MR0923334 (89b:03063) 03D05 68Q45

Staiger, Ludwig (DDR-ADW-I)

Research in the theory of ω-languages. (English. German, Russian summary)

Mathematical aspects of informatics (Mägdesprung, 1986).
J. Inform. Process. Cybernet. **23** (1987), *no.* 8-9, 415–439.

A survey on several topics in the theory of ω-languages (sets of one-sided infinite strings) is given with a definitive bibliography (117 titles). The topics chosen by the author are as follows: (1) Acceptance types and topology. The strong relationship between acceptance types and topological subclasses (in the Borel hierarchy) of the ω-languages is established for classes of accepting devices such as Turing machines, finite automata and pushdown automata. (2) Operations on ω-languages. A number of families of ω-languages are represented by the corresponding families of languages (sets of finite strings) using (three types of) limit operations: concatenation, ω-closure and union. (3) Complexity of acceptance. Complexity of acceptance is investigated in connection with topological complexity. (4) Finite-state ω-languages. Finite-state ω-languages (those having finite syntactic monoids) are not necessarily ω-regular (i.e., accepted by a finite automaton), in contrast to the finite language case. The author investigates the conditions under which finite state ω-languages are ω-regular. Some open problems are presented. (5) Systems of equations and formal proof systems for ω-regular expressions. Though the minimal solution principle for linear equations is not applicable in the ω-language case, it is possible to construct a proof system for ω-regular expressions based on Salomaa's system for regular expressions. *Hideki Yamasaki*

J. Inf. Process. Cybern. EIK 23 (1987) 8/9, 415–439
(formerly: Elektron. Inf.verarb. Kybern. EIK)

Research in the Theory of ω-languages[1]

By *Ludwig Staiger*

Abstract: A survey on several results in the theory of ω-languages emphasizing topological and algebraic methods is given.

The theory of ω-languages was initiated in the early sixties by the papers of *Büchi* [7] and *Trakhtenbrot* [99]. In both cases the emerging point was the question on the decidability of the monadic second-order arithmetic of one successor function. In this connection the regular (i.e. definable by finite automata) ω-languages appeared as sets of predicates definable in this theory. (The paper [8] explains the very natural appearance of finite automata in monadic second-order formulas.)

As Büchi's paper [7] proved his famous decidability result, it attracted more attention, and subsequently the acceptance of ω-languages by finite automata was investigated in more detail. We mention here McNaughton's lemma [54] who proved the equivalence of deterministic and nondeterministic automata also in the ω-case, and Landweber's paper [46] showing a strong correspondence between acceptance types and topology.

These early days are surveyed in the book by *Trakhenbrot* and *Barzdin* [100]. At the same time the text by *Siefkes* [73] appeared dealing more with aspects of monadic second-order logic.

Recently the complete work, including the results on ω-languages, of the late *J. Richard Büchi* was reviewed in [74].

In the last time the theory of ω-languages became popular as it might be seen from the list of references. Among them there are several papers [109, 63, 67, 88] and books [100, 48, 61] surveying in a more or less complete manner some part of this theory.

In the present paper I try not only to deal with some recent developments in the theory of ω-languages which are not (fully) covered by the above papers, but also, going back to the roots in the seventies and sixties, to reveal several connections between acceptance and internal (topological or state-)structure of ω-languages. In these discussions, I shall concentrate on those subjects which I know best:

I confine to ω-string languages not considering ω-trees or two-sided infinite strings. This confinement is also extended to the list of references where I included mainly papers dealing with ω-string languages. Moreover, except for a small number of thesises [33, 36, 51, 66, 76, 78, 105], I included no technical reports, but only papers published or accepted for publication. (Other bibliographies on work on ω-languages were compiled by *H. Jürgensen* in 1980 and by *M. Takahashi, H. Yamasaki* and *T. Hayashi* recently.)

---

[1]) Extended version of a lecture given at the Workshop 'Mathematical Aspects of Informatics', Mägdesprung (GDR), April 1986.

Among the topics which are possible I have chosen the following five:

- acceptance types and topology
- operations for ω-languages
- complexity of acceptance
- finite-state ω-languages
- systems of equations and formal proof systems for ω-regular expressions.

After some necessary preliminaries we start with the first topic in Section 2.

It is already known from Landweber's paper [46] that for finite automata the classes of ω-languages accepted by several acceptance types can be characterized topologically. In a subsequent paper [93] we completed Landweber's investigations in two directions: we extended them to the nondeterministic case and introduced the sixth acceptance type ($(E, =)$ below) which was overlooked by *Landweber*. (Section 5 reveals that this type possesses remarkable properties).

In Section 2 we investigate extensively the strong relationship between acceptance types and topological characterization for the classes of ω-languages accepted by Turing machines, finite automata and pushdown automata. This joint treatment of the deterministic and nondeterministic cases of several automata types allows us to present common features and differences in a comprehensive manner unifying several results and methods scattered in the literature.

The next section (Operations for ω-languages) shows that the families of ω-languages defined previously by acceptance are obtained from families of languages by several operations. This correspondence between ω-languages and languages is considered in more detail. Here we also derive the complete Init-lemma — a thorough characterization of the initial word languages of the above-mentioned classes of ω-languages.

The part of complexity of acceptance tries to introduce the reader into the extensive work of *Wagner* [105, 109] on complexity of acceptance and its connections to topological complexity.

The phenomenon that, unlike the language case, finite-state ω-languages (i.e. those having a finite syntactic monoid) are not necessarily regular (i.e. accepted by a finite automaton) is the topic of the fifth section. Here we investigate conditions under which finite-state ω-languages are regular, and we consider the syntactic monoid and the associated automaton of a finite-state ω-language. This latter concept allows a partial approach to the minimization problem for ω-automata.

In the sixth section we consider work on equation and axiom systems for regular ω-languages done in the seventies.

Some concluding remarks are added finally in order to hint for some applications of the theory of ω-languages.

This paper as well as its list of references tend not to be complete. We have not surveyed a lot of material and certainly not quoted many papers which deserve attention.

Among the topics not treated here we mention the following ones:

- the work of *Cohen* and *Gold* [15, 16] and *Nivat* [58, 59] on the generation of context-free ω-languages by grammers
- several results on algebraic aspects of the acceptance of ω-languages by finite automata obtained recently in France which were surveyed by *Perrin* in [63]
- various new research results which are contained in the texts of the Ecole de printemps d'Informatique Théorique 1984 (edited by *M. Nivat* and *D. Perrin*) [61]
- the collection [88] of up to now achieved and some new results on ω-languages accepted by Turing machines and on their connection to the arithmetical hierarchy

— the results on alternating accepting devices by *Lindsay* [50, 115], and *Miyano/Hayashi* [55]
— the results by *Kobayashi, Takahashi* and *Yamasaki* ([43] and [112]) on the characterization of classes of regular ω-languages by logical formulas.

This last topic is interesting in that it reverses the original direction from which the study of ω-languages emerged: as a tool for solving logical problems.

1. Preliminaries

The set $\{0, 1, 2, \ldots\}$ of natural numbers is denoted by N, and for a finite alphabet X^* (X^ω) denotes the set of finite words (infinite sequences) on X. For a word $w \in X^*$ and a string $b \in X^* \cup X^\omega$ let wb be their concatenation. This in an obvious way defines a product $W \cdot B$ as well as an n-th power W^n ($n \geqq 1$) of sets $W \subseteqq X^*$ and $B \subseteqq X^* \cup X^\omega$.

We introduce into $X^* \cup X^\omega$ a partial ordering

$$w \sqsubseteq b :\Leftrightarrow w \cdot b' = b \text{ for some } b' \in X^* \cup X^\omega .$$

By

$$A(b) := \{w : w \in X^* \text{ and } w \sqsubseteq b\} \quad \text{and}$$

$$A(B) := \bigcup_{b \in B} A(b)$$

we denote the set of initial words (prefixes) of $b \in X^* \cup X^\omega$ and $B \subseteqq X^* \cup X^\omega$ resp.

For a word w its length is $|w|$, and e denotes the empty word in X^*.

We extend the operators * and $^\omega$ to arbitrary subsets $W \subseteqq X^*$ in the usual way: $W^* := \bigcup_{n \in N} W^n$ where $W^0 := \{e\}$, and

$$W^\omega := \{w_0 \cdot w_1 \cdot \ldots \cdot w_i \cdot \ldots : i \in N \text{ and } w_i \in W \setminus \{e\}\}$$

is the set of (infinite) sequences in X^ω formed by concatenating members of W.

We will refer to subsets of X^* (X^ω) as languages (ω-languages).

In X^ω we will consider the (natural) product topology which is defined by the basis $\{w \cdot X^\omega : w \in X^*\}$ or otherwise by the closure operator C, where for $F \subseteqq X^\omega$ its closure

$$C(F) := \{\beta : A(\beta) \subseteqq A(F)\}$$

is the smallest closed set containing the set F (cf. [48]).

As usual we define the Borel hierarchy in X^ω: $G_\delta(F_\sigma)$ is the class of denumerable intersections (unions) of open (closed) sets. Then $G_{\delta\sigma}$, $G_{\delta\sigma\delta}$, ... and $F_{\sigma\delta}$, $F_{\sigma\delta\sigma}$, ... are defined in the usual manner.

Closed sets and G_δ-sets in X^ω can be characterized by languages in X^* using the following limits.

Definition. For $W \subseteqq X^*$ we will refer to

$$\text{ls } W := \{\beta : \beta \in X^\omega \text{ and } A(\beta) \subseteqq A(W)\}$$

as the *limit* (in [11, 12]: adherence) of the language (cf. [76], [48]), and to

$$W^\delta := \{\beta : \beta \in X^\omega \text{ and } A(\beta) \cap W \text{ infinite}\}$$

as the *δ-limit* of the languages (cf. [22]).

