)\},$$

and the outer frontier $fr^+(t)$ contains the points $wi \notin dom(t)$ where $w \in dom(t)$ and $i < k$.

We set $dom^+(t) = dom(t) \cup fr^+(t)$. The (proper) prefix relation over $\{0,...,k-1\}^*$ is written $<$. A path through t is a maximal subset of $dom(t)$ linearly ordered by $<$. The subtree t_w of t at node $w \in dom(t)$ is given by $dom(t_w) = \{v \in \{0,...,k-1\}^* \mid wv \in dom(t)\}$ and $t_w(v) = t(wv)$ for $v \in dom(t_w)$.

Often trees arise as terms (possibly infinite terms). In this case one refers to a ranked alphabet $A = A_0 \cup ... \cup A_k$ where A_i contains i-ary function symbols. The example tree t_0 above represents the term $f(f(f(c,c),g(c)),c)$ over the ranked alphabet $A = A_0 \cup A_1 \cup A_2$ with $A_0 = \{c\}$, $A_1 = \{g\}$, $A_2 = \{f\}$. (One may even allow $A_i \cap A_j \neq \emptyset$ for $i \neq j$.)

For easier exposition we shall restrict to binary trees in the sequel: By T_A denote the set of finite binary A-valued trees (where a node has either two sons or no sons), and by T^ω_A the set of infinite A-valued trees with domain $\{0,1\}^*$. Set $T^\infty_A = T_A \cup T^\omega_A$. Subsets T of T_A or T^ω_A will sometimes be called tree languages. Binary trees represent the typical case from which all notions and results to be discussed below are easily transferred to the general situation. For instance, arbitrary k-ary A-valued trees can be represented within binary trees from $T^\omega_{A \cup \{\Omega\}}$, where Ω is a new symbol, via the

coding which maps node $n_1...n_r$ ($n_i < k$) to node $1^{n_1}0 \ldots 1^{n_r}0$ and associates value Ω to all nodes of $\{0,1\}^*$ which are outside the range of this map. In a similar way it is possible to handle countably branching trees in the framework of binary trees.

We now introduce notation concerning tree concatenation (defined here in terms of tree substitution). Let $T,T' \subseteq T_A$ and $c \in A$. Then $T \cdot^c T'$ contains all trees which result from some $t \in T$ by replacing each occurrence of c on $fr(t)$ by a tree from T', where different trees are admitted for different occurrences of c. Define a corresponding star operation $*^c$ by

$$T^{*c} = \bigcup_{n\geq 0} T^{nc}$$

where

$$T^{0c} = \{c\}, \quad T^{(n+1)c} = T^{nc} \cup (T \cdot^c T^{nc}).$$

A tree language $T \subseteq T_A$ is called regular iff for some finite set C disjoint from A, T can be obtained from finite subsets of $T_{A \cup C}$ by applications of union, concatenations $\cdot^c$, and star operations $*^c$ where $c \in C$. Note that this notion of regularity generalizes that for sets of finite words, if a word $w = a_1 a_2 ... a_r$ over A is considered as a unary tree over $A \cup \{c\}$ of the form $a_1(a_2(...a_r(c)...))$.

We shall also refer to tuples $\mathbf{c} = (c_1,...,c_m)$ of concatenation symbols instead of single symbols c. For $T,T_1,...,T_m \subseteq T_A$ let $T \cdot^{\mathbf{c}} (T_1,...,T_m)$ be the set of trees obtained from trees $t \in T$ by substituting, for $i = 1,...,m$, each occurrence of c_i on $fr(t)$ by some tree in T_i. - Furthermore, the set $(T_1,...,T_m)^{\omega \mathbf{c}}$ is defined to consist of all infinite trees obtained by ω-fold iteration of this tree concatenation; more precisely, it contains all trees $t \in T^\omega_A$ for which there are trees $t_0,t_1,...$ such that $t_0 \in \{c_1,...,c_m\}$, $t_{m+1} \in \{t_m\} \cdot^{\mathbf{c}} (T_1,...,T_m)$, and t is the common extension of the trees t'_m which result from the t_m by deleting the symbols c_i at their frontiers.

We shall use expressions like $t_1 \cdot^c t_2$ or $t_0 \cdot^{\mathbf{c}} (t_1,...,t_m)^{\omega \mathbf{c}}$ as shorthands for $\{t_1\} \cdot^c \{t_2\}$, $\{t_0\} \cdot^{\mathbf{c}} (\{t_1\},...,\{t_m\})^{\omega \mathbf{c}}$, respectively. This notation is extended to infinite trees t_i with domain $\{0,1\}^*$ by the convention that instead of frontier occurrences of the c_i the "first" occurrences of the c_i are used for replacement, i.e. their occurrences at nodes w such that no $v < w$ exists with a value c_j. We write $t_1 \cdot t_2$ and $t_0 \cdot (t_1,...,t_m)^\omega$ if the symbols c, $\mathbf{c}$ are clear from the context.

8. Tree automata

Tree automata generalize sequential automata in a simple way: On a given A-valued tree, the automaton starts its computation at the root in an initial state and then simultaneously works down the paths of the tree level by level. The transition relation specifies which pairs (q_1,q_2) of states can be assumed at the two sons of a node, given the node's value in A and the state assumed there. The tree automaton accepts the tree if there is a run built up in this fashion which is "successful". A run is successful if all its paths are successful in a sense given by an acceptance condition for sequential automata. It turns out that for infinite trees the reference to Büchi and Muller acceptance leads to nonequivalent types of tree automata. In this section we introduce these tree automata, first studied by Rabin [1969,1970].

As a preparation we collect some basic notions and facts concerning automata over finite trees. A (nondeterministic top-down) tree automaton over A is of the form $\mathcal{A} = (Q,Q_0,\Delta,F)$, where Q is nonempty and finite, $Q_0,F \subseteq Q$ are the sets of initial, resp. final states, and $\Delta \subseteq Q{\times}A{\times}Q{\times}Q$ is the transition relation. A run of $\mathcal{A}$ on t is a tree $r:\mathrm{dom}^+(t) \to Q$ where $r(\varepsilon) \in Q_0$ and $(r(w),t(w),r(w0),r(w1)) \in \Delta$ for each $w \in \mathrm{dom}(t)$; it is successful if $r(w) \in F$ for all $w \in \mathrm{fr}^+(t)$. The tree language $T(\mathcal{A})$ recognized by $\mathcal{A}$ consists of all trees t which admit a successful run of $\mathcal{A}$ on t, and $T \subseteq T_A$ is recognizable if $T = T(\mathcal{A})$ for some tree automaton $\mathcal{A}$.

Most of the basic results on regular word languages can be reproved for recognizable tree languages, including a Kleene theorem and closure under boolean operations (cf. 8.1 below). However, an important difference between the sequential and the tree case appears in the question of determinism, since deterministic top-down tree automata, where a function $\delta:Q{\times}A \to Q{\times}Q$ replaces the transition relation, are strictly weaker than nondeterministic ones. (For instance, any deterministic top-down tree automaton accepting the trees f(a,b) and f(b,a) would have to accept f(a,a) as well; so it does not recognize the finite set {f(a,b),f(b,a)}, which is clearly recognized by a nondeterministic top-down tree automaton.) Intuitively, tree properties specified by deterministic top-down automata can depend only on path properties. A reduction to determinism is possible when the working direction of tree automata is reversed from "top-down" to "bottom-up". Nondeterministic bottom-up tree automata are of the form (Q,Q_0,Δ,F) with $\Delta \subseteq Q{\times}Q{\times}A{\times}Q$ and Q_0, F as before. A successful run on a tree t should have a state from Q_0 at each point of $\mathrm{fr}^+(t)$ and a state from F at the root ε. By an obvious correspondence, nondeterministic bottom-up tree automata are equivalent to nondeterministic top-down tree automata. However, for the bottom-up version it is possible to carry out the "subset construction" (as for usual finite automata) to obtain equivalent deterministic bottom-up tree automata. Note that the computation of such an automaton, say with state set Q, on a term t as input tree may be viewed as a parallel inside-out evaluation of t in the finite domain Q. For a detailed treatment see Gecseg, Steinby [1984]. In the sequel we refer to the nondeterministic top-down version.

Let us summarize the properties of recognizable tree languages that are needed in the sequel:

8.1. Theorem (Thatcher, Wright [1968], Doner [1970])

(a) The emptiness problem for tree automata over finite trees is decidable (in polynomial time).

(b) A tree language $T \subseteq T_A$ is recognizable iff T is regular.

(c) The class of recognizable tree languages $T \subseteq T_A$ is closed under boolean operations and projection.

Proof. For (a) the decidability claim is clear from the fact that a tree automaton $\mathcal{A}$, say with n states, accepts some tree iff $\mathcal{A}$ accepts one of the finitely many trees of height $\leq n$ (eliminate state repetitions on paths). A polynomial algorithm results from the observation that for the decision it even suffices to build up partial run trees in which each transition of the automaton is used at most once.

Part (b) is shown in close analogy to the proof of Kleene's Theorem for sets of finite words. For details see Gecseg, Steinby [1984].

In (c), the step concerning union is straightforward. Closure under complement is shown using the equivalence between nondeterministic top-down and deterministic bottom-up tree automata: For the latter, complementation simply means to change the nonfinal states into final ones and vice versa. For projection, assume $T \subseteq T_{A\times B}$ is recognizable and consider its projection to the A-component, i.e. the set

$$T' = \{s \in T_A \mid \exists t \in T_B\ s\hat{\ }t \in T\}$$

where $s\hat{\ }t$ is given by $s\hat{\ }t(w) = (s(w),t(w))$. If T is recognized by the tree automaton $\mathcal{A}$, then T' is recognized by an automaton which guesses on a tree $s \in T_A$ the B-component and works on the resulting tree like $\mathcal{A}$. ❑

For nondeterministic automata it is possible to restrict the sets of initial, resp. final states to singletons. For technical reasons we shall henceforth assume that tree automata have a single initial state, which moreover occurs only at the root of run trees (i.e., it does not appear in the third and fourth component of transitions).

We now turn to tree automata over infinite trees. We consider two basic types, the Büchi tree automaton and the Rabin tree automaton, which inherit their acceptance modes from sequential Büchi automata, resp. sequential Rabin automata. A Büchi tree automaton over the alphabet A is of the form $\mathcal{A} = (Q,q_o,\Delta,F)$ with finite state set Q, $q_o \in Q$, $\Delta \subseteq Q\times A\times Q\times Q$, and $F \subseteq Q$. A Rabin tree automaton has the form (Q,q_o,Δ,Ω), where Q, q_o, Δ are as before and $\Omega = \{(L_1,U_1),\ldots,(L_n,U_n)\}$ is a collection of "accepting pairs" of state sets L_i, $U_i \subseteq Q$. Runs of these tree automata over trees from T_A^ω are mappings $r:\{0,1\}^* \to Q$, defined in the same way as over finite trees. A tree $t \in T_A^\omega$ is accepted if there is a successful run of the automaton on t. For a Büchi automaton $\mathcal{A}$ as given above a run r is successful if

$$\text{for all paths } \pi,\ In(r|\pi) \cap F \neq \emptyset,$$

where $r|\pi$ denotes the restriction of r to the path π. The reference to Büchi acceptance on paths motivates the name "Büchi tree automaton" (cf. Vardi, Wolper [1986]); Rabin [1970] used the term "special automaton".

For a Rabin tree automaton $\mathcal{A}$ of the form above a run is successful if

$$\text{for all paths } \pi \text{ there exists } i \in \{1,\ldots,n\} \text{ with } In(r|\pi) \cap L_i = \emptyset \text{ and } In(r|\pi) \cap U_i \neq \emptyset.$$

A set $T \subseteq T_A^\omega$ is Büchi recognizable, resp. Rabin recognizable, if it consists of the trees accepted by a Büchi, resp. Rabin tree automaton. Since any Büchi tree automaton may be regarded as a Rabin tree automaton (set $\Omega = \{(\emptyset,F)\}$), any Büchi recognizable set of infinite trees is Rabin recognizable.