A limit similar to ls was introduced by *Elgot* [24]:

$$\lim W := \{\beta : \beta \in X^\omega \text{ and } A(\beta) \subseteqq W\} .$$

The three limits are related in the following manner

$$\lim W \subseteq W^{\delta} \subseteq \mathrm{ls}\, W = \lim \boldsymbol{A}(W) = \boldsymbol{A}(W)^{\delta} = \mathrm{ls}\, \boldsymbol{A}(W)\,. \tag{1.1}$$

For $F \subseteq X^{\omega}$ the closure $\boldsymbol{C}(F)$ equals (cf. [76, 48])

$$\boldsymbol{C}(F) = \mathrm{ls}\, \boldsymbol{A}(F) = \boldsymbol{A}(F)^{\delta} = \lim \boldsymbol{A}(F)\,. \tag{1.2}$$

Moreover, we have

Proposition 1.1 ([22]). *A subset $F \subseteq X^{\omega}$ is a $\boldsymbol{G}_{\delta}$-set iff there is a $W \subseteq X^*$ such that $F = W^{\delta}$.*

With (1.1) we obtain (cf. [76])

Corollary 1.2. *A subset $F \subseteq X^{\omega}$ is closed iff there is a $W \subseteq X^*$ such that $F = \mathrm{ls}\, W$ (iff there is a $W' \subseteq X^*$ such that $F = \lim W'$).*

The reader is warned not to confuse our **lim** due to *Elgot* [24] with the lim-notation e.g. in [14, 17, 18, 51, 52, 53] where the δ-limit is denoted by lim. We prefer the notation W^{δ} for the reason given in Proposition 1.1 above.

Furthermore we introduce the following notion.

Definition. A *generalized sequential machine* (or short: gsm) is a 6-tuple $\mathfrak{A} = (X, Y, Z, f, g, z_0)$ where

X and Y are the finite input and output alphabets resp.,
Z is the finite set of states,
$z_0 \in Z$ is the initial state,
$f: Z \times X \to Z$ is the next state function, and
$g: Z \times X \to Y^*$ is the output function.

As usual f ang g may be extended to $Z \times X^*$ via

$$f(z, e) := z\,, \quad f(z, w \cdot v) := f(f(z, w), v)\,, \quad \text{and}$$
$$g(z, e) := e\,, \quad g(z, w \cdot v) := g(z, w) \cdot g(f(z, w), v)\,.$$

A generalized sequential machine $\mathfrak{A}$ defines a mapping $\varphi_{\mathfrak{A}}: X^* \to Y^*$ by $\varphi_{\mathfrak{A}}(w) := g(z_0, w)$. This *gsm-mapping* $\varphi_{\mathfrak{A}}$ may be extended to a gsm-mapping $\overline{\varphi_{\mathfrak{A}}}: X^{\omega} \to Y^* \cup Y^{\omega}$ via the relation $\boldsymbol{A}(\overline{\varphi_{\mathfrak{A}}}(\beta)) = \boldsymbol{A}(\overline{\varphi_{\mathfrak{A}}}(\boldsymbol{A}(\beta)))$ for $\beta \in X^{\omega}$ (cf. [48, 91]).

We call a gsm-mapping $\varphi_{\mathfrak{A}}$ *totally unbounded* provided its extension satisfies $\overline{\varphi_{\mathfrak{A}}}(X^{\omega}) \subseteq Y^{\omega}$. Totally unbounded gsm-mappings and their behaviour on ω-languages were extensively studied in the book [48].

A particular important case of gsm-mappings is the *projection* pr: $X^* \to X^*$ which erases every other letter, i.e. $\mathrm{pr}(e) := e$, and

$$\mathrm{pr}(w \cdot x) := \begin{cases} \mathrm{pr}(w) \cdot x\,, & \text{if } |w| \text{ is even}\,, \\ \mathrm{pr}(w)\,, & \text{if } |w| \text{ is odd}\,, \end{cases} \quad \text{for} \quad w \in X^* \quad \text{and } x \in X\,.$$

A parallel version of projection, mapping $(X \times X)^*$ to X^* in connection with properties of ω-languages has been considered in [86]. It should be noted that the projection $\overline{\mathrm{pr}}$, and in general all totally unbounded gsm-mappings $\overline{\varphi_{\mathfrak{A}}}$ are continuous mappings of the space X^{ω} to Y^{ω}.

Since our study of families of ω-languages in the subsequent sections tries to relate these families to families of languages, we assume the reader to be familiar with the basic facts in the theory of languages and automata as presented e.g. in the textbooks [6, 23, 26, 70, 71, 111]. In particular, we assume the reader to be familiar with the families R, DCF, CF, REC, and RE of regular, deterministic context-free, context-free, recursive and recursively enumerable languages resp.

2. Acceptance types and topology

Classes of ω-languages were mainly introduced via devices accepting ω-languages under various acceptance types. The first such type (termed $(\boldsymbol{U}, \subseteqq)$ below) was introduced by *Büchi* [7] in connection with nondeterministic finite automata, another (termed $(\boldsymbol{U}, =)$ below) was introduced by *Müller* [56] in connection with deterministic finite automata. It was soon observed that these and the subsequently introduced acceptance types correspond to low classes of the Borel hierarchy (cf. [46] and [93]; some of the results of the latter paper were rederived in the note [96].)

In this section we consider results relating the classes of ω-languages accepted by finite automata, pushdown automata and Turing machines to the classes of the Borel hierarchy.

In [110, 95] we have shown that the classes of ω-languages accepted by Turing machines are well-described by means of the arithmetical hierarchy (cf. the standard text by *Rogers* [68]). These results are also surveyed in [88], so we will not recall them here in full detail.

Our intention is to stress more the topological details where we are based on the following completion $\mathfrak{B}$ of the Borel hierarchy: to the usual $\boldsymbol{G}_\alpha$- and $\boldsymbol{F}_{\tilde\alpha}$-classes we add the intersections $\boldsymbol{G}_\alpha \cap \boldsymbol{F}_{\tilde\alpha}$ and the Boolean closures $\mathcal{B}(\boldsymbol{G}_\alpha)$. As it is well-known (cf. [45]) the proper inclusions shown in Fig. 1 hold true.

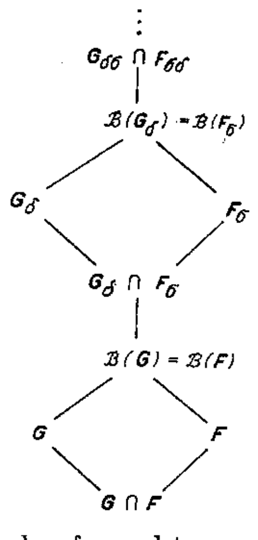

Fig. 1

For the sake of completeness we add the hierarchy $\mathfrak{P}$ of projective sets (cf. [45; § 38]). It was shown by *Arnold* [2] (cf. also [86]) that even for infinite-state accepting devices the topological complexity of the accepted ω-languages does not reach beyond the first projective class.

So, for our purposes, it suffices to consider only the classes

$$\boldsymbol{P}_0 := \{F : F \subseteqq X^\omega \text{ and } F \text{ a Borel set}\}, \quad \text{and}$$
$$\boldsymbol{P}_1 := \{\overline{\mathrm{pr}}\, F : F \in \boldsymbol{P}_0\} = \{\overline{\mathrm{pr}}\, F : F \in \boldsymbol{G}_\delta\},$$

and we need not go into further details concerning the hierarchy $\mathfrak{P}$. (The situation is a little bit different in the case of acceptance by alternating Turing machines as it was shown by *Lindsay* [50].)

Definition. We say that a hierarchy $\mathfrak{H}$ of classes of ω-languages is *topologically based* provided there is a mapping $h\colon \mathfrak{H} \to \mathfrak{B} \cup \mathfrak{P}$ satisfying the following three conditions:

(i) $h(H) \in \mathfrak{B} \cup \mathfrak{P}$ for all $H \in \mathfrak{H}$,
(ii) $H \subseteq h(H)$ for all $H \in \mathfrak{H}$, and
(iii) if $H \subsetneqq H'$ for $H, H' \in \mathfrak{H}$, then $H \setminus h(H') \neq \emptyset$.

Condition (ii) guarantees that all ω-languages in one class H have a topological complexity bounded by $h(H)$, whereas condition (iii) states that the distinctness of classes $H, H' \in \mathfrak{H}$ can be explained already by topological means.

Topologically based hierarchies of classes of ω-languages are presented below where we generally choose $h(H)$ to be the smallest class in our completed hierarchy $\mathfrak{B} \cup \mathfrak{P}$ which contains the class $h \in \mathfrak{H}$. Hierarchies of classes of ω-languages which are not topologically based are presented in [88].

Now let us consider in more detail the accepting process for ω-languages. Since finite automata as well as pushdown automata may be viewed as special cases of Turing machines, we start with this more general case of accepting devices.

We consider Turing machines $\mathfrak{M} = (X, Z, z_0, R)$ with an input tape on which the read-only-head moves only to right, n working tapes, X as its alphabet (for all tapes), Z the finite set of internal states, z_0 the initial state, and the relation

$$R \subseteq (Z \times X^{n+1}) \times \bigl(Z \times \{\text{stop, move}\} \times (X \cup \{\text{left, right}\})^n\bigr)$$

defining the next configuration.

If $n = 0$, then $\mathfrak{M}$ is called a finite automaton, and a pushdown automaton may be conceived as a Turing machine with only one working tape on which the head has to erase the written symbols when moving to the left.

Unless stated otherwise, in the sequel we shall assume that our accepting devices be fully defined, i.e. the transition relation R is to contain for every situation $(z, x_0, x_1, \dots, x_n)$ in $Z \times X^{n+1}$ at least one (exactly one, if the device is deterministic) move $(z, x_0, x_1, \dots, x_n; z', y_0, y_1, \dots, y_n)$.

Let the input of the Turing machine be some sequence $\beta \in X^\omega$. We accept β if the sequence of states the Turing machine runs through in its (some of its, if the machine is nondeterministic) computation with input β fulfils a certain condition. Here we consider the types of acceptance known from the classical paper by *Landweber* [46] and the additional type $(\boldsymbol{E}, =)$ introduced in [93] completing the acceptance considerations in a natural way.

To every Turing machine $\mathfrak{M}$ and to every input sequence β we assign the set $\Phi_{\mathfrak{M}}(\beta)$ of all sequences $\eta \in Z^\omega$ the Turing machine runs through during a computation with input β. Clearly, if $\mathfrak{M}$ is determinstic, then $\Phi_{\mathfrak{M}}(\beta) = \{\eta\}$. For a sequence $\eta \in Z^\omega$ of states let $\boldsymbol{E}(\eta)$ $\bigl(\boldsymbol{U}(\eta)\bigr)$ be the set of states occurring in η (occuring in η infinitely often).

Let $\mathfrak{Z}$ be a system of subsets of Z, and let $\alpha \in \{\boldsymbol{E}, \boldsymbol{U}\}$ and σ be one of relations $=$, $\subseteq$, $\subsetneqq$. Then

$$T_\sigma^\alpha(\mathfrak{M}, \mathfrak{Z}) := \{\beta : \beta \in X^\omega \wedge \exists Z' \; \exists \eta (Z' \in \mathfrak{Z} \wedge \eta \in \Phi_{\mathfrak{M}}(\beta) \wedge \alpha(\eta) \; \sigma \; Z'\}$$

is the set of sequences accepted by $(\mathfrak{M}, \mathfrak{Z})$ with respect to the type (α, σ).

In this manner six different types of acceptance are defined which will be denoted by the following suggesting abbreviations $(\boldsymbol{E}, =)$, $(\boldsymbol{E}, \subseteq)$, $(\boldsymbol{E}, \subsetneqq)$ ($\boldsymbol{E}$ stands for every-where) and $(\boldsymbol{U}, =)$, $(\boldsymbol{U}, \subseteqq)$, $(\boldsymbol{U}, \not\subseteqq)$ ($\boldsymbol{U}$ resembles Markwald's infinity quantifier, cf. [68; § 14.8]).

To illustrate the acceptance types just introduced, we present the ω-languages accepted by the finite automata $\mathfrak{A}_1$ and $\mathfrak{A}_2$ given in Figs. 2 and 3 according to our six types.

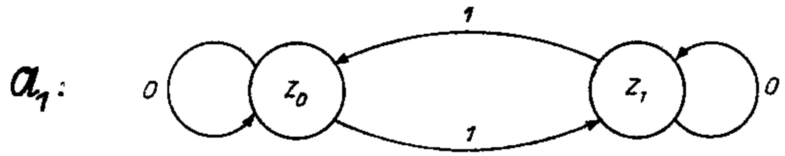

Fig. 2

Example 2.1. We consider the automaton $\mathfrak{A}_1$ given in Fig. 2 and present the sets accepted by $\mathfrak{A}_1$ and several final families as follows ($X := \{0, 1\}$).

	$\{\{z_0\}, \{z_1\}\}$	$\{\{z_0, z_1\}\}$	$\{\{z_0\}, \{z_0, z_1\}\}$
$(\boldsymbol{E}, \subseteqq)$	$0^\omega \cup 1 \cdot 0^\omega$	X^ω	X^ω
$(\boldsymbol{E}, \not\subseteqq)$	$X \cdot 0^* \cdot 1 \cdot X^\omega$	$\emptyset$	$X \cdot 0^* \cdot 1 \cdot X^\omega$
$(\boldsymbol{E}, =)$	$0^\omega \cup 1 \cdot 0^\omega$	$X^\omega \setminus (0^\omega \cup 1 \cdot 0^\omega)$	$X^\omega \setminus 1 \cdot 0^\omega$
$(\boldsymbol{U}, \subseteqq)$	$X^* \cdot 0^\omega$	X^ω	X^ω
$(\boldsymbol{U}, \not\subseteqq)$	$(0^* \cdot 1)^\omega$	$\emptyset$	$(0^* \cdot 1)^\omega$
$(\boldsymbol{U}, =)$	$X^* \cdot 0^\omega$	$(0^* \cdot 1)^\omega$	$((0^* \cdot 1)^2)^* \cdot 0^\omega \cup (0^* \cdot 1)^\omega$

Example 2.2. We consider the automaton $\mathfrak{A}_2$ given in Fig. 3 and present the sets accepted by $\mathfrak{A}_2$ and several final families (again $X := \{0, 1\}$).

	$\{\{z_0\}, \{z_1\}\}$	$\{\{z_0, z_1\}\}$	$\{\{z_0\}, \{z_0, z_1\}\}$
$(\boldsymbol{E}, \subseteqq)$	0^ω	X^ω	X^ω
$(\boldsymbol{E}, \not\subseteqq)$	$0^* \cdot 1 \cdot X^\omega$	$\emptyset$	$0^* \cdot 1 \cdot X^\omega$
$(\boldsymbol{E}, =)$	0^ω	$0^* \cdot 1 \cdot X^\omega$	X^ω
$(\boldsymbol{U}, \subseteqq)$	$X^* \cdot 0^\omega$	X^ω	X^ω
$(\boldsymbol{U}, \not\subseteqq)$	$(0^* \cdot 1)^\omega$	$\emptyset$	$(0^* \cdot 1)^\omega$
$(\boldsymbol{U}, =)$	$X^* \cdot 0^\omega$	$(0^* \cdot 1)^\omega$	X^ω

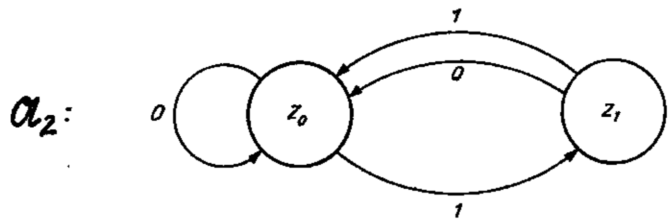

Fig. 3

In connection with the above example we mention the following property of regular ω-languages:

Since any one-state automaton accepts either X^ω or $\emptyset$, both of the above considered automata $\mathfrak{A}_1$ qnd $\mathfrak{A}_2$ are minimal automata accepting the ω-language $X^* \cdot 0^\omega$. Moreover, $\mathfrak{A}_1$ and $\mathfrak{A}_2$ are nonisomorphic automata.

Thus, in contrast to the case of regular languages, a regular ω-language may possess several (nonisomorphic) deterministic minimal accepting automata.

We shall return to this phenomenon in Section 5.

If (α, σ) is any one of the above defined types, the class of languages accepted by deterministic (nondeterministic) Turing machines according to (α, σ) is denoted by $\mathrm{DT1}(\alpha, \sigma)$ $(\mathrm{NT1}(\alpha, \sigma))$ (see [95, 110], [88]). In [95] and [110] we denoted by DT2 and NT2 a kind of acceptance where $\Phi_{\mathfrak{M}}(\beta)$ was defined in a more complicated way, thereby yielding in some cases larger classes of accepted ω-languages. However, $\mathrm{NT2}(\alpha, \sigma) \subseteq \mathrm{NT1}(\boldsymbol{U}, =)$ for all pairs (α, σ).

We should add some remarks on the compatibility of our results with the ones of *Cohen* and *Gold* [18, 19]. The classes $\mathrm{DT1}(\alpha, \sigma)$ coincide with the corresponding classes in [18] (ecxept for $\mathrm{DT1}(\boldsymbol{E}, =)$ which was not considered there), whereas in the nondeterministic case *Cohen* and *Gold* require for $\eta \in \Phi_{\mathfrak{M}}(\beta)$ that η be the state sequence of a complete nonoscillating run. This involves a further $\forall\exists$-condition resulting in the identity $\mathrm{NT}_{\mathrm{CG}}(\alpha, \sigma) = \mathrm{NT1}(\boldsymbol{U}, =)$ for all pairs (α, σ).

Analogously, $\mathrm{DFA}(\alpha, \sigma)$ $(\mathrm{NFA}(\alpha, \sigma))$ denotes the class of ω-languages accepted by deterministic (nondeterministic) according to the type (α, σ), and by $\mathrm{DPDA}(\alpha, \sigma)$ $(\mathrm{NPDA}(\alpha, \sigma))$ we denote the class of ω-languages accepted by deterministic (nondeterministic) pushdown automata.

Remark. *Linna* [51, 52] and *Cohen* and *Gold* in [15, 16] require the additional acceptance condition that the pushdown automaton read the whole input tape. Similar to the above remark this again involves an additional $\forall\exists$-condition which, however, can be eliminated in the cases of $(\boldsymbol{U}, \nsubseteq)$- and $(\boldsymbol{U}, =)$-acceptance (see [51, 52, 53]). Thus, it is to note that our classes $\mathrm{NPDA}(\alpha, \sigma)$ do not coincide with the corresponding ones from [15, 16] in the cases $(\boldsymbol{E}, \sigma)$ (σ arbitrary) and $(\boldsymbol{U}, \subseteq)$.