We mention two equivalent variants of Rabin tree automata. In the first, the Muller tree automaton, the collection Ω is replaced by a system $\mathcal{F}$ of state sets, and acceptance is defined via existence of a run r such that for each path π, $In(r|\pi) \in \mathcal{F}$. The second variant, the Streett automaton as introduced by Streett [1982], is specified as a Rabin tree automaton but uses the negation of the Rabin condition on a given path of a run: A run r of a Streett automaton (Q,q_o,Δ,Ω) is successful if

$$\text{for all paths } \pi \text{ and for all } i \in \{1,\ldots,n\}\text{: } In(r|\pi) \cap U_i \neq \emptyset \text{ implies } In(r|\pi) \cap L_i \neq \emptyset.$$

To illustrate the function of Büchi and Rabin tree automata consider an example tree language T_o over $A = \{a,b\}$:

$$T_o = \{t \in T_A^\omega \mid \text{some path through t carries infinitely many a}\}.$$

A Büchi tree automaton $\mathcal{A}$ which recognizes T_o may work as follows: By nondeterministic choice, $\mathcal{A}$ guesses a path down the tree and on this path assumes a final state iff letter a is met; on the other parts of the tree only a fixed final state is computed.

Then the existence of a successful run amounts to existence of a path in t with infinitely many values a. Thus T_o is Büchi recognizable (and of course Rabin recognizable). The complement language

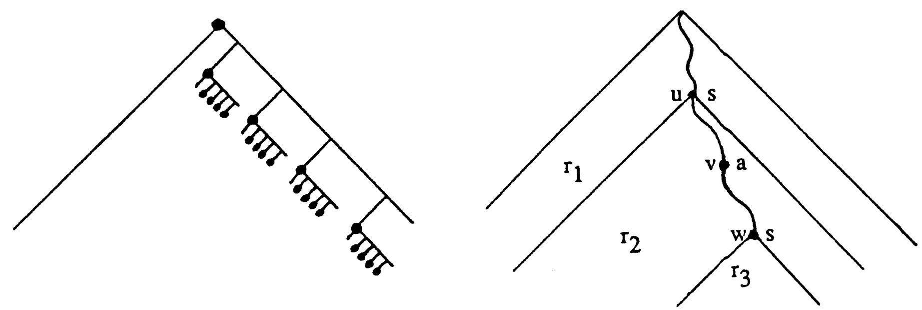
$$T_1 = \{t \in T_A^\omega \mid \text{all paths through t carry only finitely many a}\}$$

is recognized by a Rabin automaton with one accepting pair $(\{q_a\},Q)$ where Q is the state set and q_a is computed iff letter a is encountered. Büchi tree automata, however, do not offer in their acceptance condition such a "finiteness test" along paths. Indeed, they cannot recognize T_1:

8.2. Theorem (Rabin [1970])

The set T_1 is a tree language which is Rabin recognizable but not Büchi recognizable.

Proof. Assume for contradiction that T_1 is recognized by the Büchi tree automaton $\mathcal{A} = (Q,q_o,\Delta,F)$, say with n-1 states. $\mathcal{A}$ accepts all trees t_i which have letter a at the positions $\varepsilon, 1^{m1}0, \ldots, 1^{m1}0\ldots1^{mi}0$ where $m_1,\ldots,m_i > 0$, and letter b elsewhere. The figure on the left indicates the nodes with value a in t_2.



Consider a successful run r of $\mathcal{A}$ on t_n. One shows (by induction on n) that there must be a path in t_n with three nodes $u < v < w$ such that $r(u) = r(w) = s \in F$ and $t_n(v) = a$. Nodes u and w induce a decomposition of r (and t_n) into three parts as follows: Obtain r_1 from r by deleting r_u and setting $r_1(u) = c$. Obtain r_2 similarly from r_u by deleting r_w, and let $r_3 = r_w$. Then $r = r_1 \cdot r_2 \cdot r_3$. In the analogous way a decomposition of the underlying tree t_n in the form $t_n = s_1 \cdot s_2 \cdot s_3$ is defined. Now consider $r_1 \cdot r_2^\omega$. It is a successful run of $\mathcal{A}$ on the tree $s_1 \cdot s_2^\omega$. Since by choice of u,v,w this tree contains a path with infinitely many letters a, we obtain a contradiction. ❑

The following sections present results analogous to Theorem 8.1 for Büchi and Rabin tree automata, excepting only a Kleene type characterization for Rabin recognizable sets and complementation for Büchi tree automata. For Rabin tree automata, however, closure under boolean operations and projection holds, which leads to an equivalence with monadic second-order logic as in 3.1.

9. Emptiness problem and regular trees

In this section the structure of successful runs of Büchi and Rabin tree automata is analyzed.

First we consider Büchi tree automata and show a simple representation of Büchi recognizable sets in terms of recognizable sets of finite trees.

Let $\mathcal{A} = (Q,q_o,\Delta,F)$ be a Büchi tree automaton, where $F = \{q_1,\dots,q_m\}$, and let $r:\{0,1\}^* \to Q$ be a successful run of $\mathcal{A}$ on the tree $t \in T^{\omega}_{A}$. We claim that r can be built up from finite run trees of $\mathcal{A}$ which are delimited by final states of $\mathcal{A}$. Indeed, the following observation shows that starting from any node u of r the next occurrences of final states enclose a finite tree: Let

$$D_u := \{w \in u\cdot\{0,1\}^* \mid r(v) \notin F \text{ for all } v \text{ with } u < v \leq w\}.$$

D_u defines a finitely branching tree which does not contain an infinite path (since r is a successful run). So by König's Lemma D_u is finite, and the nodes of its outer frontier have r-values which are final states. This argument allows to decompose r into "layers" consisting of finite trees: The first layer consists of D_ε, and given the outer frontier F_n of the n-th layer, the (n+1)-th layer is the union of all D_u with $u \in F_n$.

As a consequence we can represent $T(\mathcal{A})$ using recognizable sets of finite trees. We refer to trees from the set $T_{A\cup F}$ which have values from F exactly at the frontier and are otherwise valued in A. (If t is such a tree, $\bar{t}$ results from t by deleting the F-valued frontier.) Now for $q \in Q$ let T_q consist of all such trees t where there is a run of $\mathcal{A}$ on $\bar{t}$ which starts in q and reaches on $fr^+(\bar{t})$ exactly the states of fr(t). Each set T_q is recognizable. The argument above shows that an infinite tree is accepted by the Büchi tree automaton $\mathcal{A}$ iff it belongs to

$$(*) \qquad T_{q_o} \cdot^{\mathbf{q}} (T_{q_1},\dots,T_{q_m})^{\omega,\mathbf{q}}$$

where $\mathbf{q} = (q_1,\dots,q_m)$ is the sequence of all final states in F. Conversely, it is easy to see that for <u>any</u> (m+1)-tuple $(T_o,T_1,\dots,T_m)$ of recognizable sets of finite trees, the expression corresponding to (*) defines a Büchi recognizable set of infinite trees:

9.1. Theorem

A set $T \subseteq T^{\omega}_{A}$ is Büchi recognizable iff there are recognizable sets $T_o,T_1,\dots,T_m \subseteq T_{A\cup C}$ (where $C = \{c_1,\dots,c_m\}$) such that $T = T_o\cdot^{c}(T_1,\dots,T_m)^{\omega c}$. ❑

The representation is implicit in Rabin's [1970] solution of the emptiness problem; for a stronger statement see Takahashi [1986] (where also rational expressions for tree languages are introduced).

Using the sets T_q above, the emptiness problem for Büchi tree automata is shown decidable. For this we set up an algorithm which eliminates step by step those states of a given Büchi tree automaton $\mathcal{A} = (Q,q_o,\Delta,F)$ which are useless for successful runs. Certainly a state q cannot appear in a successful run if the set T_q is empty. So eliminate successively those states q from $\mathcal{A}$ where T_q is empty and update the transition relation of $\mathcal{A}$ accordingly. (Note that each T_q is recognizable, so its emptiness can be checked in polynomial time by 8.1(a).) The elimination procedure stops after at most $|Q|$ steps, delivering a state set Q_o. We claim that $\mathcal{A}$ accepts some infinite tree iff Q_o still contains the initial state q_o (which establishes the desired algorithm). To prove this, assume q_o is not eliminated and let $q_1,\dots,q_m$ be the final states remaining in Q_o; note that $m \geq 1$ by nonemptiness of T_{q_o} and that also $T_{q_1},\dots,T_{q_m}$ are nonempty. So the set (*) as displayed above is nonempty, which is a subset of $T(\mathcal{A})$. - Conversely, it is clear that in case $T(\mathcal{A}) \neq \emptyset$ the state q_o will not be eliminated. A closer analysis of the algorithm yields also a polynomial complexity bound:

9.2. Theorem (Rabin [1970], Vardi, Wolper [1986])

The emptiness problem for Büchi tree automata is decidable; moreover, it is logspace complete for PTIME. ❑

An example tree t in a nonempty Büchi recognizable set T can be obtained by choosing finite trees $t_o,\dots,t_m$ from the sets $T_{q_o},\dots,T_{q_m}$ which the above algorithm produces, and setting $t = t_o\cdot(t_1,\dots,t_m)^{\omega}$. This infinite tree t is "finitely generated" from $t_o,\dots,t_m$: There are only finitely many distinct subtrees t_u in t (their number is bounded by $\Sigma_i|dom(t_i)|$). Trees with this property are called "regular". An equivalent definition states that $t:\{0,1\}^* \to A$ is a <u>regular tree</u> if there is for each letter $a \in A$ a regular expression r_a which defines the language $\{u \in \{0,1\}^* \mid t(u) = a\}$. In terms of automata this means that there is a finite automaton $\mathcal{A}$ over finite words which "generates t", i.e.

whose state set is partitioned into sets Q_a ($a \in A$) such that $\mathcal{A}$ reaches a state in Q_a via input u iff $t(u) = a$.

Regular trees represent the most basic infinite terms and hence are fundamental in several areas of computer science, for instance in semantics of program schemes (where they appear as unravellings of finite flowcharts) and in the foundations of logic programming. Courcelle [1983] is a survey which covers the basic theory and applications in semantics. For the role of regular trees in logic programming see e.g. Colmerauer [1982]. In Braquelaire, Courcelle [1984] the complexity of (term based) rational expressions for given regular trees is studied.

Extending the above consideration on Büchi recognizable sets, we shall show that also any nonempty Rabin recognizable set contains a regular tree, and thereby see that the emptiness problem for Rabin tree automata is decidable. Since unary regular trees are ultimately periodic ω-words, this generalizes Theorem 1.3 on Büchi automata in a natural way.

9.3. Theorem (Rabin [1972])

(a) Any nonempty Rabin recognizable set of trees contains a regular tree.

(b) The emptiness problem for Rabin tree automata is decidable (in exponential time).

Proof. (a) First reduce the problem to "input-free" tree automata with a transition relation $\Delta \subseteq Q \times Q \times Q$. For this, transform a given Rabin tree automaton $\mathcal{A} = (Q,q_0,\Delta,\Omega)$ over A into $\mathcal{A}' = (Q{\times}A,Q_0,\Delta',\Omega')$, where $\Delta' \subseteq (Q{\times}A){\times}(Q{\times}A){\times}(Q{\times}A)$ contains a transition $((q,a),(q',a'),(q'',a''))$ iff $(q,a,q',q'') \in \Delta$. Q_0 contains all states (q_0,a), and Ω' those pairs of sets of states from $Q{\times}A$ where the Q-components yield a pair from Ω. Then the successful runs r' of $\mathcal{A}'$ are the pairs r^t where r is a successful run of $\mathcal{A}$ on t; and in this case r is regular provided r' is regular. A corresponding statement holds if $\mathcal{A}'$ has been reduced to an automaton with a single initial state (as mentioned after 8.1).

So it suffices to show that an input-free Rabin tree automaton with some successful run admits also a regular successful run. Let $\mathcal{A} = (Q,q_0,\Delta,\Omega)$ be a Rabin tree automaton with $\Delta \subseteq Q{\times}Q{\times}Q$. Call a state $q \in Q$ live if $q \neq q_0$ and the automaton is not forced to stay in q by the single available transition (q,q,q). Using induction on the number of live states of $\mathcal{A}$ we transform a given successful run r into a regular successful run.

If there are no live states, the run r will be stationary from the sons of its root onwards and hence be regular.

In the induction step, distinguish three cases. First assume that in r some live state q

of $\mathcal{A}$ is missing. Then the induction hypothesis can be applied to the automaton where q has been cancelled, and we obtain the desired regular run of $\mathcal{A}$. - Secondly suppose in r there is a node u such that $r(u) = q$ is live but some live state q' does not appear beyond node u (viewing run trees top-down). We find two regular runs r_1, r_2, replacing the r-parts "up to first occurrences of q" and "from first occurrences of q onwards", such that the regular run $r_1 \cdot^q r_2$ is successful for $\mathcal{A}$. Run r_1 is obtained by declaring q as non-live in $\mathcal{A}$, i.e. we consider the modified automaton where only transition (q,q,q) is available for q and apply the induction hypothesis. Run r_2 is found from the modified automaton where q is taken as initial state and q' is deleted, again by induction hypothesis.