Define

$$\mathsf{REC}_\omega := \bigcup_{\text{all } (\alpha, \sigma)} \mathrm{NT1}(\alpha, \sigma)\,,$$

to be the class of ω-languages accepted by Turing machines (recursive ω-languages),

$$\mathsf{R}_\omega := \bigcup_{\text{all } (\alpha, \sigma)} \mathrm{NFA}(\alpha, \sigma)$$

to be the class of all ω-languages accepted by finite automata (regular ω-languages),

$$\mathsf{CF}_\omega := \bigcup_{\text{all } (\alpha, \sigma)} \mathrm{NPDA}(\alpha, \sigma) \quad \text{and}$$

$$\mathsf{DCF}_\omega := \bigcup_{\text{all } (\alpha, \sigma)} \mathrm{DPDA}(\alpha, \sigma)$$

to be the classes of context-free and deterministic context-free ω-languages resp.

Now let us have a closer look to the interconnections between the acceptance types. From the very definition one immediately gets

$$T^\alpha_{\subseteq}(\mathfrak{M}, \mathfrak{Z}) = T^\alpha_{\subseteq}(\mathfrak{M}, \hat{\mathfrak{Z}}) \tag{2.1}$$

where $\hat{\mathfrak{Z}} := \{Z'' : Z'' \subseteq Z' \text{ for some } Z' \in \mathfrak{Z}\}$ and α, $\mathfrak{M}$ and $\mathfrak{Z}$ are arbitrary.

Moreover for deterministic devices one has

$$X^\omega \setminus T^\alpha_{\subseteq}(\mathfrak{M}, \mathfrak{Z}) = T^\alpha_{\nsubseteq}(\mathfrak{M}, \hat{\mathfrak{Z}}) \tag{2.2}$$

for arbitrary α and $\mathfrak{Z}$; $\hat{\mathfrak{Z}}$ being defind as above. By the construction of Lemma 7 in [93] (cf. also Lemma 4.1.2 in [17]) we can modify the machine $\mathfrak{M}$ (by adding a parallel finite-state control) in such a way that

$$T^\alpha_\sigma(\mathfrak{M}, \mathfrak{Z}) = T^\alpha_\sigma(\mathfrak{M}', \{Z'\}) \tag{2.3}$$

where $\sigma \in \{\subseteq, \nsubseteq\}$, thus accepting an ω-language by a single final set. In the remaining cases $(\boldsymbol{U}, =)$ and $(\boldsymbol{E}, =)$, however, as it was pointed out by *Wagner* [106, 109] (cf. also [41]), it is not possible to bound the cardinality of $\mathfrak{Z}$.

In these cases, due to the relation

$$T^{\alpha}_{=}(\mathfrak{M}, \{Z'\}) = T^{\alpha}_{\subseteqq}(\mathfrak{M}, \{Z'\}) \setminus \bigcap_{z \in Z'} T^{\alpha}_{\subseteqq}(\mathfrak{M}, \{Z' \setminus \{z\}\}) \tag{2.4}$$

holding true for deterministic machines, one obtains that in the deterministic case the types $(\alpha, =)$ are characterized by the types $(\alpha, \subseteqq)$ and $(\alpha, \nsubseteqq)$ via Boolean operations. If one adopts the point of view that acceptance by deterministic devices is merely a type of reduction via the continuous mapping $\Phi_{\mathfrak{M}}: X^{\omega} \to Z^{\omega}$ (cf. [93], and more explicitly in [109]), one can easily verify the inclusions of the DT1(α, σ) classes into the corresponding classes of the Borel hierarchy as presented in Theorem 2.1 below. The relations between DT1- and NT1-classes can be easily verified by the fact that NT1(α, σ) is the closure of DT1(α, σ) under projection $\overline{\mathrm{pr}}$ (cf. Theorem 5 in [110] or Theorem 12 in [95]). The same relation has been observed between NFA and DFA (cf. [100, 78, 86]); whether this same relation holds for NPDA and DPDA is an open problem.

Now, from [95] and [110] the following characterization of the classes of recursive ω-languages is known.

Theorem 2.1.

$$\begin{aligned}
&\mathrm{DT1}(\boldsymbol{E}, \subseteqq) = \mathrm{NT1}(\boldsymbol{E}, \subseteqq) \subseteqq \boldsymbol{F}\\
&\mathrm{DT1}(\boldsymbol{E}, \nsubseteqq) = \mathrm{NT1}(\boldsymbol{E}, \nsubseteqq) \subseteqq \boldsymbol{G}\\
&\mathrm{DT1}(\boldsymbol{E}, =) \subseteqq \mathscr{B}(\boldsymbol{F})\\
&\mathrm{NT1}(\boldsymbol{E}, =) = \mathrm{DT1}(\boldsymbol{U}, \subseteqq) = \mathrm{NT1}(\boldsymbol{U}, \subseteqq) \subseteqq \boldsymbol{F}_{\sigma}\\
&\mathrm{DT1}(\boldsymbol{U}, \nsubseteqq) \subseteqq \boldsymbol{G}_{\delta}\\
&\mathrm{DT1}(\boldsymbol{U}, =) \subseteqq \mathscr{B}(\boldsymbol{G}_{\delta})\\
&\mathrm{NT1}(\boldsymbol{U}, \nsubseteqq) = \mathrm{NT1}(\boldsymbol{U}, =) = \mathsf{REC}_{\omega} \subseteqq \boldsymbol{P}_1,
\end{aligned}$$

and, moreover these classes form a topologically based hierarchy.

We add some closure properties of the classes of recursive ω-languages mentioned in Theorem 2.1 (cf. [110, 18, 95, 88]).

Lemma 2.2. (i) *All classes are closed under union and intersection and inverse totally unbounded gsm-mappings.*

(ii) DT1$(\boldsymbol{E}, =)$ *and* DT1$(\boldsymbol{U}, =)$ *are closed under complementation, whereas all other classes are not closed under complementation.*

(iii) *All classes except* DT1$(\boldsymbol{E}, =)$ *and* DT1$(\boldsymbol{U}, \nsubseteqq)$ *are closed under projection.* (*As mentioned above* $\{\overline{\mathrm{pr}}\, E : E \in \mathrm{DT1}(\boldsymbol{U}, \nsubseteqq)\} = \mathrm{NT1}(\boldsymbol{U}, \nsubseteqq) = \mathsf{REC}_{\omega}$, *and* $\{\overline{\mathrm{pr}}\, E : E \in \mathrm{DT1}(\boldsymbol{E}, =)\} = \mathrm{NT1}(\boldsymbol{E}, =) = \mathrm{DT1}(\boldsymbol{U}, \subseteqq)$.)

In a similar manner we obtain for regular ω-languages the following (see [109] for detailed references).

Theorem 2.3.

$$\begin{aligned}
&\mathrm{DFA}(\boldsymbol{E}, \subseteqq) = \mathrm{NFA}(\boldsymbol{E}, \subseteqq) = \mathsf{R}_{\omega} \cap \boldsymbol{F}\\
&\mathrm{DFA}(\boldsymbol{E}, \nsubseteqq) = \mathrm{NFA}(\boldsymbol{E}, \nsubseteqq) = \mathsf{R}_{\omega} \cap \boldsymbol{G}\\
&\mathrm{DFA}(\boldsymbol{E}, =) = \mathsf{R}_{\omega} \cap \mathscr{B}(\boldsymbol{F}) = \mathsf{R}_{\omega} \cap \boldsymbol{F}_{\sigma} \cap \boldsymbol{G}_{\delta}\\
&\mathrm{NFA}(\boldsymbol{E}, =) = \mathrm{DFA}(\boldsymbol{U}, \subseteqq) = \mathrm{NFA}(\boldsymbol{U}, \subseteqq) = \mathsf{R}_{\omega} \cap \boldsymbol{F}_{\sigma}\\
&\mathrm{DFA}(\boldsymbol{U}, \nsubseteqq) = \mathsf{R}_{\omega} \cap \boldsymbol{G}_{\delta}\\
&\mathrm{NFA}(\boldsymbol{U}, \nsubseteqq) = \mathrm{DFA}(\boldsymbol{U}, =) = \mathrm{NFA}(\boldsymbol{U}, =) = \mathsf{R}_{\omega} \subseteqq \mathscr{B}(\boldsymbol{G}_{\delta}),
\end{aligned}$$

and, moreover, these classes form a topologically based hierarchy.

The classes of regular ω-languages presented in Theorem 2.3 have the same closure properties as the corresponding classes of recursive ω-languages. This may be derived from the closure properties of the class R^ω obtained by *Büchi* [7] and *Trakhtenbrot* [99] and the closure properties of the involved Borel classes, and from the results of [65] and [48].

Theorem 2.4 ([7, 99, 54]). (i) R_ω *is closed under Boolean operations and projection.*

(ii) $F \in \mathsf{R}_\omega$ *iff there are an* $n \in \mathbb{N}$ *and regular languages* W_i, V_i *such that*

$$F = \bigcup_{i=1}^{n} W_i \cdot V_i^\omega .$$

If we compare the results for recursive and regular ω-languages, we have to mention that such a tight topological characterization as in the case of the classes of regular ω-languages is impossible for the classes of recursive ω-languages; more exactly, in [88] we have exhibited an even finite ω-language in $\mathsf{REC}_\omega \setminus \mathrm{DT1}(\boldsymbol{U}, =)$.

Furthermore, a characterization like Theorem 2.4 (ii) is impossible for the class REC_ω. To this end we derive the following property (cf. [48, 94]).

Property 2.5. *Let* $F \subseteq X^\omega$ *be of the form* $F = \bigcup_{i=1}^{n} W_i \cdot V_i^\omega$ *for suitable languages* W_i, $V_i \subseteq X^*$. *If* F *is at most countable then* F *consists of only ultimately periodic sequences, and if* F *is not finite or countable then* card $F = 2^{\aleph_0}$.

Clearly, $\{010^2 10^3 1 \ldots\}$ is a finite set in REC_ω not containing an ultimately periodic sequence.

Before proceeding to context-free ω-languages we verify by the help of Examples 2.1 and 2.2 that the hierarchies of Theorems 2.1 and 2.3 are topologically based.

Example 2.3. The set $F := X^* \cdot 0^\omega$ is countable and dense in itself, i.e. $F \subseteq \boldsymbol{C}(F \setminus \{\beta\})$ for all $\beta \in X^\omega$. Hence, it is no $\boldsymbol{G}_\delta$-set (cf. [45]). Thus we verified $F \in \mathrm{DFA}(\boldsymbol{U}, \subseteq) \setminus \boldsymbol{G}_\delta$.

Accordingly we have $X^\omega \setminus F \in \mathrm{DFA}(\boldsymbol{U}, \subsetneqq) \setminus \boldsymbol{F}_\sigma$.

Similarly, $0^\omega \in \mathrm{DFA}(\boldsymbol{E}, \subseteq) \setminus \boldsymbol{G}$, and $X^\omega \setminus 0^\omega \in \mathrm{DFA}(\boldsymbol{E}, \subsetneqq) \setminus \boldsymbol{F}$.

These considerations prove that the hierarchy of regular ω-languages (Theorem 2.3) and the lower part of the hierarchy in Fig. 4 are topologically based.

It remains to give examples of ω-languages in $\mathsf{REC}_\omega \setminus \mathscr{B}(\boldsymbol{G}_\delta)$. Such examples are provided by the ω-languages S_n and P_n for $n \geq 3$ defined in [95, 110] (cf. also [88]).

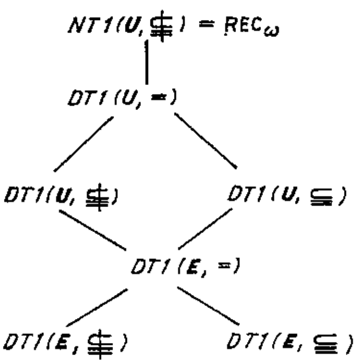

Fig. 4

Now, we turn to context-free ω-languages. In this case the relations between acceptance and topology are not explored to this extent as in the previous case.

First, we mention some properties of CF_ω derived by *Linna* [51, 52] and *Cohen* and *Gold* [15].

Lemma 2.6. (i) *$F \in \mathsf{CF}_\omega$ iff there are an $n \in N$ and context-free languages W_i, V_i such that*

$$F = \bigcup_{i=1}^{n} W_i \cdot V_i^\omega .$$

(ii) *CF_ω is closed under union and projection, intersection with regular ω-languages, but neither under intersection nor under complementation.*

Thus CF_ω is derived in the same (natural) way from the family CF as R_ω is derived from the family R.

Next, we state some topological results about the classes of accepted languages. Since, on the one hand, the acceptance results for nondeterministic pushdown automata obtained by *Cohen* and *Gold* [15, 16] are in many cases incomparable to our notion of acceptance and, on the other hand, it is still open whether a projection lemma linking the NPDA-classes to the DPDA-classes holds, we can only recall some results on DPDA($\boldsymbol{E}$, σ) by *Cohen* and *Gold* [17] and on DPDA ($\boldsymbol{U}$, σ) by *Linna* [53].

Theorem 2.7.

$$\begin{aligned} \mathrm{DPDA}(\boldsymbol{E}, \subseteqq) &= \mathsf{DCF}_\omega \cap \boldsymbol{F} \\ \mathrm{DPDA}(\boldsymbol{E}, \subsetneqq) &= \mathsf{DCF}_\omega \cap \boldsymbol{G} \\ \mathrm{DPDA}(\boldsymbol{U}, \subseteqq) &= \mathsf{DCF}_\omega \cap \boldsymbol{F}_\sigma \\ \mathrm{DPDA}(\boldsymbol{U}, \subsetneqq) &= \mathsf{DCF}_\omega \cap \boldsymbol{G}_\delta , \end{aligned}$$

and moreover, these classes form a topologically based hierarchy.

Since the class DCF_ω does not possess many closure properties (cf. [51, 52, 17]), we will not investigate the closure properties of the above classes.

After the discussion of acceptance results for fully defined devices, we shall add some words about partially defined devices. As it might be seen immediately by adding a dead sink in case of a nondefined next move and an obvious change of the family of final sets $\mathfrak{Z}$ in case of the type ($\boldsymbol{U}$, $\subsetneqq$), we can expect new results only for the type ($\boldsymbol{E}$, $\subsetneqq$).

We present here these results obtained by *Wagner* [105] and in [78] for finite automata.

Lemma 2.8.

$$\mathrm{DPFA}(\boldsymbol{E}, \subsetneqq) = \{E \cap F \colon E \in \mathrm{DFA}(\boldsymbol{E}, \subsetneqq) \text{ and } F \in \mathrm{DFA}(\boldsymbol{E}, \subseteqq)\}$$

$$\mathrm{NPFA}(\boldsymbol{E}, \subsetneqq) = \mathrm{DFA}(\boldsymbol{U}, \subseteqq)$$

(*The additional P denotes that partial automata are involved.*)

It seems to be straightforward to extend these results to the case of Turing machines utilizing a projection lemma for the nondeterministic case, as it was done in [78] for finite automata.

A characterization corresponding to the one of DPFA was obtained by *Cohen* and *Gold* [17; Theorem 4.2.7] for deterministic pushdown automata, however, since in this case the sets E and F have to satisfy some compatibility condition, we just mention this reference and omit details.

Concluding this section we mention some open problems.

1. The results of the Theorems 2.1 and 2.3 contain the following projection lemmas:

$$\mathrm{NFA}(\alpha, \sigma) = \{\overline{\mathrm{pr}}\, F : F \in \mathrm{DFA}(\alpha, \sigma)\}, \quad \text{and}$$
$$\mathrm{NT1}(\alpha, \sigma) = \{\overline{\mathrm{pr}}\, F : F \in \mathrm{DT1}(\alpha, \sigma)\}.$$

(In fact, in the latter case the projection lemma was utilized to characterize the NT1-classes [95, 110].) Moreover, from [86] we know that projection lemmas hold also in the case of infinite automata.

What can be said in the case of pushdown automata? We have the obvious inclusion

$$\mathrm{NPDA}(\alpha, \sigma) \supseteqq \{\overline{\mathrm{pr}}\, F : F \in \mathrm{DPDA}(\alpha, \sigma)\},$$

but is this inclusion proper or not?

2. Property 2.5 and the example following it show that a characterization of REC_ω analogous to Theorem 2.4. ii and Lemma 2.6. i is not possible. What can be said about the following two subclasses of REC_ω:

$$\{\bigcup_{i=1}^{n} W_i \cdot V_i^\omega : W_i, V_i \in \mathsf{REC}, n \in N\} \subseteqq \{\bigcup_{i=1}^{n} W_i \cdot V_i^\omega : W_i, V_i \in \mathsf{RE}, n \in N\}.$$

Is this inclusion proper? (The properness proof given by *Istrail* [34; Proposition 3] is not valid.)

Which one of the above classes is contained in $\mathrm{DT1}(\boldsymbol{U}, =)$?

3. Theorem 2.3 shows $\mathrm{DFA}(\boldsymbol{E}, =) = \mathrm{DFA}(\boldsymbol{U}, \subseteqq) \cap \mathrm{DFA}(\boldsymbol{U}, \nsubseteq)$, and in [88] it is shown that $\mathrm{DT1}(\boldsymbol{E}, =) \subset \mathrm{DT1}(\boldsymbol{U}, \subseteqq) \cap \mathrm{DT1}(\boldsymbol{U}, \nsubseteq)$. For deterministic pushdown automata we obtain from Theorems 2.1 and 2.7 the inclusion $\mathrm{DPDA}(\boldsymbol{E}, =) \subseteqq \subseteqq \mathrm{DPDA}(\boldsymbol{U}, \subseteqq) \cap \mathrm{DPDA}(\boldsymbol{U}, \nsubseteq)$. It is conjectured that proper inclusion holds true.

3. Operations for ω-languages

In the previous section we have considered in some detail the classes of ω-languages R_ω, DCF_ω, CF_ω, and REC_ω. Characterizations of the classes R_ω and CF_ω via languages from R and CF resp. are given in Theorem 2.4 and Lemma 2.6. In this section we study in more detail the relations of the above mentioned classes of ω-languages to their language counterparts. It is quite obvious that the counterparts to R_ω, DCF_ω, and CF_ω are the families R, DCF, and CF resp. when we quote the Init-lemma (Lemma 7.21 in [18]) which characterizes the initial word languages of the ω-languages in those classes.

Lemma 3.1. *If* $F \in \mathsf{R}_\omega$ (CF_ω, DCF_ω *resp.*) *then* $\boldsymbol{A}(F) \in \mathsf{R}$ (CF, DCF *resp.*).

The R- and the CF-part of this lemma are easily obtained from Theorem 2.4 and Lemma 2.6 resp., and the DCF-part is shown by *Cohen* and *Gold* (Corollary 2.11 in [17]).

For recursive ω-languages, however, *Cohen* and *Gold* in [18] could only show that $\boldsymbol{A}(F) \in \mathsf{RE}$ when $F \in \mathrm{DT1}(\boldsymbol{E}, \nsubseteq)$ and that $\boldsymbol{A}(F)$ need not be recursively enumerable if $F \in \mathrm{DT1}(\boldsymbol{E}, \subseteqq)$.

Utilizing well-known techniques from predicate calculus (cf. [68; chs. 14—16]), a characterization of the initial word languages of all classes of recursive ω-languages introduced in the previous section was obtained in [89]. Here, we confine ourselves to the consideration of the whole class REC_ω. To this end we define the first Σ-class of the analytical hierarchy of languages (cf. [68]):

$$W \in \Sigma_1^1 \text{ iff } W = \{w : \exists\beta\forall v\exists u(u \sqsubset \beta \wedge (w, v, u) \in M)\}$$

where $M \subseteq X^* \times X^* \times X^*$ is a recursive relation.