It remains to treat the case that all live states appear in r beyond any given node. Then we may choose a path π_0 through r where all live states appear again and again (and hence no non-live states can occur). Since r is successful, there is an accepting pair, say (L_1,U_1), such that $In(r|\pi_0) \cap L_1 = \emptyset$ and $In(r|\pi_0) \cap U_1 \neq \emptyset$. Note that L_1 contains only non-live states, since $In(r|\pi_0)$ is the set of live states. Pick q from $In(r|\pi_0) \cap U_1$. Again we find two regular runs r_1, r_2, now aiming at the property that the regular run $r_1 \cdot^q r_2^{\omega q}$ is successful for $\mathcal{A}$. Run r_1 is given as in the second case above. Run r_2 is obtained from a modification of $\mathcal{A}$, where q is taken as initial state and used non-live when revisited the first time; to this modified automaton the induction hypothesis can be applied. In the verification that $r_1 \cdot^q r_2^{\omega q}$ is indeed successful, the interesting case concerns those paths π where q appears infinitely often. We show for such π that (L_1,U_1) is an appropriate accepting pair: Concerning U_1 it suffices to note that $q \in U_1$. Suppose some L_1-state q' occurs infinitely often on π. Since L_1 contains only non-live states, q' is non-live and hence must be the only state occurring infinitely often on π; but this contradicts the fact that already the (live) state q occurs infinitely often.

(b) As in the preceding proof, it is enough to consider input-free automata. Suppose an input-free Rabin automaton $\mathcal{A}$ has to be checked for existence of a successful run. Assume $\mathcal{A}$ has n live states. Only finitely many automata can be obtained from $\mathcal{A}$ by the above mentioned modifications which reduce the number of live states by 1. Iterating these reductions, we obtain in a constructive way finitely many automata derived from $\mathcal{A}$, where the number i of live states ranges from n to 0. In case $i = 0$ (non-live states only), it is trivial to decide whether a successful run exists. For an automaton with i+1 live states, the proof above shows how to decide existence of a successful run, given this information for the automata with i live states. Using n such steps the answer concerning $\mathcal{A}$ is computed. The analysis of this algorithm (see Rabin [1972]) yields an exponential time bound for its execution. ❑

10. Complementation and determinacy of games

In this survey it is not possible to give a proof of the complementation theorem for Rabin tree automata, the most intricate part of Rabin [1969]. We shall explain, however, a treatment of this problem in the framework of "infinite games". This approach was proposed by Büchi [1977,1983] and Gurevich, Harrington [1982]. It allows to study the complementation problem in the context of descriptive set theory and clarifies interesting connections between tree automata and sequential automata. Moreover, some of the earliest problems on ω-automata, the questions of Church [1963] on "solvability of sequential conditions", can be settled via these connections.

We consider infinite games as studied by Gale, Stewart [1953] and Davis [1964]. Let A and B be alphabets (each with at least two letters) and let $\Gamma \subseteq (A\times B)^\omega$ be an ω-language. Γ defines a game between two players I and II, where a single play is performed as follows: First I picks some $a_0 \in A$, then II picks some $b_0 \in B$, then I some $a_1 \in A$, and so on in turns. Player I wins the play if the resulting ω-word $(a_0,b_0)(a_1,b_1)\ldots$ is in Γ, otherwise II wins. (If $\alpha = a_0a_1\ldots$ and $\beta = b_0b_1\ldots$ we shall denote the sequence $(a_0,b_0)(a_1,b_1)\ldots$ by $\alpha\wedge\beta$.) A <u>strategy</u> for I is a function $f:B^* \to A$, telling I to choose $a_n = f(b_0\ldots b_{n-1})$ if II has chosen $b_0,\ldots,b_{n-1}$. The strategy f induces a transformation $\bar{f}:B^\omega \to A^\omega$; if II builds up β and I plays strategy f, the play $\bar{f}(\beta)\wedge\beta$ will emerge. We say that f is a <u>winning strategy</u> for I if for all $\beta \in B^\omega$, $\bar{f}(\beta)\wedge\beta \in \Gamma$. If there is such a strategy for I we say that I <u>wins</u> Γ. Analogous definitions apply to player II (where a strategy for II is a map $g:A^+ \to B$, inducing a transformation $\bar{g}:A^\omega \to B^\omega$).

A game Γ is called <u>determined</u> if player I wins Γ or player II wins Γ. Determinacy of infinite games is a central topic of descriptive set theory, closely related to the continuum problem (see e.g. Moschovakis [1980, Chapter 6]). The claim that Γ is determined amounts to an infinitary version of a quantifier law:

$$\neg\exists a_0\forall b_0\exists a_1\forall b_1 \ldots (a_0,b_0)(a_1,b_1)\ldots \in \Gamma \quad (\text{"I does not win }\Gamma\text{"})$$
$$\text{iff } \forall a_0\exists b_0\forall a_1\exists b_1 \ldots (a_0,b_0)(a_1,b_1)\ldots \notin \Gamma \quad (\text{"II wins }\Gamma\text{"}).$$

But the assumption that <u>all</u> games Γ are determined is a strong set-theoretic hypothesis which contradicts the axiom of choice. For "nice" games Γ, however, determinacy has been shown: By Martin [1975], a game $\Gamma \subseteq (A\times B)^\omega$ is determined provided Γ belongs to the Borel hierarchy. In the sequel, the relevance of determinacy lies in the fact that it allows to transform the statement

$$\neg\exists \text{ strategy } f\ \forall\beta\ \bar{f}(\beta)\wedge\beta \in \Gamma \quad (\text{"I does not win }\Gamma\text{"})$$

into the form

$$\exists \text{ strategy } g\ \forall\alpha\ \alpha\wedge\bar{g}(\alpha) \notin \Gamma \quad (\text{" II wins }\Gamma\text{"})$$

and hence to write the negation of an existential statement again as an existential statement. Complementation of Rabin automata is a natural application, since it requires to express nonexistence of successful runs by one automaton as the existence of successful runs by the complement automaton.

For this purpose the following observation is crucial: Runs of tree automata (and, more generally, valued trees) are strategies. Just view a strategy $f:B^* \to A$ as a $|B|$-ary A-valued tree: its nodes are represented by the words from B^* (the root corresponding to the empty word), and the value $a \in A$ at node w indicates that $f(w) = a$. Similarly, strategies $g:A^+ \to B$ are $|A|$-ary B-valued trees with a default value at the root. If such a tree is <u>regular</u>, it codes a special kind of strategy: Since in this case the value of the tree say at a node w is computable by a finite automaton (determined by the state reached after reading w), the corresponding strategy is "executable by a finite automaton", or shorter: a <u>finite state strategy</u>. Thus for a finite state strategy f, the choice f(w) depends only on uniformly bounded finite information in w.

We now specialize the games Γ in two ways : First Γ is defined in terms of automata. The key example are the <u>regular games</u> $\Gamma \subseteq (A\times B)^\omega$, i.e. games which are recognized by Büchi (or Muller) automata when considered as ω-languages over A×B. Note that in this case Γ belongs to the Borel hierarchy; hence by Martin's result stated above regular games are determined. (Since by 5.2 a regular game Γ is even in the boolean closure of the Borel class F_σ, determinacy may be inferred from an easier result of Davis [1964].) Secondly, we impose also corresponding restrictions on the two players: their strategies are now required to be <u>finite state</u>. So the following sharpened questions on determinacy arise :

<u>Solvability</u>: Given (a presentation of) a regular game Γ, can one decide effectively who wins Γ ?

<u>Synthesis</u>: Can one exhibit a finite state winning strategy for the winner of a regular game Γ ?

Both problems were proposed by Church [1963], referring, however, to different motivation and terminology. Γ was considered as a "sequential condition" on pairs of sequences (i.e., $\Gamma \subseteq A^\omega \times B^\omega$), expressed in the sequential calculus as a "synthesis requirement" for digital circuits. The circuits should realize a transformation producing for any input sequence β a sequence α such that $(\alpha,\beta) \in \Gamma$.

Church's problems were solved positively by Landweber in his thesis (cf. Büchi, Landweber [1969]). We sketch here a short proof due to Rabin [1972] which exploits the correspondence between strategies and trees :

10.1. Theorem (Büchi, Landweber [1969])

Regular games are determined in the following strong sense: It can be decided effectively who wins, and the winner has a finite state winning strategy.

Proof (Rabin [1972]). As a typical example consider the case $A=\{0,1\}$, $B=\{0,1\}$. Let $\Gamma \subseteq (\{0,1\}\times\{0,1\})^\omega$ be regular, say recognized by the Muller automaton $\mathcal{M} = (Q,q_o,\delta,\mathcal{F})$ over $\{0,1\}\times\{0,1\}$. We transform $\mathcal{M}$ into a (deterministic) tree automaton $\mathcal{R} = (Q,q_o,\bar{\delta},\mathcal{F})$ over $\{0,1\}$, by defining

$$\bar{\delta}(q,a) = (q',q'') \quad \text{iff} \quad \delta(q,(a,0)) = q' \text{ and } \delta(q,(a,1)) = q'' \quad (a\in\{0,1\}).$$

$\mathcal{R}$ accepts a tree $t \in T_A^\omega$ iff along all paths β the states assumed infinitely often by $\mathcal{R}$ form a set in $\mathcal{F}$. This means that for all $\beta = d_1d_2...$ (where $d_i\in\{0,1\}$) the sequence $(t(\varepsilon),d_1)\ (t(d_1),d_2)\ (t(d_1d_2),d_3)\ ...$ is accepted by $\mathcal{M}$ and thus in Γ. Hence $\mathcal{R}$ accepts t iff t is a winning strategy for I in Γ. We may assume that $\mathcal{R}$ is (redefined as) a Rabin tree automaton. The existence of a winning strategy for I can now be decided effectively, by deciding nonemptiness of $T(\mathcal{R})$ (see 9.3(b)), and a finite state strategy is guaranteed in this case by 9.3(a). The case that II wins is handled similarly. ❑

The complementation problem for Rabin tree automata requires a more general type of game: With any Rabin tree automaton $\mathcal{A} = (Q,q_o,\Delta,\Omega)$ (accepting A-valued trees) and any tree $t \in T_A^\omega$ we associate a game $\Gamma_{\mathcal{A},t} \subseteq (\Delta\times\{0,1\})^\omega$. Thus player I picks transitions from Δ, and player II picks elements from $\{0,1\}$, i.e. directions building up a path through the tree t. $\Gamma_{\mathcal{A},t}$ contains all sequences $\alpha\wedge\beta \in (\Delta\times\{0,1\})^\omega$ which "describe a successful path for $\mathcal{A}$ on t": Formally,

$$\alpha\wedge\beta = ((s_o,a_o,s_o',s_o''),d_1)\ ((s_1,a_1,s_1',s_1''),d_2)\ ...$$

should satisfy $s_o = q_o$, $a_i = t(d_1...d_{i-1})$, $s_{i+1} = s_i'$ if $d_{i+1} = 0$, $s_{i+1} = s_i''$ if $d_{i+1} = 1$, and the state sequence $s_os_1s_2...$ should fulfill the acceptance condition Ω. Then the winning strategies $f:\{0,1\}^* \to \Delta$ for I are in one-to-one correspondence with the successful runs of $\mathcal{A}$ on t, and we have

(*) $\mathcal{A}$ accepts t iff I wins $\Gamma_{\mathcal{A},t}$.

Note that the underlying tree t is completely arbitrary and one can no more expect that the winning strategies are finite state. However, it turns out that relativized finite state strategies (as we call them) can be guaranteed. Such a strategy, say for

player II, is executed by a finite automaton $\mathcal{C}$ which is allowed to use an auxiliary tree $t' \in T_{A'}^\omega$, over an alphabet A' in addition to the given tree $t \in T_A^\omega$ (and the transitions from Δ picked by player I). $\mathcal{C}$ works over $\Delta\times A\times A'$ and outputs directions from $\{0,1\}$ (via a partition of its state set into "0-states" and "1-states"). More precisely, consider the game situation

$$\begin{array}{ll} I : \bar{\tau} = & \tau_o\ \ \tau_1\ ...\ \ \tau_n \\ II : w = & \quad\ d_1\ ...\ d_n \end{array}$$

and denote by $(t|w)\wedge(t'|w)$ the sequence of $A\times A'$-values of $t\wedge t'$ for the nodes visited along w (namely ε, d_1, d_1d_2, ... , $d_1...d_n$). Then the state reached by $\mathcal{C}$ after reading the word $\bar{\tau}\wedge(t|w)\wedge(t'|w) \in (\Delta\times A\times A')^{n+1}$ fixes the next choice of player II.