Furthermore, we denote by $\mathcal{A}(\mathsf{L}_\omega)$ for a family L_ω of ω-languages the smallest family of languages obtained from the set $\{A(F): F \in \mathsf{L}_\omega\}$ by closure under intersection with (regular) languages of the form $Y^* \cdot x$ ($Y \subset X$ and $x \in X \setminus Y$) and removal of the last letter (endmarker).

Now, from Lemma 3.1 and Theorem 9 of [89] we get the following concise form of the Init-lemma.

Lemma 3.2 (Init-lemma).

$$\mathcal{A}(\mathsf{R}_\omega) = \mathsf{R}\,,$$
$$\mathcal{A}(\mathsf{DCF}_\omega) = \mathsf{DCF}\,,$$
$$\mathcal{A}(\mathsf{CF}_\omega) = \mathsf{CF}\,, \quad \textit{and}$$
$$\mathcal{A}(\mathrm{DT1}(\boldsymbol{U}, \subsetneqq)) = \mathcal{A}(\mathsf{REC}_\omega) = \Sigma_1^1\,.$$

Remark. The last identity yields a simple proof that REC_ω is not closed under complementation utilizing that the class Σ_1^1 is not closed under complementation.

After having explored the initial word languages we turn to the question which one of the operations **ls**, **lim**, $^\delta$, and $^\omega$ defined in Section 1 leads back from the above classes of languages to their ω-language counterpart.

A first result is readily seen.

Property 3.3. *If W is a regular language then* **ls** W, **lim** W, W^δ, *and* W^ω *are regular ω-languages.*

Utilizing the Tarski-Kuratowski algorithm and the properties of the class Σ_1^1 (cf. [68; chs. 14—16]) one easily derives the following result.

Property 3.4. *If $W \in \Sigma_1^1$ then* **ls** W, **lim** W, W^δ *as well as* W^ω *are recursive ω-languages.*

Remark. Although **ls** W, **lim** W are closed sets and $W^\delta \in G_\delta$, one cannot guarantee that **ls** W or **lim** W are in $\mathrm{DT1}(\boldsymbol{E}, \subseteq)$, or W^δ is in $\mathrm{DT1}(\boldsymbol{U}, \subsetneqq)$. For more detailed information refer back to Lemma 3.2, to Table 3 below or to [88] and [89].

In the context-free case we have the following.

Property 3.5. (i) *If V is a deterministic context-free language then* **ls** V, **lim** V, *and* V^δ *are in* DCF_ω.

(ii) *If U is a context-free language then* **ls** U *and* U^ω *are in* CF_ω.

The proof of (i) is straightforward by automata constructions (cf. [17, 51, 52]), and in [11] it is shown that **ls** $U \in \mathsf{CF}_\omega$ when $U \in \mathsf{CF}$. The remaining assertion follows from Lemma 2.6.

Other closure properties do not hold:

1. Generalizing an example of *Ginsburg* and *Greibach* [27] *Cohen* and *Gold* [17; Proposition 2.17] showed that $V^\omega \notin \mathsf{DCF}_\omega$ for some $V \in \mathsf{DCF}$.

2. *Ginsburg*, *Hibbard* and *Ullian* [29; p. 330] gave an example of a context-free language U' such that **lim** $U' = \{\beta\}$ where β is not ultimately periodic. Hence, according to Lemma 2.6 and Property 2.5 **lim** $U' \notin \mathsf{CF}_\omega$.

3. Finally, *Linna* in [51] and [52; Example 4.1] constructed an even linear language U such that U^δ does not contain any ultimately periodic $\beta \in X_\omega$. Hence, $U^\delta \notin \mathsf{CF}_\omega$.

In Tables 1, 2 and 3 we give representations of the classes of ω-languages discussed in Section 2 by their language counterparts and the operations considered here. The references correspond to the earliest (to the knowledge of the author) appearence of the representation and/or condition. In Tables 1 and 2 (not in 3), due to Properties 3.3 and 3.5. i, we can substitute the operation **ls** by **lim** as well, and as usual we call a language W prefix-free provided $w \subsetneqq v$ for $w, v \in W$ implies $w = v$.

Table 1
Representation of regular ω-languages by regular languages

Type	Representation	Condition	References
$\mathrm{DFA}(\boldsymbol{E}, \subsetneqq) = \mathsf{R}_\omega \cap \boldsymbol{G}$	$W \cdot X^\omega$	—	[31], [46]
$\mathrm{DFA}(\boldsymbol{E}, \subseteqq) = \mathsf{R}_\omega \cap \boldsymbol{F}$	$\mathbf{ls}\, W$	—	[76], [93], [48]
$\mathrm{DPFA}(\boldsymbol{E}, \subsetneqq)$	$\bigcup_{i=1}^{n} W_i \cdot \mathbf{ls}\, V_i$	$\bigcup_{i=1}^{n} W_i$ prefix-free	[78]
$\mathrm{DFA}(\boldsymbol{E}, =) = \mathsf{R}_\omega \cap \boldsymbol{G}_\delta \cap \boldsymbol{F}_\sigma$	$\bigcup_{i=1}^{n} W_i \cdot \mathbf{ls}\, V_i$	W_i prefix-free	[93]
$\mathrm{DFA}(\boldsymbol{U}, \subseteqq) = \mathsf{R}_\omega \cap \boldsymbol{F}_\sigma$	$\bigcup_{i=1}^{n} W_i \cdot \mathbf{ls}\, V_i$	—	[93]
$\mathrm{DFA}(\boldsymbol{U}, \subsetneqq) = \mathsf{R}_\omega \cap \boldsymbol{G}_\delta$	W^δ; $\bigcup_{i=1}^{n} W_i \cdot V_i^\omega$	W_i, V_i prefix-free	[14], [23], [93]
$\mathrm{DFA}(\boldsymbol{U}, =) = \mathsf{R}^\omega$	$\overline{\mathrm{pr}}(W^\delta)$; $\bigcup_{i=1}^{n} W_i \cdot V_i^\omega$	V_i prefix-free	[100], [14], [23]

Table 2
Representation of deterministic context-free ω-languages by deterministic context-free languages

Type	Representation	References
$\mathrm{DPDA}(\boldsymbol{E}, \subsetneqq) = \mathsf{DCF}_\omega \cap \boldsymbol{G}$	$W \cdot X^\omega$	[17]
$\mathrm{DPDA}(\boldsymbol{E}, \subseteqq) = \mathsf{DCF}_\omega \cap \boldsymbol{F}$	$\mathbf{ls}\, W$	[17]
$\mathrm{DPDA}(\boldsymbol{U}, \subsetneqq) = \mathsf{DCF}_\omega \cap \boldsymbol{G}_\delta$	W^δ	[51, 52]

Table 3
Representation of recursive ω-languages by recursive (recursively enumerable) languages

Type	Representation	Condition	References
$\mathrm{DT1}(\boldsymbol{E}, \subsetneqq)$	$W \cdot X^\omega$	—	[18]
$\mathrm{DT1}(\boldsymbol{E}, \subseteqq)$	$\mathbf{lim}\, W$	only for recursive W	[18], [88]
$\mathrm{DT1}(\boldsymbol{U}, \subsetneqq)$	W^δ	—	[18]
REC_ω	$\overline{\mathrm{pr}}(W^\delta)$	—	[95], [110]

It is worth mentioning that the investigations of the relations between classes of ω-languages and their language counterparts may also lead back to contributions to language theory. As an example we quote the study of centers of languages (cf. [12]) which was initiated by *Nivat* [59]. The center of a language $W \subseteq X^*$ is the set of all words in $\boldsymbol{A}(W)$ having infinitely many successors in W, i.e. center$(W) := \boldsymbol{A}(\mathbf{ls}\ W)$.

So it turns out that center$(W) \in \mathsf{R}$ (DCF, CF) for every $W \in \mathsf{R}$ (DCF, CF) could be also derived by using the above Lemma 3.1 and Property 3.5. Recently, using the theory of recursive ω-languages developed in [88], we gave in [92] a thorough characterization of the centers of recursive languages which were shown to be not necessarily recursively enumerable already by *Prodinger* and *Urbanek* [64].

Finally, we return to algebraic questions in the theory of ω-languages.

Up to now there seems to be no unified treatise in developing closure properties of families of ω-languages by deriving them from closure properties of their language counterparts. Scattered in many papers, however, there are results stating more or less implicitly closure properties of families of ω-languages derived from closure properties of the respective families of languages. More detailed are some results by *Cohen* and *Gold* [16] and in [91].

Furthermore, *Wagner* and the author in [94] proposed a theory of an algebraic treatise of ω-languages analogous to the AFL-theory of *Ginsburg* and *Greibach* [28, 26]. In connection with this, one can see the characterization of families of ω-languages accepted under empty-storage acceptance by AFAs in [80].

4. Complexity of acceptance

The aim of this section is to lead the reader's attention to a further refinement of the topologically based hierarchy of regular ω-languages (cf. [105, 109]). Since it would go beyond the scope of this paper, for details we refer to [109] and scetch here only the basic ideas.

Already in [102, 103] *Wagner* introduced complexity measures for deterministic automata $\mathfrak{A}$ accepting ω-languages with respect to the type $(\boldsymbol{U}, =)$. It turns out that these measures are independent of the particular pair $(\mathfrak{A}, \mathfrak{Z})$ accepting an ω-language $F = T^{U}_{=}(\mathfrak{A}, \mathfrak{Z})$. Moreover, these measures and the corresponding complexity classes represent natural properties of regular ω-languages, in particular the classes DFA$(\boldsymbol{E}, \subseteqq) = \boldsymbol{F} \cap \mathsf{R}_\omega$, DFA$(\boldsymbol{E}, \nsubseteq) = \boldsymbol{G} \cap \mathsf{R}_\omega$, DFA$(\boldsymbol{U}, \subseteqq) = \boldsymbol{F}_\sigma \cap \mathsf{R}_\omega$ and DFA$(\boldsymbol{U}, \nsubseteq) = \boldsymbol{G}_\delta \cap \mathsf{R}_\omega$ appear as complexity classes, and DFA$(\boldsymbol{E}, =) = \boldsymbol{F}_\sigma \cap \boldsymbol{G}_\delta \cap \mathsf{R}_\omega$ is a union of complexity classes.

Wagner characterizes these complexity classes also in two other ways:

The first is by using m-reducibility with totally unbounded gsm-mappings, i.e. an ω-language E is said to be reducible to an ω-language F iff there is a totally unbounded gsm $\mathfrak{A}$ such that $E = \bar{\varphi}_{\mathfrak{A}}^{-1}(F)$ (cf. with m-reducibility of sets of numbers [68; ch. 7]).

The second way is by topological difficulty. A weaker form of this classification was investigated by *Kaminski* [41] who termed it the "Boolean hierarchy". *Kaminski* measures the necessary complexity of representation of a set $F \in \mathsf{R}_\omega \subseteqq \mathscr{B}(\boldsymbol{G}_\delta)$ as a certain Boolean normal form with entries from the family $\boldsymbol{G}_\delta \cap \mathsf{R}_\omega$. *Wagner* still refines this hierarchy by more complicated topological arguments, and adds below the classes $\boldsymbol{G}_\delta \cap \mathsf{R}_\omega$ and $\boldsymbol{F}_\delta \cap \mathsf{R}_\omega$ the "Boolean hierarchy" in DFA$(\boldsymbol{E}, =) = \mathscr{B}(\boldsymbol{G}) \cap \mathsf{R}_\omega$ with respect to the representation with entries from $\boldsymbol{G} \cap \mathsf{R}_\omega$.

We illustrate the concept of Boolean hierarchy by the following simple example connecting it to another complexity measure for regular ω-languages ([106] and [109; Sec. 8]).

Example 4.1. We consider Rabin's acceptance notion [65]: An ω-language F is accepted by a deterministic automaton $\mathfrak{A}$ and a final set $\mathfrak{S}$ of pairs of subsets of Z $(F = T_R(\mathfrak{A}, \mathfrak{S}))$ iff

$$F = \bigcup_{(Z',Z'') \in \mathfrak{S}} \left(T^U_{\subseteq}(\mathfrak{A}, \{Z'\}) \cap T^U_{\nsubseteq}(\mathfrak{A}, \{Z''\})\right).$$

The Rabin index (complexity) $I_R(F)$ of an ω-language F is the smallest number of pairs needed for the acceptance of the set F by any finite deterministic automaton. In Theorem 17 of [106] it is proved that I_R strongly corresponds to one of the above mentioned complexity measures.

From Theorem 2.3 it is evident that

$$\{F: I_R(F) \leqq m\} = \{\bigcup_{i=1}^{m} (F_i \setminus E_i): F_i, E_i \in G_\delta \cap \mathsf{R}_\omega\},$$

and that this class is closed under the above mentioned m-reducibility with totally unbounded gsm-mappings.

Though Wagner's results are interesting for their own, they can be also applied to obtain hierarchy results for other classes of ω-languages. We mention here only the following.

The complexity of acceptance in Wagner's hierarchy is measured by two parameters m and n (ranging up to infinity). Theorem 18 of [109] states that if $F = Z^U_=(\mathfrak{A}, \mathfrak{Z})$ has complexity parameters m and n then the family $\mathfrak{Z}$ of final sets contains at least $(m \cdot n - 1)/2$ elements. Since the complexity of acceptance is equivalent to topological complexity, one can conclude that even a Turing machine cannot accept the ω-language F using a family of final sets having less than $(m \cdot n - 1)/2$ elements. This observation might be helpful in simplifying hierarchy results in [19].

5. Finite-state ω-languages

Up to now we have only considered classes of ω-languages defined by accepting devices. But already one of the first papers dealing with ω-languages, the initiating paper by *Trakhtenbrot* [99] introduces the class of finite-state ω-languages by using the structure of an automaton associated with the given ω-language.

We start with some definitions.

Definition. Let $B \subseteqq X^* \cup X^\omega$ and $w \in X^*$. We refer to the set $B/w := \{b: w \cdot b \in B\}$ as the *state* (or *left derivative* [13]) of the set B derived by the word w, and we call B *finite-state* provided its set of states $Z_B := \{B/w: w \in X^*\}$ is finite.

To any set $B \subseteqq X^* \cup X^\omega$ we define the automaton associated with the set B as $\mathfrak{A}_B := (X, Z_B, f, z_0)$ where Z_B is the set of states of B, $f(B/w, x) := B/w \cdot x$, and $z_0 := B = B/e$.

Thus, B is finite-state iff $\mathfrak{A}_B$ is a finite automaton. An equivalent condition (cf. [38, 39, 40]) is that the syntactic monoid $\mathrm{Syn}(B) := X^*/\sim$ be finite, where the syntactic congruence $\sim$ is defined as usual:

$$w \sim v \quad \text{iff} \quad \forall u \forall b \left(u \in X^* \wedge b \in X^* \cup X^\omega \rightarrow (u \cdot w \cdot b \in B \leftrightarrow u \cdot v \cdot b \in B)\right)$$

Remark. Note that for regular ω-languages there are also other definitions of syntactic congruence in use (cf. [4, 63]).

As it is well-known the finite-state languages are exactly the regular languages. But in the case of ω-languages the two classes are distinct, as it was shown by *Trakhtenbrot* in [99], who gave an example of an uncountable family of countable one-state

ω-languages. One-state ω-languages were investigated in detail by *Jürgensen* and *Thierrin* [38] as the ω-languages whose syntactic monoid is trivial, i.e. $w \sim v$ for arbitrary $w, v \in X^*$. Moreover, in the subsequent paper [40] they obtained the following difference between finite-state languages and ω-languages. As it is well-known, there is a finite monoid nonisomorphic to any syntactic monoid of a language. In the case of ω-languages, however this statement is not true (cf. [40]).

Theorem 5.1. *For every finite monoid M there are an alphabet X and an ω-language $F \subseteq X^\omega$ such that* $\mathrm{Syn}(F)$ *is isomorphic to M.*

Now, we shall return to Trakhtenbrot's results. In order to simplify the notation we introduce

$$\mathsf{FS}_\omega := \{F : F \subseteq X^\omega \text{ and } F \text{ is finite-state}\} .$$

It is shown in [99] (cf. also [48] and [84]):

Proposition 5.2.

(i) $\mathsf{R}_\omega \subseteq \mathsf{FS}_\omega$,
(ii) $\mathsf{FS}_\omega \cap \boldsymbol{F}_\sigma \subsetneqq \mathsf{R}_\omega$,
(iii) $\mathsf{FS}_\omega \cap \boldsymbol{F} \subseteq \mathsf{R}_\omega$.

The following closure properties are shown in [99], [84] and [48; Lemma 6.25].

Proposition 5.3. *The family of finite-state ω-languages is closed under Boolean operations, multiplication with regular languages from the left, gsm-mappings and their inverse mappings.*

In [84] it is proved an exact borderline in the Borel hierarchy between the ranges where FS_ω and R_ω coincide and where not. Together with the above Proposition 5.2. ii the following inclusion states the result.

$$\mathsf{FS}_\omega \cap \boldsymbol{F}_\sigma \cap \boldsymbol{G}_\delta \subseteq \mathsf{R}_\omega . \tag{5.1}$$

Other attempts, to discover computational constraints under which finite-state ω-languages are necessarily regular, failed so far. So in [78; 81] a finite-state ω-language accepted by a deterministic one-counter automaton under empty-storage acceptance is constructed as follows.

Starting from the simple deterministic language $V \subseteq \{0, 1\}^*$ (in the sense of *Korenjak* and *Hopcroft* [44]) defined by the equation

$$V = 0 \cup 1 \cdot V^2 \tag{5.2}$$

utilizing Theorem 3.6 of [17] one can show that V^ω is accepted by a deterministic one-counter automaton under empty-storage acceptance. From (5.2) we get immediately $V^\omega = X \cdot V^\omega$, thus V^ω has only one state.

In order to verify that V^ω is not regular we use the following measure-category theorem for regular ω-languages from [79] (cf. also [81]).

Theorem 5.4. *Let $\mu: X \to (0, 1)$ be a probability measure, and let $\bar{\mu}$ be the corresponding product measure on X^ω.*

If $F \subseteq X^\omega$ is regular then $\bar{\mu}(F) = 0$ iff F is of the first Baire category (i.e. F is a countable union of nowhere dense subsets of X^ω).

As an immediate consequence of this theorem we obtain that the property $\bar{\mu}(F) = 0$ is independent of the particular measure $\mu: X \to (0, 1)$ when F is regular. However, for the ω-language V^ω, utilizing (5.2), one can easily calculate

$$\bar{\mu}(V^\omega) = \begin{cases} 1 & \text{if } \mu(0) \geq \frac{1}{2}, \\ 0 & \text{if } \mu(0) < \frac{1}{2}. \end{cases}$$

Hence, V^ω is not regular.

We conclude this section by considering the relationship between the associated and accepting automata for a regular ω-language. A first result is Theorem 4 of [84].