Determinacy for the games $\Gamma_{\mathcal{A},t}$ by such relativized finite state strategies was stated by Büchi [1977] with a short proof hint and given a detailed exposition in Büchi [1983]. A different and simpler proof was given by Gurevich, Harrington [1982]. We state this difficult result here without proof and conclude from it the complementation for Rabin tree automata.

10.2. Theorem (Büchi [1977,1983], Gurevich, Harrington [1982]).

Let $\mathcal{A}$ be a Rabin tree automaton over A and $t \in T_A^\omega$. The game $\Gamma_{\mathcal{A},t}$ is determined, and the winner has a relativized finite state strategy. ❑

10.3. Corollary

For any Rabin tree automaton $\mathcal{A}$ there is (by effective construction) a Rabin tree automaton $\mathcal{A}'$ recognizing $T_A^\omega - T(\mathcal{A})$.

Proof of Corollary. Let $\mathcal{A} = (Q,q_o,\Delta,\Omega)$ be a Rabin tree automaton over A. We have to find a Rabin tree automaton $\mathcal{A}'$ such that for all $t \in T_A^\omega$

$\mathcal{A}$ does not accept t iff $\mathcal{A}'$ accepts t.

By (*) above, $\mathcal{A}$ does not accept t iff I does not win $\Gamma_{\mathcal{A},t}$. By 10.2, this holds iff II wins $\Gamma_{\mathcal{A},t}$ in the following sense, using a finite automaton $\mathcal{C}$:

for some tree $t' \in T_{A'}^\omega$, over an auxiliary alphabet A': if $\alpha \in \Delta^\omega$ is proposed by I and $\beta \in \{0,1\}^\omega$ is the response by II (computed by $\mathcal{C}$ from α,t,t'), then $\alpha\wedge\beta$ does not describe a successful path for $\mathcal{A}$ on t.

This can be formulated as follows:

(1) for some $t' \in T^{\omega}_{A'}$:

(2) for all $\beta \in \{0,1\}^{\omega}$:

(3) for all $\alpha \in \Delta^{\omega}$: if β results from α,t,t' by output of $\mathcal{C}$, then α^β does not describe a successful path for $\mathcal{A}$ on t.

Note that (3) is a condition on sequences of the form β^$(t|\beta)$^$(t'|\beta) \in (\{0,1\}\times A\times A')^{\omega}$, (2) is a condition on trees from $T^{\omega}_{A\times A'}$, and (1) a condition on trees from T^{ω}_{A}. Since (3) is easily expressed in S1S, it can be defined by a Muller automaton $\mathcal{M}$. Now construct from $\mathcal{M}$ a deterministic tree automaton $\mathcal{R}$ as in the proof of 10.1. Then $\mathcal{R}$ accepts the trees satisfying (3) on each path β, i.e. all trees $t \in T^{\omega}_{A\times A'}$ with property (2). Projection to T^{ω}_{A} yields a (nondeterministic) Rabin automaton $\mathcal{A}'$, accepting the trees $t \in T^{\omega}_{A}$ with property (1). Since these trees were just the trees for which II wins $\Gamma_{\mathcal{A},t}$, $\mathcal{A}'$ is a Rabin tree automaton as desired. ❑

The complementation problem for Rabin tree automata has been (and continues to be) investigated by several authors. Muchnik [1985] presents an elegant proof using an induction over the number of states in the automata. An interesting approach has recently been developed by Muller, Schupp [1987], based on the idea of alternating automata over trees. In addition to performing nondeterministic choice, these automata are able to pursue several computations simultaneously. The operation of an alternating automaton with state set Q is described by the elements of the free lattice over $\{0,1\}\times Q$. The intended meaning of such an element, say

$$((0,q_1) \wedge (1,q_2)) \vee ((1,q_1) \wedge (1,q_2)),$$

is that the automaton proceeds from a given node with q_1 to the left and q_2 to the right, or proceeds with q_1 to the right and also with q_2 to the right. The second possibility is missing in Rabin tree automata. One can now collect all possible histories of the alternating automaton $\mathcal{A}$ over t in a computation tree $C(\mathcal{A},t)$, where one history follows one simultaneous realization of states through the levels of the tree. (So a history is a kind of "multirun".) $\mathcal{A}$ accepts t if for some infinite history, i.e. some path in $C(\mathcal{A},t)$, all state sequences along paths described by it are successful in the sense of Muller acceptance. Complementation for these alternating automata is easy, performed by dualizing the given automaton (exchange $\wedge$ and $\vee$ in the transitions, and complement the system of final state sets). The hard closure property is projection, which is shown again by an application of 10.2. An advantage of this approach is that fragments of the monadic theory of the tree which are defined in terms of restricted second-order quantifiers may be handled by suitable restrictions of the projection operation for alternating automata and hence by weakened versions of 10.2.

11. Monadic tree theory and decidability results

For a transfer of the preceding results from automata theory to logic, trees are represented as model theoretic structures. If A is the alphabet $\{0,1\}^n$, a tree $t \in T^{\omega}_{A}$ is coded by a model of the form

$$\underline{t} = (\{0,1\}^*, \varepsilon, succ_0, succ_1, <, P_1,\dots,P_n),$$

where $succ_0$, $succ_1$ are the two successor functions over $\{0,1\}^*$ with $succ_0(w) = w0$, $succ_1(w) = w1$, $<$ is the prefix relation over $\{0,1\}^*$, and $P_1,\dots,P_n$ are subsets of $\{0,1\}^*$ with $w \in P_i$ iff the i-th component of t(w) is 1. We introduce the interpreted formalism S2S ("second-order theory of two successors") as the corresponding monadic second-order language with the canonical interpretation in these models. The language contains variables x,y,.. and X,Y,... (ranging over elements, resp. subsets of $\{0,1\}^*$). Terms are obtained from the individual variables x,y,... and the constant ε by applications of $succ_0$, $succ_1$; we write x0 instead of $succ_0(x)$ etc. Atomic formulas are of the form $t \in X$, $t = t'$, $t < t'$ where t,t' are terms and X is a set variable; and arbitrary formulas are generated from atomic formulas by boolean connectives and the quantifiers $\exists,\forall$ (ranging over either kind of variable). If $\varphi(X_1,\dots,X_n)$ is a S2S-formula and $\underline{t}$ a tree model as above, we write $\underline{t} \models \varphi(X_1,\dots,X_n)$ if φ is satisfied in $\underline{t}$ with P_i as interpretation for X_i. Let $T(\varphi) = \{t \in T^{\omega}_{A} | \underline{t} \models \varphi(\bar{X})\}$. If $T = T(\varphi)$ for some S2S-formula φ, T is called definable in S2S.

The system WS2S is obtained when the set quantifiers range over finite subsets of $\{0,1\}^*$ only. If $T = T(\varphi)$ for some WS2S-formula φ (i.e., some S2S-formula using this "weak interpretation"), T is definable in WS2S (or simply: "weakly definable").

The above definitions are analogously applied to finite tree models $\underline{t}$ (where $t \in T_A$). In this case there is no difference between the weak and the strong interpretation.

Note that S1S (as introduced in section 3) results from S2S by deleting the successor function $succ_1$ and by restriction of the underlying models to the domain 0*. Similarly to the case of S1S, the primitives ε and $<$ are definable in terms of $succ_0$, $succ_1$ and hence could be cancelled; we use them for easier formalizations. By "the infinite binary tree" (as a model theoretic structure) we shall mean the structure $(\{0,1\}^*, succ_0, succ_1)$.

We list some examples of S2S-formulas (using abbreviations such as $x \leq y$, $X \subseteq Y$, etc.):

Chain(X): $\forall x\forall y(x \in X \wedge y \in X \rightarrow x < y \vee x = y \vee y < x)$

Path(X): $Chain(X) \wedge \neg\exists Y(X \subseteq Y \wedge X \neq Y \wedge Chain(Y))$

$x \langle y$: $x \leq y \vee \exists z(z0 \leq x \wedge z1 \leq y)$
(This is the total lexicographical ordering of $\{0,1\}^*$.)

Fin(X): $\forall Y(Y \subseteq X \wedge Y \neq \emptyset \rightarrow (\exists y$ "y is $\langle$-minimal in Y" $\wedge \exists y$ "y is $\langle$-maximal in Y"))
(This shows definability of finiteness in S2S; hence WS2S can be interpreted in S2S.)

As an example tree language definable in S2S consider the set T_o of section 8; it is defined as follows (identifying letters a,b with 1,0):

$$\exists Y(\text{Path}(Y) \wedge \forall x(x \in Y \rightarrow \exists y(y \in Y \wedge x < y \wedge y \in X_1))).$$

We now can state the analogue of Büchi's Theorem 3.1 for sets of trees:

11.1. Theorem

(a) (Thatcher, Wright [1968], Doner [1970])
A set $T \subseteq T_A$ of finite trees is definable in (W)S2S iff T is recognizable.

(b) (Rabin [1969])
A set $T \subseteq T_A^\omega$ is definable in S2S iff T is Rabin recognizable.

Proof. Assume $A = \{0,1\}^n$. For the implications from right to left formalize the acceptance condition for the given tree automaton $\mathcal{A}$: Suppose, for (b), that the Rabin tree automaton $\mathcal{A}$ has the states 0,...,m and the accepting pairs $(L_1,U_1),\ldots,(L_r,U_r)$. $T(\mathcal{A})$ is defined by an S2S-formula which says

$$\exists Y_o \ldots \exists Y_m(\text{"}Y_o,\ldots,Y_m \text{ represent a run of } \mathcal{A} \text{ on } X_1,\ldots,X_n\text{"} \wedge \forall Z(\text{Path}(Z) \rightarrow \bigvee_{1\leq i\leq r}(\bigwedge_{j\in L_i} \text{"}\exists^{<\omega}x(x \in Z \wedge x \in Y_j)\text{"} \wedge \bigvee_{j\in U_i} \text{"}\exists^{\omega}x\,(x \in Z \wedge x \in Y_j)\text{"}))).$$

The converse is also shown similarly to 3.1: First S2S is reduced to a pure second-order formalism $S2S_o$ with atomic formulas of the form $Succ_0(X_i,X_j)$, $Succ_1(X_i,X_j)$, $X_i \subseteq X_j$ only. Induction over $S2S_o$-formulas $\varphi(X_1,\ldots X_n)$ shows recognizability, resp. Rabin recognizability of $T(\varphi)$. The steps for $\vee$ and $\exists$ are easy since the (nondeterministic) automata are closed under union and projection. (For (a) apply 8.1(c); the proof for Rabin tree automata is similar.) Concerning negation, use again 8.1(c), resp. the complementation result 10.3. ❑

Büchi tree automata correspond to a proper fragment of S2S which still allows to express many interesting tree properties; they also are used for a beautiful characterization of WS2S. We state the result here only with hints for the proof of the easier part (a):

11.2. Theorem (Rabin [1970])

Let $A = \{0,1\}^n$.

(a) A set $T \subseteq T_A^\omega$ is Büchi recognizable iff T is definable by a S2S-formula $\exists Y_1 \ldots \exists Y_m \varphi(Y_1,\ldots,Y_m,X_1,\ldots,X_n)$ where φ is a WS2S-formula.

(b) A set $T \subseteq T_A^\omega$ is definable in WS2S iff T and T_A^ω-T are Büchi recognizable.

Proof. (of (a)). For a description of a Büchi recognizable set of trees by an existential closure of a weak formula the representation of 9.1 can be used: The formula says that sets Y_q exist (where q ranges over the final states of the given automaton (Q,q_o,Δ,F)) which define a decomposition of the tree into finite subtrees; such a finite subtree is required to be accepted by the (finite-) tree automaton (Q,q,Δ,F) iff its root is in Y_q. The converse direction is more involved and requires a nontrivial closure property of Büchi tree automata, concerning universal quantification over finite sets. We omit the details. ❑

A direct automata theoretic characterization of WS2S (in terms of alternating automata over trees) is given by Muller, Saoudi, Schupp [1986].

In 4.6 it was shown that S1S and WS1S are expressively equivalent. From 11.2(a) and the failure of complementation for Büchi tree automata (cf. 8.2) it follows that WS2S is strictly less expressive than S2S and even than Büchi tree automata. However, this applies only to formulas $\varphi(X_1,\ldots,X_n)$ which speak about sets: Läuchli, Savioz [1987] have shown that any S2S-formula $\varphi(x_1,\ldots,x_n)$, where only individual variables occur free, can be expressed as a WS2S-formula.

A possible application of the above equivalence theorems is the analysis of variants of Büchi, resp. Rabin tree automata. Let us mention two such automaton models: the subtree automaton of Vardi, Wolper [1986] and the hybrid automaton of Vardi, Stockmeyer [1985].

A subtree automaton over A is of the form $\mathcal{A} = (Q,\Delta,f,F)$ where Q,Δ,F are as for Büchi tree automata and $f{:}A \rightarrow Q$. A tree $t \in T_A^\omega$ is accepted by $\mathcal{A}$ if all its nodes x are roots of finite trees accepted by the tree automaton $\mathcal{A}' = (Q,f(x),\Delta,F)$. Hence the properties recognized by subtree automata are definable by formulas

(+) $\forall x \exists X\, \varphi(x,X)$

where $\varphi(x,X)$ expresses that the finite tree with root x and (finite) frontier X is recognized by $\mathcal{A}'$. Since (+) is a WS2S-formula, subtree automata recognize only WS2S-definable sets and hence are a proper specialization of Büchi tree automata. Subtree automata are tailored for obtaining good upper complexity bounds for program logics. Vardi, Wolper [1986] show that for several program logics the satisfiability problem amounts to a test on existence of certain trees ("Hintikka-trees") which are defined in terms of eventuality properties of the form (+).

The hybrid automaton of Vardi, Stockmeyer [1985] is also introduced as a tool for the analysis of program logics. It is a pair $(\mathcal{A},\mathcal{B})$ where $\mathcal{A}$ is a Rabin tree automaton and $\mathcal{B}$ a (sequential!) Büchi automaton. A tree $t \in T_A^\omega$ is accepted if $\mathcal{A}$ accepts t and $\mathcal{B}$ rejects all paths of t (identified with ω-words over A). Hence hybrid automata generalize Rabin tree automata. However, since their acceptance condition is easily expressed in S2S, they recognize the same sets as Rabin automata. The application in program logics rests on the fact that the emptiness problem for hybrid automata is harder than for Rabin tree automata by one exponential; on the other hand the known transformation into Rabin tree automata is of doubly exponential complexity. This leads to a gain of one exponential in the solution of the satisfiability problem for certain program logics.

We turn to applications of theorem 11.1 in decision problems. The starting point is the following fundamental result:

11.3. Rabin's Tree Theorem [1969]

The monadic second-order theory of the infinite binary tree is decidable.

Proof. For any S2S-sentence φ one can construct, by the proof of 11.1, an input-free Rabin tree automaton $\mathcal{A}_\varphi$ such that φ is true in the infinite binary tree iff $\mathcal{A}_\varphi$ has a successful run. The latter condition is decided effectively by 9.3. ❑

The result is easily generalized to the monadic second-order theory of the full n-ary tree, where n sucessor functions $succ_0$, ..., $succ_{n-1}$ are allowed in the formulas. Similarly, the monadic second-order theory SωS of countably branching trees is proved decidable; here one usually refers to the signature with < (prefix relation over ω*) and ⟨ (lexicographic order over ω*), because each of the infinitely many successor functions $succ_i$ is definable in terms of < and ⟨.

A large number of theories of mathematical logic have been shown to be decidable via Rabin's Tree Theorem; as examples already established by Rabin [1969] we mention: the monadic second-order theory of countable orderings, the monadic second-order theory of unary functions over countable domains, and the theory of Boolean algebras with second-order quantification over ideals.

In the following, we outline a standard application in dynamic logic, concerning the solution of the satisfiability problem for modal logics of programs. The method (and refinements of it) have been used for several logics, for example propositional dynamic

logic and extensions (Streett [1982]), process logic (Harel, Kozen, Parikh [1982]), the calculus L_μ (Streett, Emerson [1984]), and computation tree logic CTL* (Emerson, Sistla [1984], Emerson, Halpern [1986]). In all instances the satisfiability question "Is there a model $\mathcal{M}$ satisfying the formula φ?" is effectively transformed to a question "Is there a tree t satisfying the S2S-sentence φ'?" (Logics allowing this reduction are said to share the tree model property.) We present the conceptually simplest form of this translation for the example CTL* of computation tree logic. Further developments of the method (with much better complexity bounds) are surveyed in Emerson [1988]. Muller, Saoudi, Schupp [1988] present a uniform method to obtain exponential time bounds, based on the alternating automata mentioned at the end of section 10.

Computation tree logic CTL* is a system of modal logic which allows to specify properties of paths through Kripke structures. From atomic propositions, say $p_1,...,p_n$ for the following discussion, the CTL*-formulas are built up using boolean connectives, the linear time temporal operators O,◊,□,U, and the additional unary operator E. Recall (from section 6) that a Kripke structure is of the form $\mathcal{M} = (S,R,\Phi)$ where S is a (here at most countable) set of states, $R \subseteq S\times S$ the transition relation, and $\Phi: S \to 2^{\{p_1,...,p_n\}}$ a truth valuation. For simplicity we assume that there is a distinguished start state s_0. The semantics of CTL*-formulas in Kripke structures is based on the usual meaning of O,◊,□,U over given state paths and the interpretation of E by "there is an infinite state path". Consider an example: The CTL*-formula

$$p_1 \wedge E(\Box p_2 \wedge \Diamond E \Diamond p_1)$$

says that

"p_1 is true in s_0, and starting from s_0 there is an infinite path π through the model such that all states on π satisfy p_2, and in some state of π a path π' starts with some state satisfying p_1".

The decision procedure for satisfiability of CTL*-formulas is based on the unravelling of Kripke structures in tree form: Given the Kripke structure $\mathcal{M} = (S,R,\Phi)$, define the structure $\mathcal{M}' = (S',R',\Phi')$ by

$$S' = S^+,$$
$$(r_1...r_m)R'(s_1...s_k s) \text{ iff } k = m, r_i = s_i \text{ for } 1\le i\le k, s_k R s,$$
$$\Phi'(s_1...s_k) = \Phi(s_k).$$

$\mathcal{M}'$ can be considered as a (at most countably branching) $\{0,1\}^n$-valued tree in which the relation "is father of" represents R' and where the valuation represents Φ'. $\mathcal{M}'$ is encoded over the binary tree by the map $n_1...n_r \to 10^{n_1}...10^{n_r}$. We obtain a $\{0,1\}^{n+1}$-valued binary tree $t_{\mathcal{M}'}$ where the additional component of the valuation indicates the range of $\mathcal{M}$ under this map. Let $\mu(X,Y_1,\ldots,Y_n)$ be a S2S-formula which says that $X \subseteq (10^*)^*$ is a range of a tree under the coding and that $Y_1,\ldots,Y_n \subseteq X$. Any tree of form $t_{\mathcal{M}}$ satisfies μ; conversely any tree satisfying μ induces a Kripke model (over $(10^*)^*$).

It is straightforward to reformulate a given CTL*-formula φ as a S2S-formula $\varphi'(X,Y_1,\ldots,Y_n)$ such that it is true over a tree $t_{\mathcal{M}}$ iff φ holds in $\mathcal{M}$. Hence φ is satisfiable iff the S2S-sentence

$$\exists X \exists Y_1 \ldots \exists Y_n\, (\mu(X,Y_1,\ldots,Y_n) \wedge \varphi'(X,Y_1,\ldots,Y_n))$$

is true over the infinite binary tree. So by Rabin's Tree Theorem satisfiability of CTL*-formulas is decidable.

The decision procedure induced by the above transformation is nonelementary. (Each level of negation in the given formula requires a corresponding complementation of a Rabin automaton and hence an at least exponential blow-up in the size of the automata.) Better procedures are obtained by incorporating more information than just for the atomic formulas in the tree model $t_{\mathcal{M}}$; a possible approach is to include the "Fischer-Ladner closure" of the given formula. For details see Emerson [1988].

We end this section with the formulation of two interesting generalizations of Rabin's Tree Theorem and some remarks on undecidable extensions of the monadic theory of the binary tree.

Stupp [1975], continuing work of Shelah [1975], extended Rabin's techniques (in particular, concerning the complementation theorem) to "higher-dimensional trees" and similar structures.

11.4. Theorem (Shelah [1975], Stupp [1975])

Let $\mathcal{M} = (M,(R_i)_{i<k})$ be a relational structure (say with binary relations $R_i \subseteq M\times M$). Define the structure $\mathcal{M}^* = (M^*,<,(R_i^*)_{i<k})$ by

$u < v$ iff u is a proper prefix of v (over M^*)

$u\, R_i^*\, v$ iff $\exists\, m_1,\ldots,m_k,m,m' \in M$ such that

$$u = m_1\ldots m_k m,\ v = m_1 \ldots m_k m',\ m\, R_i\, m'.$$

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

The special case of 11.4 which yields Rabin's Tree Theorem concerns the finite structure $\mathcal{M}_o = (\{0,1\},<_o)$ where $<_o$ is the usual order on $\{0,1\}$ (and which of course has a decidable monadic second-order theory). We have $\mathcal{M}_o^* = (\{0,1\}^*,<_1,<_o)$ where

$u <_1 v$ iff u is a proper prefix of v,

$u <_o v$ iff $u = w0$, $v = w1$ for some $w \in \{0,1\}^*$.

The functions $succ_o$, $succ_1$ are (monadic second-order) definable in terms of $<_1$, $<_o$ and vice versa. So $\mathcal{M}_o^*$ is essentially the infinite binary tree.

Another extension of Rabin's Tree Theorem, which also covers the models $\mathcal{M}_o^{*\cdots *}$, is due to Muller, Schupp [1985]. They consider infinite directed graphs with labeled edges. Such a graph is said to be "finitely generated" if it has a distinguished vertex v_o ("origin"), a finite label alphabet A and a fixed finite bound on the degrees of the vertices. The model theoretic structures which represent these graphs are of the form $\underline{G} = (G,v_o,(R_a)_{a\in A})$ such that $u\ R_a\ v$ iff there is an edge labeled a from u to v. A finitely generated graph is called a <u>context-free graph</u> if it is "finitely behaved at infinity" in the following sense: One obtains only finitely many distinct isomorphism types by collecting, for all vertices v, the substructures $\Gamma(v)$ which remain as connected components when the points with smaller distance to v_o than between v and v_o are deleted. The binary tree, with root v_o and labels 0 and 1 on edges pointing to left and right successors, respectively, is a simple example with only one such isomorphism type. (The terminology is motivated by a connection with group theory: As shown in Muller, Schupp [1983], a finitely generated group has a context-free word problem iff its Cayley graph is context-free in the sense above.)

11.5. Theorem (Muller, Schupp [1985])

The monadic second-order theory of any context-free graph is decidable. ❑

The proof of 11.5 is based on Rabin's Tree Theorem. The general problem of reducing monadic theories of graphs to theories of trees is further investigated in Seese [1988]; he considers the conjecture that <u>any</u> decidable monadic theory of a class of graphs has an interpretation in the monadic theory of a class of trees, and shows that this is true for the class of planar graphs. Courcelle [1988] presents a thorough analysis of monadic second-order properties of graphs in connection with an algebraic notion of recognizability.

The theories in 11.4 and 11.5 seem close to the margin of undecidability. We mention some variants, resp. extensions of the monadic theory of the binary tree which are undecidable.

The most basic example is the monadic theory of the "grid" $(\omega\times\omega,s_o,s_1)$ where $s_o(m,n) = (m+1,n)$ and $s_1(m,n) = (m,n+1)$. This structure is obtained from the free algebra $(\{0,1\}^*,succ_o,succ_1)$ by adding the relation $succ_o \cdot succ_1 = succ_1 \cdot succ_o$.

11.6. Theorem (Seese [1972])

The (weak) monadic second-order theory of the grid $(\omega\times\omega, s_0, s_1)$ is undecidable.

Proof. The idea is similar as in the undecidability proof for the origin constrained domino problem. For any Turing machine $\mathcal{A}$ construct a sentence $\varphi_{\mathcal{A}}$ in the weak monadic second-order language of the grid which expresses existence of a halting computation of $\mathcal{A}$ when $\mathcal{A}$ is started on the empty tape (the tape is assumed here left bounded and right infinite). As in the domino problem, the i-th cell of the j-th configuration is represented by point $(i,j) \in \omega\times\omega$. Using existential quantification over auxiliary predicates (which code the letters and states of $\mathcal{A}$) it is easy to formalize that a halting configuration is reached. Since only finitely many steps and a finite portion of the tape are involved, weak second-order quantification suffices. ❑

11.7. Corollary

(a) The (weak) monadic second-order theory of the infinite binary tree extended by the function s with $s(w) = 0w$ (for $w \in \{0,1\}^*$) is undecidable.

(b) The (weak) monadic second-order theory of the infinite binary tree extended by the "equal level predicate" E, given by $u\ E\ v$ iff $|u| = |v|$ (for $u,v \in \{0,1\}^*$) is undecidable.

Proof. (a) Identify the (weakly definable) subset 0^*1^* of the binary tree with $\omega\times\omega$. Note that $(\omega\times\omega, s_0, s_1)$ is isomorphic to $(0^*1^*, s, succ_1)$.

(b) Using the predicate E, the function s is weakly definable on 0^*1^*, since we have (for $u,v \in 0^*1^*$)

$$s(u) = v \text{ iff } \quad (u \in 0^* \wedge v = u0) \vee$$
$$\exists w \in 0^*\, \exists u'(u \in w1^* \wedge u' \in w01^* \wedge u\ E\ u' \wedge u'1 = v).$$

Now (a) can be applied. ❑

As a consequence of 11.7 (and by decidability of S2S) one obtains that the function s and the relation E are not definable in S2S; this is noted (also by other proofs) in Buszkowski [1980], Läuchli, Savioz [1987], and Emerson [1987].

12. Classifications of Rabin recognizable sets

In this final section we give a short overview of the (mostly ongoing) work which studies the "fine structure" of the class of Rabin recognizable sets of trees. The results presented here fall in three categories, depending on the formalism in which tree properties are classified: Monadic second-order logic, tree automata, and fixed point calculi.

12.1. Restrictions of monadic second-order logic

Natural subsystems of the monadic second-order formalism S2S are obtained when the range of set quantification is narrowed to special subsets of $\{0,1\}^*$.

First we consider the question for which restricted set quantifiers the same sentences are true (over the infinite binary tree) as in the case of quantification over arbitrary subsets. Rabin [1972] showed (as a corollary of his result given in 9.3(a) above) that the regular subsets of $\{0,1\}^*$ constitute such a restriction. Siefkes [1975] proved that this fails for the recursive subsets of $\{0,1\}^*$, using the fact that the "recursive version" of König's Lemma does not hold. (There are finitely branching infinite trees which are recursive but have no recursive infinite path.) - A related question is the uniformization problem: Is there for any given formula $\varphi(X,Y)$, such that $\forall X \exists Y \varphi(X,Y)$ holds, a "definable choice function", i.e. a (set) function defined by a formula $\psi(X,Y)$ such that $\forall X \forall Y(\psi(X,Y) \rightarrow \varphi(X,Y))$? Siefkes [1975] proves this for S1S, and Gurevich, Shelah [1983] show by a very intricate proof method that it fails for S2S.

We discuss two further restricted set quantifiers: those ranging over finite sets, and those ranging over paths through the infinite binary tree. (For other syntactic fragments of S2S see e.g. Mostowski [1981].)

The weakly definable sets of trees, already considered in the preceding section, have been further classified in Thomas [1982]. It is shown there that an infinite hierarchy is induced by the alternation depth of "unbounded" quantifiers over finite sets and elements. (The quantifier in $\exists X \varphi(X,Y)$ is called bounded if the formula is equivalent to $\exists X(X<Y \wedge \varphi(X,Y))$, where $X<Y$ abbreviates $\forall x(x\in X \rightarrow \exists y(y\in Y \wedge x\leq y))$.) As an example, consider the formula (+) of the preceding section for the description of subtree automata; (+) has two unbounded quantifiers, which shows that subtree automata recognize only sets up to the second level of that hierarchy. Over ω-words the analogous hierarchy is finite, because by McNaughton's Theorem two unbounded quantifiers suffice for the definition of regular ω-languages (cf. 4.6).

When set quantifiers refer to chains in trees (i.e. sets linearly ordered by the prefix relation $<$) or to paths (i.e. maximal chains), one obtains "chain logic", resp. "path logic" over the binary tree. In Thomas [1987] the sets of trees which are definable in these systems are characterized in terms of the regular and star-free ω-languages. A similar approach is developed by Clarke, Grumberg, Kurshan [1987] for a system of extended branching time logic. The close connection between path quantifiers and branching time logic is illustrated in Hafer, Thomas [1987], where computation tree logic CTL* and path logic are shown expressively equivalent, provided that only binary tree models are admitted.

The most general decidability result on path quantifiers over trees was shown by Gurevich, Shelah [1985]. By a combination of tree automata with a model theoretic composition technique (for finite fragments of theories), they prove that in the language of path logic the theory of arbitrary trees (considered as any partial orders where each set $\{y \mid y \leq x\}$ is totally ordered) is decidable.

12.2. Restrictions in Rabin tree automata

Several authors investigated the possibility of extending Landweber's Theorem (cf. section 5) to Rabin recognizable sets of trees. The notions of 1- and 2-acceptance can be transferred in a natural way from sequential automata to tree automata (namely, as conditions for all paths of a run). Also the Cantor topology is extended canonically from A^ω to the space T_A^ω (the sets $t \cdot T_A^\omega$, where t is a finite tree, form an open basis). Results on inclusion relations for these acceptance conditions and their topological meaning are presented in Hayashi, Miyano [1985] and Moriya [1987]. An infinite hierarchy of closed sets based on quantifier alternation is introduced in Mostowski [1987]. Mostowski, Skurczynski, Wagner [1985] obtain a partial transfer of Landweber's Theorem (including decidability results) to tree languages recognized by deterministic Rabin tree automata. Further results on deterministic tree automata and the power of several acceptance conditions derived from Muller and Büchi acceptance are given in Saoudi [1984,1986].

Recently, Niwinski [1986a] showed that the Rabin index (the number of accepting pairs in Rabin tree automata) defines an infinite hierarchy of sets of trees: For each n there is a set $T_n \subseteq T_A^\omega$ which is recognized only by Rabin tree automata with at least n accepting pairs. The example languages T_n belong to the boolean closure of the Büchi recognizable sets of trees; Hafer [1987] proves that this boolean closure is still properly contained in the class of Rabin recognizable sets.

Mostowski [1984] presents a "standard form" of Rabin tree automata in which an ordering of the state set enters the acceptance condition. As an application, a calculus of regular-like expressions is set up which allows to define the Rabin recognizable sets.

12.3. Fixed point calculi

Niwinski [1986] and Takahashi [1986] studied the specification of tree properties by a fixed-point calculus in which least and greatest fixed points are included. Following Niwinski [1986], we define the <u>μ-terms</u> over an alphabet A and with variables $x_1, x_2, \ldots$ by the clauses

- each variable x_i is a μ-term,
- if τ_1, τ_2 are μ-terms and $a \in A$, then $a(\tau_1,\tau_2)$ and $\tau_1 \cup \tau_2$ are μ-terms,
- if τ is a μ-term and x is a variable, then $\mu x\tau$ and $\nu x\tau$ are μ-terms.

Any μ-term $\tau(x_1, \ldots x_n)$ with free variables $x_1, \ldots, x_n$ defines a function F_τ: $(T_A^\omega)^n \rightarrow T_A^\omega$. For $\tau = x_i$ it is the i-th projection. If $\tau(x_1, \ldots, x_n)$ has the form $\tau = \tau_1 \cup \tau_2$, let

$$F_\tau(T_1, \ldots, T_n) = F_{\tau_1}(T_1, \ldots, T_n) \cup F_{\tau_2}(T_1, \ldots, T_n),$$

similarly for $\tau = a(\tau_1, \tau_2)$

$$F_\tau(T_1, \ldots, T_n) = \{t \in T_A^\omega \mid t(\varepsilon) = a,\ t_0 \in F_{\tau_1}(T_1, \ldots, T_n),\ t_1 \in F_{\tau_2}(T_1, \ldots, T_n)\ \}.$$

Finally, for $\tau = \mu y\tau_0(y, x_1, \ldots, x_n)$, resp. $\tau = \nu y\tau_0(y, x_1, \ldots, x_n)$, let $F_\tau(T_1, \ldots, T_n)$ be the least, resp. greatest fixed point of the function $T \mapsto F_{\tau_0}(T, T_1, \ldots, T_n)$. (These fixed points exist by the Knaster-Tarski Theorem.)

Each μ-term τ without free variables defines a set $T \subseteq T_A^\omega$, denoted here by $T(\tau)$. As an example over the alphabet $A = \{a,b\}$ consider the μ-term

$$\mu x_1\, \nu x_0\, (b(x_0, x_0) \cup a(x_1, x_1));$$

it defines the set T_1 of section 8, containing all trees such that on each path there are only finitely many letters a.

By induction over the μ-terms τ one verifies that the functions F_τ are definable in S2S. It follows that the sets $T(\tau)$ are Rabin recognizable. Niwinski [1988] proves also the converse; so the μ-terms have the same expressive power as Rabin tree automata (or S2S). Already in Niwinski [1986] it is shown that a strict hierarchy of sets of trees is generated by increasing the number of alternations between the least and greatest fixed point operators μ and ν in the defining terms. Moreover, the second level of the hierarchy (given by the terms where all ν-operators precede all μ-operators) characterizes the Büchi recognizable sets of trees (Niwinski [1986], Takahashi [1986]). Arnold, Niwinski [1987] show that this characterization continues to hold when the intersection operator $\cap$ is added to the μ-terms.

References

Alaiwan, H.
[1984] Equivalence of infinite behavior of finite automata, Theor. Comput. Sci. 31, 297-306.

Alpern, B., Schneider, F.B.
[1985] Defining liveness, Inform. Process. Lett. 21, 181-185.
[1985a] Verifying temporal properties without using temporal logic, Rep. TR 85-723, Cornell Univ., Ithaca, N.Y.
[1987] Recognizing safety and liveness, Distributed Computing 2, 117-126.
[1987a] Proving boolean combinations of deterministic properties, Proc. 2nd IEEE Symp. on Logic in Comput. Sci., Ithaca, N.Y., 131-137.

Arnold, A.
[1983] Topological characterizations of infinite behaviours of transition systems, Proc. 10th ICALP (J. Diaz, ed.), LNCS 154, 28-38.
[1983a] Rational ω-languages are non-ambiguous, Theor. Comput. Sci. 26, 221-223.
[1985] A syntactic congruence for rational ω-languages, Theor. Comput. Sci. 39, 333-335.
[1987] Logical definability of fixed points, Rep. I-8730, Univ. de Bordeaux I.

Arnold, A., Niwinski, D.
[1987] Fixed point characterization of Büchi automata on infinite trees, Rep. I-8728, Univ. de Bordeaux.

Beauquier, D.
[1984] Bilimites de langages reconnaissables, Theor. Comput. Sci.33, 335-342.

Beauquier, D., Perrin, D.
[1985] Codeterministic automata on infinite words, Inform. Process. Lett. 20, 95-98.

Blanchard, F., Perrin, D.
[1980] Relèvement d'une mesure ergodique par un codage, Z. Wahrsch.theorie 54, 303-311.

Boasson, L., Nivat, M.
[1980] Adherences of languages, J. Comput. System Sci. 20, 285-309.

Braquelaire, J.P., Courcelle, B.
[1984] The solutions of two star-height problems for regular trees, Theor. Comput. Sci. 30, 205-239.

Büchi, J.R.
[1960] Weak second-order arithmetic and finite automata, Z. Math. Logik Grundlagen Math. 6, 66-92.
[1962] On a decision method in restricted second order arithmetic, in: "Logic, Methodology and Philosophy of Science. Proc. 1960 Intern. Congr." (E. Nagel et al., eds.), Stanford Univ. Press, pp. 1-11.
[1973] The monadic theory of ω_1, in: Decidable Theories II, Lecture Notes in Mathematics 328, Springer Verlag, Berlin, pp. 1-127.
[1977] Using determinacy to eliminate quantifiers, in: Fundamentals of Computation Theory (M. Karpinski, ed.), LNCS 56, 367-378.
[1983] State-strategies for games in $F_{\sigma\delta} \cap G_{\delta\sigma}$, J. Symb. Logic 48, 1171-1198.

Büchi, J.R., Landweber, L.H.
[1969] Solving sequential conditions by finite-state strategies, Trans. Amer. Math. Soc. 138, 295-311.

Büchi, J.R., Siefkes, D.
[1973] Axiomatization of the monadic second order theory of ω_1, in: Decidable Theories II, Springer Lect. Notes in Math. 328, 129-217.

Buszkowski, W.
[1980] Logical complexity of some classes of tree languages generated by multiple-tree-automata, Z. math. Logik u. Grundl. Math. 26, 41-49.

Choueka, Y.
[1974] Theories of automata on ω-tapes: A simplified approach, J. Comput. System Sci. 8, 117-141.
[1978] Finite automata, definable sets, and regular expressions over ω^n-tapes, J. Comput. System Sci. 17, 81-97.

Choueka, Y., Peleg, D.
[1983] A note on ω-regular languages, Bull. of the EATCS 21, 21-23.

Church, A.
[1963] Logic, arithmetic and automata, Proc. Int. Congr. Math.; Almqvist and Wiksells, Uppsala, 23-25.

Clarke, E.M., Grumberg, O., Kurshan, R.P.
[1987] A synthesis of two approaches for verifying finite state concurrent systems, manuscript, Carnegie-Mellon Univ., Pittburgh.

Cohen, R.S., Gold, A.Y.
[1977] Theory of ω-languages, J. Comput. System Sci. 15, 169-184 and 185-203.
[1978] ω-computations of deterministic pushdown machines, J. Comput. System Sci. 16, 275-300.
[1978a] ω-computations on Turing machines, Theor. Comput. Sci. 6 , 1-23.
[1980] On the complexity of ω-type Turing acceptors, Theor. Comput. Sci. 10, 149-272.

Colmerauer, A.
[1982] Prolog and infinite trees, in: "Logic Programming" (K.L. Clark, S.A. Tarnlund, eds.), Academic Press, New York.

Courcelle, B.
[1978] Frontiers of infinite trees, RAIRO Inform. Théor. 12, 319-337.
[1983] Fundamental properties of infinite trees, Theor. Comput. Sci. 25, 95-169.
[1988] The monadic second-order logic of graphs, I: Recognizable sets of finite graphs, Rep. I-8837, Univ. Bordeaux I.

Darondeau, P., Kott, L.
[1984] Towards a formal proof system for ω-rational expressions, Inform. Process. Letters 19, 173-177.
[1985] A formal proof system for infinitary rational expressions, in: "Automata on Infinite Words" (M. Nivat, D. Perrin, eds.), LNCS 192, 81-88.

Dauchet, M., Timmerman, E.
[1986] Continuous monoids and yields of infinite trees, RAIRO Theoretical Informatics and Applications 20, 251-274.

Davis, M.
[1964] Infinite games of perfect information, in: "Advances in game theory", Princeton Univ. Press, 85-101.

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

Eilenberg, S.
[1974/76] Automata, Languages and Machines, Vols. A and B, Academic Press, New York.

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

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

Emerson, E.A.
[1987] Uniform inevitability is tree automaton ineffable, Inform. Process. Lett. 24, 77-80.
[1988] Temporal and modal logic, in: Handbook of Theoretical Computer Science (to appear).

Emerson, E.A., Halpern, J.Y.
[1986] "Sometimes" and "Not Never" revisited: On branching time versus linear time, J. Assoc. Comput. Mach. 33, 151-178.

Emerson, E.A., Sistla, A.P.
[1984] Deciding full branching time logic, Inform. Contr. 61, 175-201.

Francez, N.
[1987] Fairness, Springer-Verlag, Berlin, Heidelberg, New York.

Gabbay, A., Pnueli, A., Shelah, S., Stavi, J.
[1980] On the temporal analysis of fairness, in: 7th ACM Symp. on Principles of Progr. Languages, Las Vegas, Nev., 163-173.

Gale,D., Stewart, F.M.
[1953] Infinite games with perfect information, in: "Contributions to the Theory of Games", Princeton Univ. Press, Princeton, N.J., 245-266.

Gecseg, F., Steinby, M.,
[1984] Tree Automata, Akademiai Kiado, Budapest.

Gire, F., Nivat, M.
[1984] Relations rationelles infinitaires, Calcolo 21, 91-125.

Guessarian, I., Niar, W.
[1988] An automaton characterization of fairness in SCCS, in: Proc. 5th STACS, Symp. on Theor. Aspects of Comput. Sci. (R. Cori, M. Wirsing, eds.), LNCS 294, 356-372.

Gurevich, Y.
[1985] Monadic second-order theories, in: Model-theoretic logics (J. Barwise, S. Feferman, eds.), Springer Verlag, Berlin, Heidelberg, New York, pp. 479-506.

Gurevich, Y., Harrington, L.A.
[1982] Automata, trees, and games, Proc. 14th ACM Symp. on the Theory of Computing, San Francisco, 60-65.

Gurevich, Y., Magidor, M., Shelah, S.
[1983] The monadic theory of ω_2, J. Symbolic Logic 48, 387-398.

Gurevich, Y., Shelah, S.
[1983] Rabin's uniformization problem, J. of Symbolic Logic 48, 1105-1119.
[1985] The decision problem for branching time logic, J. of Symbolic Logic 50, 668-681.

Hafer, T.
[1987] On the boolean closure of Büchi tree automaton definable sets of ω-trees, Aachener Inform. Ber. Nr. 87-16, RWTH Aachen.

Hafer, T., Thomas, W.
[1987] Computation tree logic CTL* and path quantifiers in the monadic theory of the binary tree, Proc. 14th ICALP (T. Ottmann, ed.), LNCS 267, 269-279.

Harel, D., Kozen, D., Parikh, R.
[1982] Process Logic: Expressiveness, decidability, completeness, J. Comput. System Sci. 25 (1982), 144-170.

Hayashi, T., Miyano, S.
[1985] Finite tree automata on infinite trees, Bull. of Inform. and Cybern. 21, 71-82.

Heilbrunner, S.
[1980] An algorithm for the solution of fixed-point equations for infinite words, RAIRO Inform. Théor. 14, 131-141.

Hoogeboom, H.J., Rozenberg, G.
[1986] Infinitary languages: Basic theory and applications to concurrent systems, in: Current Trends in Concurrency (J.W. de Bakker et alt., eds.), LNCS 224, 266-342.

Izumi, H., Inagaki, Y., Honda, N.
[1984] A complete axiom system for algebra of closed-regular expressions, in: Proc. 11th ICALP (J. Paredaens, ed.), LNCS 172, 260-269.

Jantzen, M.
[1986] Language theory of Petri nets, in: Petri Nets: Applications and Relationships to Other Models of Concurrency (W. Brauer et alt., eds.), LNCS 255, 397-412.

Jürgensen, H., Thierrin, G.
[1983] On ω-languages whose syntactic monoid is trivial, Intern. J. Comput. Inform. Syst. 12, 359-365.

Kaminski, M.
[1985] A classification of ω-regular languages, Theor. Comput. Sci. 36, 217-239.

Kamp, H.W.
[1968] Tense logic and the theory of linear order, Ph.D. Thesis, Univ. of California, Los Angeles.

Karpinski, M.
[1975] Almost deterministic ω-automata with output condition, Proc. Amer. Math. Soc. 53, 449-452.

Kurshan, R.P.
[1987] Testing containment of ω-regular languages, manuscript.

Ladner, R.E.
[1977] Application of model-theoretic games to discrete linear orders and finite automata, Inform. Contr. 33, 281-303.

Läuchli, H, Savioz, C.
[1987] Monadic second order definable relations on the binary tree, J. Symb. Logic 52, 219-226.

Lamport, L.
[1983] What good is temporal logic?, in: Information Processing (R.E.A. Mason, ed.), Elsevier Sci. Publ. (North-Holland), pp. 657-668.

Landweber, L.H
[1969] Decision problems for ω-automata, Math. Syst. Theory 3, 376-384.

Lichtenstein, O., Pnueli, A.
[1985] Checking that finite-state concurrent programs satisfy their specification, Proc. 12th ACM Symp. on Principles of Progr. Lang., New Orleans, La., 97-107.

Lichtenstein, O., Pnueli, A., Zuck, L.
[1985] The glory of the past, in: Logics of Programs (R. Parikh, ed.), LNCS 193, 196-218.

Lindsay, P.A.
[1986] Alternation and ω-type Turing acceptors, Theor. Comput. Sci. 43, 107-115.

Linna, M.
[1976] On ω-sets associated with context-free languages, Inform. Contr. 31, 272-293.
[1977] A decidability result for deterministic ω-context-free languages, Theor. Comput. Sci. 4 , 83-98.

Litovski, I., Timmerman, E.
[1987] On generators of rational ω-power languages, Theor. Comput. Sci. 53, 187-200.

Manna, Z., Pnueli, A.
[1981] Verification of concurrent programs: The temporal framework, in: The Correctness Problem in Computer Science (R.S. Boyer, J.S. Moore, eds.), Academic Press, London, 215-273.
[1987] Specification and verification of concurrent programs by ∀-automata, Proc. 14th ACM Symp. on Principles of Progr. Lang., Munich, 1-12.

Martin, D.A.
[1975] Borel determinacy, Ann. Math. 102, 363-371.

McNaughton, R.
[1966] Testing and generating infinite sequences by a finite automaton, Inform. Contr. 9, 521-530.

McNaughton, R., Papert, S.
[1971] Counter-Free Automata, MIT-Press, Cambridge, Mass.

Meyer, A.R.
[1975] Weak monadic second order theory of successor is not elementary recursive, in: Proc. Boston Univ. Logic Colloquium, Lecture Notes in Mathematics 453, Springer-Verlag, Berlin, Heidelberg, New York, 132-154.

Michel, M.
[1988] Complementation is more difficult with automata on infinite words, manuscript, CNET, Paris.

Milner, R.
[1984] A complete inference system for a class of regular behaviours, J. Comput. System Sci. 28, 439-466.

Miyano, S., Hayashi, T.
[1984] Alternating automata on ω-words, Theor. Comput. Sci. 32, 321-330.

Moriya,T.
[1987] Topological characterizations of infinite tree languages, Theor. Comput. Sci. 52, 165-171.

Moschovakis, Y.N.
[1980] Descriptive Set Theory, North-Holland, Amsterdam.

Mostowski, A.W.
[1977] Operations on ω-regular languages, in: Fundamentals of Computation Theory (M. Karpinski, ed.), LNCS 56, 135-141.
[1981] The complexity of automata and subtheories of monadic second-order arithmetics, in: Fundamentals of Computation Theory (F. Gecseg, ed.), LNCS 117, 453-466.
[1982] Determinancy of sinking automata on infinite trees and inequalities between various Rabin's pair indices, Inform. Process. Lett. 15, 159-163
[1984] Regular expressions for infinite trees and a standard form of automata, in: Computation Theory (A. Skowron, ed.), LNCS 208, 157-168.
[1987] Hierarchies of weak monadic formulas for two successors arithmetic, J. Inf. Process. Cybern. EIK 23, 509-515.

Mostowski, A.W., Skurczynski, J., Wagner, K.
[1985] Deterministic automata on infinite trees and the Borel hierarchy, Proc. Fourth Hung. Comput. Sci. Conf., 103-115.

Muchnik, A.A.
[1985] Games on infinite trees and automata with dead-end markers - a new proof of the decidability of the monadic theory of two successors, Semiotika 1985, 17-40 (in Russian).

Muller, D.E.
[1963] Infinite sequences and finite machines, in: Proc. 4th IEEE Ann. Symp. on Switching Circuit Theory and Logical Design, 3-16.

Muller, D.E., Saoudi, A., Schupp, P.E.
[1986] Alternating automata, the weak monadic theory of the tree, and its complexity, Proc. 13th ICALP (L. Kott, ed.), LNCS 226, 275-283.
[1988] Weak alternating automata give a simple explanation of why most temporal and dynamic logics are decidable in exponential time, in: Proc. 3rd IEEE Symp. on Logic in Computer Science, Edinburgh.

Muller, D.E., Schupp, P.E.
[1983] Groups, the theory of ends, and context-free languages, J. Comput. System Sci. 26, 295-310.
[1985] The theory of ends, pushdown automata, and second-order logic, Theor. Comput. Sci. 37, 51-75.
[1987] Alternating automata on infinite trees, Theor. Comput. Sci. 54, 267-276.

Nivat, M.
[1977] Mots infinis engendrés par une grammaire algébrique, RAIRO Inform. Théor. 11, 311-327.
[1978] Sur les ensembles de mots infinis engendrés par une grammaire algébrique, RAIRO Inform Théor. 12, 259-278.
[1989] Infinite words, infinite trees, infinite computations, in: "Foundations of Computer Science III.2" (J.W. de Bakker, J. van Leeuwen, Eds.), Math. Centre Tracts 109, 3-52.

Nivat, M., Perrin, D.
[1986] Ensembles reconnaissables de mots biinfinis, Canad. J. Math. 38, 513-537.

Niwinski, D.
[1984] Fixed-point characterization of context-free ∞-languages, Inform. Contr. 63, 247-276.
[1986] On fixed-point clones, Proc 13th ICALP (L. Kott, ed.), LNCS 226, 464-473.
[1986a] A note on indices of Rabin pair automata, unpublished manuscript.
[1988] Fixed points vs. infinite generation, in: Proc. 3rd IEEE Symp. on Logic in Computer Science, Edinburgh.

Parigot, M., Pelz, E.
[1985] A logical approach of Petri net languages, Theor. Comput. Sci. 39, 155-169

Park, D.
[1980] On the semantics of fair parallelism, in: Abstract Software Specification (D. Bjoerner, ed.), LNCS 86, 504-526.
[1981] Concurrency and automata on infinite sequences, in: Theoretical Computer Science (P. Deussen, ed.), LNCS 104, 167-183.

Pécuchet, J.P.
[1985] Automates boustrophédons et mots infinis, Theor. Comput. Sci. 35, 115-122.
[1986] On the complementation of Büchi automata, Theor. Comput. Sci. 47, 95-98.

[1986a] Etude syntaxique des parties reconnaissables de mots infinis, Proc. 13th ICALP (L. Kott, ed.), LNCS 226, 294-303.
[1987] Variétés de semigroupes et mots infinis, Proc. 3rd Symp. on Theor. Aspects of Comput. Sci. (G. Vidal-Naquet, ed.), LNCS 210, 180-191.

Pelz, E.
[1987] Infinitary languages of Petri nets and logical sentences, in: Advances in Petri Nets (G. Rozenberg, ed.), LNCS 266, 224-237.

Perrin, D.
[1984] Recent results on automata and inifinite words, in: Math. Found. of Comput. Sci. (M.P. Chytil, V. Koubek, eds.), LNCS 176, 134-148.
[1985] Variétés de semigroupes et mots infinis, C.R. Acad. Sci. Paris 295, 595-598.

Perrin, D., Pin, J.E.
[1986] First order logic and star-free sets, J. Comput. System Sci. 32,393-406.
[1986a] On the expressive power of temporal logic, LITP Rep. 86-17, Univ. Paris VII.

Perrin, D., Schupp, P.E.
[1986] Automata on the integers, recurrence distinguishability, and the equivalence and decidability of monadic theories, in: Proc. 1st IEEE Symp. on Logic in Computer Sci., Boston, 301-304.

Pnueli, A.
[1981] The temporal logic of concurrent programs, Theor. Comput. Sci. 13, 45-60.
[1986] Applications of temporal logic to the specification and verification of reactive systems: a survey of current trends, in: Current Trends in Concurrency (J.W. de Bakker et alt., eds.), LNCS 224, 510-584.

Priese, L., Rehrmann, R., Willecke-Klemme, U.
[1987] An introduction to the regular theory of fairness, Theor. Comput. Sci. 54, 139-163.

Rabin, M.O.
[1969] Decidability of second-order theories and automata on infinite trees, Trans. Amer. Math. Soc. 141, 1-35.
[1970] Weakly definable relations and special automata, in: "Mathematical Logic and Foundations of Set Theory" (Y. Bar-Hillel, ed.), pp.1-23, North-Holland, Amsterdam.
[1972] Automata on infinite objects and Church's problem, Amer. Math. Soc., Providence, R.I.

Redziejowski, R.R.
[1985] Infinite-word languages and continuous mappings, Theor. Comput. Sci. 43, 59-79.

Robinson, R.M.
[1958] Restricted set-theoretical definitions in arithmetic, Proc. Amer. Math. Soc. 9, 238-242.

Rosenstein, J.G.
[1982] Linear Orderings, Academic Press, New York.

Safra, S.
[1988] On the complexity of ω-automata, in: Proc. 29th Symp. on Found. of Comput. Sci., White Plains, N.Y.

Saoudi, A.
[1984] Infinitary tree languages recognized by ω-automata, Inform. Process. Lett. 18, 15-19.
[1986] Variétés d' automates descendants d' arbres infinis, Theor. Comput. Sci. 43, 315-335.

Schützenberger, M.P.
[1973] Sur les relations rationelles fonctionelles, in: Automata, Languages and Programming (M. Nivat, ed.), North-Holland, Amsterdam.

Seese, D.
[1972] Entscheidbarkeits- und Definierbarkeitsfragen der Theorie "netzartiger" Graphen I, Wiss. Z. d. Humboldt-Univ. zu Berlin, Math.-Naturwiss. R. 21, 513-517.
[1988] The structure of the models of decidable monadic theories of graphs, manuscript, Akad. d. Wiss. der DDR, Inst. f. Math., Berlin.

Semenov, A.L.
[1984] Decidability of monadic theories, in: Math. Found. of Comput. Sci. (M.P. Chytil, V. Koubek, eds.), LNCS 176 162-175.

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

Siefkes, D.
[1970] Decidable Theories I - Büchi's Monadic Second Order Successor Arithmetic, Lecture Notes in Mathematics 120, Springer-Verlag, Berlin, Heidelberg, New York.
[1975] The recursive sets in certain monadic second order fragments of arithmetic, Arch. math. Logik 17, 71-80.

Sistla, A.P.
[1985] On characterization of safety and liveness properties in temporal logic, Proc. 4th ACM Symp. on Principles of Distributed Computing, Minaki, Out., 39-48.

Sistla, A.P., Clarke, E.M.
[1985] The complexity of propositional linear time logics, J. Assoc. Comput. Mach. 32, 733-749.

Sistla, A.P., Vardi, M.Y., Wolper, P.
[1987] The complementation problem for Büchi automata with applications to temporal logic, Theor. Comput. Sci. 49, 217-237.

Staiger, L.
[1983] Finite-state ω-languages, J. Comput. System Sci. 27, 434-448.
[1986] Hierarchies of recursive ω-languages, J. Inf. Process. Cybern. EIK 22, 219-241.
[1987] Sequential mappings of ω-Languages, RAIRO Theor. Inform. and Applications 21, 147-173.
[1987a] Research in the theory of ω-languages, J. Inf. Process. Cybern. EIK 23, 415-439.

Staiger, L., Wagner, K.
[1974] Automatentheoretische und automatenfreie Charakterisierungen topologischer Klassen regulärer Folgenmengen, J. Inf. Process. Cybern. EIK 10, 379-392.

Stockmeyer, L.J., Meyer, A.R.
[1973] Word problems requiring exponential time: preliminary report, in: Proc. 5th ACM Symp. on the Theory of Computing, 1-9.

Streett, R.S.
[1982] Propositional dynamic logic of looping and converse, Inform. Contr. 54, 121-141.

Streett, R.S., Emerson, E.A.,
[1984] The propositional Mu-calculus is elementary, Proc. 11th ICALP (J. Paredaens, ed.), LNCS 172, 465-472.

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

Takahashi, M.
[1986] The greatest fixed-points and rational omega-tree languages, Theor. Comput. Sci. 44, 259-274.
[1987] Brzozowski hierarchy of ω-languages, Theor. Comput. Sci. 49, 1-12.

Takahashi, M., Yamasaki, H.
[1983] A note on ω-regular languages, Theor. Comput. Sci. 23, 214-225.

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

Thomas, W.
[1979] Star-free regular sets of ω-sequences, Inform. Contr. 42, 148-156.
[1980] On the bounded monadic theory of well-ordered structures, J. Symbolic Logic 45, 334-338.
[1981] A combinatorial approach to the theory of ω-automata, Inform. Contr. 48,261-283.
[1982] Classifying regular events in symbolic logic, J. Comput. System Sci. 25, 360-376.
[1983] A hierarchy of sets of infinite trees, in: Theoretical Computer Science (A.B. Cremers, H.P. Kriegel, eds.), LNCS 145, 335-342.
[1986] On frontiers of regular trees, RAIRO Theor. Informatics and Applications 20, 371-381.
[1987] On chain logic, path logic, and first-order logic over infinite trees, Proc. 2nd IEEE Symp. on Logic in Comput. Sci., Ithaca, N.Y., 245-256.

Trachtenbrot, B.A.
[1962] Finite automata and the logic of one-place predicates, Siberian Math. J. 3, 103-131 (Russian), Engl. transl. in: AMS Transl. 59 (1966), 23-55.

Trachtenbrot, B.A., Barzdin, Y.M.
[1973] Finite Automata, North-Holland, Amsterdam.

Valk, R.
[1983] Infinite behaviour of Petri nets, Theor. Comput. Sci. 25, 311-341.

Vardi, M.Y., Stockmeyer, L.J.
[1985] Improved upper and lower bounds for modal logics of programs: preliminary report, in: Proc. 17th ACM Symp. on the Theory of Computing, Providence, R.I., 240-251.

Vardi., M.Y., Wolper, P.
[1986] Automata-theoretic techniques for modal logics of programs, J. Comput. System Sci. 32, Boston, 183-221.
[1986a] An automata theoretic approach to automatic program verification, Proc. 1st IEEE Symp. on Logic in Comput. Sci., 332-334.
[1988] Reasoning about infinite computation paths, manuscript.

Wagner, K.
[1976] Eine Axiomatisierung der Theorie der regulären Folgenmengen, J. Inf. Process. Cybern. EIK 12, 337-354.
[1979] On ω-regular sets, Inform. Contr. 43, 123-177.

Wisniewski, K.
[1987] A generalization of finite automata, Fund. Inform. 10, 415-436.

Wojciechowski, J.
[1985] Finite automata on transfinite sequences and regular expressions, Fund. Inform. 8, 379-396.

Wolper, P.
[1983] Temporal logic can be more expressive, Inform. Contr. 56, 72-99.

Wolper, P., Vardi, M.Y., Sistla, A.P.
[1983] Reasoning about infinite computation paths, Proc. 24th IEEE Symp. on Found. of Comput. Sci., Tucson, Arizona, 185-194.

Yamasaki, H., Takahashi, M., Kobayashi, K.
[1986] Characterization of ω-regular languages by monadic second-order formulas, Theor. Comput. Sci. 46, 91-99.

Aachener Informatik-Berichte
Please write to: Informatik-Bibliothek, RWTH Aachen, Ahornstr. 55, D 5100 Aachen

87-1 Jahresbericht 1986

87-2 D. de Frutos Escrig/K. Indermark: Equivalence relations of non-deterministic Ianov-schemes

87-3 M. Nagl: A Software Development Environment Based on Graph Technology

87-4 C. Lewerentz/M. Nagl/B. Westfechtel: On Integration Mechanisms within a Graph-based Software Development Environment

87-5 R. Rinn: Über Eingabeanomalien bei verschiedenen Inferenzmodellen

87-6 W. Damm/G. Döhmen: Specifying Distributed Computer Architectures in AADL*

87-7 G. Engels/C. Lewerentz/W. Schäfer: Graph Grammar Engineering: A Software Specification Method

87-8 M. Nagl: Set Theoretic Approaches to Graph Grammars

87-9 C. Lewerentz/A. Schürr: Experiences with a Database System for Software Documents

87-10 H. Klaeren/K. Indermark: A New Implementation Technique for Recursive Function Definitions

87-11 R. Loogen: Design of a parallel programmable graph reduction machine with distributed memory

87-12 J. Börstler/U. Möncke/R. Wilhelm: Table Compression for Tree Automata

87-13 C. Lewerentz: Extended Programming in the Large in a Software Development Environment

87-14 G. Engels/Th. Janning/W. Schäfer: A Highly Integrated Tool Set for Program Development Support

87-15 R. Loogen/U. Goltz: A non-interleaving semantic model for nondeterministic concurrent processes