Theorem 5.5. *Let $F \subseteq X^\omega$ and let $\mathfrak{A}$ be a finite automaton such that $F = T^U_=(\mathfrak{A}, \mathfrak{Z})$ (or likewise $F = T^U_\subseteq(\mathfrak{A}, \mathfrak{Z})$ or $F = T^U_\supseteq(\mathfrak{A}, \mathfrak{Z})$) for some $\mathfrak{Z}$. Then the associated automaton $\mathfrak{A}_F$ is a homomorphic image of $\mathfrak{A}$.*

Remark. This theorem does not remain true if we change the acceptance type (U, σ) to (E, σ). In [109; ch. 8] *Wagner* gives an example of a three-state automaton $\mathfrak{A}_2$ accepting the set $E = 0 \cdot X^\omega$ according to the type $(E, =)$ (or $(E, \subseteq)$) and not being isomorphic to the three-state automaton $\mathfrak{A}_F$ ($= \mathfrak{A}_1$ in [109]). But, as we shall see in the following example, an accepting automaton (according to the type $(E, =)$) may have even less states than the corresponding associated automaton.

Example 5.1. Set $\mathfrak{A} := (\{0, 1\}, \{a, b, c, d\}, a, f)$ where the next-state function f is defined as follows:

f	a	b	c	d
0	b	c	d	d
1	c	d	b	d

Then we have

$$\beta \in 01 \cdot \{0, 1\}^\omega \quad \text{iff} \quad E(\Phi_{\mathfrak{A}}(\beta)) = \{b, d\},$$
$$\beta \in 10 \cdot \{0, 1\}^\omega \quad \text{iff} \quad E(\Phi_{\mathfrak{A}}(\beta)) = \{c, d\}, \quad \text{and}$$
$$E(\Phi_{\mathfrak{A}}(\beta)) \supseteq \{b, c\} \text{ otherwise}.$$

Consequently, $\{01, 10\} \cdot \{0, 1\}^\omega = T^U_=(\mathfrak{A}, \{\{b, d\}, \{c, d\}\})$, and it is easy to verify that the automaton associated with the ω-language $\{01, 10\} \cdot \{0, 1\}^\omega$ has five states.

Our Theorem 5.5 leads to the following contribution to the minimization problem for ω-automata.

Corollary 5.6. *Let $F = T^U_=(\mathfrak{A}_F, \mathfrak{Z})$ for some $\mathfrak{Z}$ and let $\mathfrak{A}_F$ be a finite automaton. Then $\mathfrak{A}_F$ is the unique (up to isomorphism) minimal automaton accepting F according to the type $(U, =)$.*

Moreover, in Theorem 24 of [84] we obtained a sufficient condition for a regular ω-language to be accepted by its associated automaton.

Theorem 5.7. *If $F \in \boldsymbol{F}_\sigma \cap \boldsymbol{G}_\delta \cap \mathsf{R}_\omega$ then there is a $\mathfrak{Z}$ such that $F = T^U_=(\mathfrak{A}_F, \mathfrak{Z})$.*

However, as it was also shown in [84], this condition is not necessary, though Example 2.2 proves that Theorem 5.7 is not valid already for ω-languages in $\boldsymbol{F}_\sigma \cap \mathsf{R}_\omega$ or $\boldsymbol{G}_\delta \cap \mathsf{R}_\omega$. Those ω-languages may have even several nonisomorphic minimal accepting automata.

We add some open problems concerning results of this section.

1. Is there any nontrivial computational constraint which guarantees that a finite-state ω-language accepted by a machine satisfying this constraint is regular?

2. Characterize those ω-languages having exactly one (up to isomorphism) minimal automaton, and characterize those ω-languages having an accepting automaton isomorphic to their associated automaton. Does the former property necessarily imply the latter one?

3. Investigate the minimal accepting automata according to the types (E, σ) for ω-languages in $\boldsymbol{F}_\sigma \cap \boldsymbol{G}_\delta \cap \mathsf{R}_\omega$.

6. Systems of equations and formal proof systems for ω-regular expressions

Formal proof systems (or axiom systems) for the algebra of regular events (languages) have been constructed by *Salomaa* [69, 70]. These axiom systems have a relatively simple structure due to the fact that regular languages are uniquely characterized as solutions of systems of linear equations of the form

$$V_i = \bigcup_{j=1}^{n} W_{ij} \cdot V_j \cup U_i \qquad (6.1)$$

where $W_{ij}, U_i \subseteqq X^*$ are finite languages.

As we shall see below, an analogous statement is no longer true in the case of ω-languages. Moreover, one of Salomaa's axiom systems relies heavily on the following derivation rule for the elimination of the unknown language V:

$$\text{If } V = W \cdot V \cup U \text{ and } e \notin W, \text{ then } V = W^* \cdot U .$$

In [66] and [77] (cf. also [48]) the same equation

$$F = W \cdot F \cup E , \qquad (6.2)$$

where $W \subseteqq X^* \setminus \{e\}$ and $E \subseteqq X^\omega$, was considered for ω-languages.

It was observed that such an equation may have infinitely many solutions (in particular, $F = X \cdot F$ has $2^{2^{\aleph_0}}$ solutions [84; Lemma 7]), all of them are included between the minimum solution $F = W^* \cdot E$ and the maximum solution $F = W^\omega \cup W^* \cdot E$. This fact was used to show that $\text{ls}\ W^* = W^\omega \cup W^* \cdot \text{ls}\ W$ [48; Lemma 5.61 b] and $(W^*)^\delta = W^\omega \cup W^* \cdot W^\delta$ [82; Lemma 14]. Furthermore, it implies that systems of linear equations of the form

$$F_i = \bigcup_{j=1}^{n} W_{ij} \cdot F_j \cup E_i , \qquad (6.3)$$

where $W_{ij} \subseteqq X^* \setminus \{e\}$ and $E \subseteqq X^\omega$, are in general not uniquely solvable for ω-languages F_i. It holds only the following [48].

Lemma 6.1. *If in* (6.3) *all languages W_{ij} are finite and all E_i are empty sets then this system has a unique maximum solution F_1 which is closed and can be obtained by solving the system according to the maximum solution principle for* (6.2).

Though *Park* [62] obtained a characterization of all regular ω-languages by systems of the form (6.3) via a complicated minimum-maximum solution principle, it was already shown by Wagner's axiom system [104] that it is possible to construct a proof system for ω-regular expressions which resembles strongly Salomaa's system for regular expressions utilizing the following simple rule for the elimination of the expression for the regular language V: If

$$(V \cup W)^\omega = W \cdot (V \cup W)^\omega \cup E , \quad e \notin W , \quad E \subseteqq X^\omega$$

then

$$(V \cup W)^\omega = W^\omega \cup W^* \cdot E ,$$

which is a special case of the maximum solution principle for (6.2).

Recently, other formal proof systems for ω-regular expressions were given by *Izumi, Inagaki* and *Honda* [35], and *Darondeau* and *Kott* [21] (cf. also their contribution in [61]).

7. Concluding remarks

As it was pointed out in the introduction, the theory of ω-languages emerged in the early sixties as a tool for deriving decision procedures in monadic second-order logic.

But since then this theory has been proved to be applicable not only to problems in mathematical logic. Here we shall briefly mention some other areas in which methods and results of the theory of ω-languages were applied.

As a first topic we mention the study of the behaviour of processes by *Nivat* et al. (cf. [12], [60] and the references quoted there), and the application of regular ω-languages to the modelling of fair concurrency as it was described by *Redziejowski* [66] and *Park* [62].

Schnorr's book [72] presents a survey on results on randomness and complexity of infinite sequences. It includes also many results belonging to the theory of ω-languages. Other results in this area concern a connection between the entropy of an ω-language F (for a definition see [48] or [87]) and the Kolmogorov program complexity of the sequences $\beta \in F$ presented in [83], and an approach to the inference of nonrandom sequences utilizing topological and measure-theoretic properties of ω-languages (cf. [47, 49, 20]).

The last example belongs to the field of error control codes. Here it became possible, by means of structural properties of ω-languages, to characterize among the many subspaces of X^ω (X being a finite field) those which are convolutional codes [85].

References

[1] *Alaiwan, H.*: Equivalence of infinite behaviour of finite automata. Theoret. Comput. Sci. **31** (1984) 4, 297—306.

[2] *Arnold, A.*: Topological characterizations of infinite behaviours of transition systems. In: Automata, Languages and Programming; ed.: *J. Diaz*; Lect. Notes Comput. Sci., 154; Springer, 1983; pp. 28—38.

[3] *Arnold, A.*: Rational ω-languages are non-ambiguous. Theoret. Comput. Sci. **26** (1983), 221—223.

[4] *Arnold, A.*: A syntactic congruence for rational ω-languages. Theoret. Comput. Sci. **39** (1985) 2/3, 333—335.

[5] *Beauquier, D., D. Perrin*: Codeterministic automata on infinite words. Inform. Process. Letters **20** (1985), 95—98.

[6] *Berstel, J.*: Transductors and Context-free Languages. Teubner, Stuttgart 1979.

[7] *Büchi, J. R.*: On a decision method in restricted second order arithmetic. In: Proc. 1960 Int. Congr. for Logic; Standford Univ. Press, Stanford 1962; pp. 1—11.

[8] *Büchi, J. R.*: State-strategies for games in $F_{\sigma\delta} \cap G_{\delta\sigma}$. J. Symb. Logic **48** (1983) 4, 1171—1198.

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

[10] *Büchi, J. R., L. H. Landweber*: Definability in the monadic second-order theory of successor. J. Symb. Logic **34** (1969) 2, 166—170.

[11] *Boasson, L., M. Nivat*: Adherences of languages. J. Comput. System Sci. **20** (1980), 285—309.

[12] *Boasson, L., M. Nivat*: Centers of languages. In: Theoretical Computer Science; ed.: *P. Deussen*; Lect. Notes Comput. Sci., 104; Springer, 1981; pp. 245—251.

[13] *Brzozowski, J. A.*: Derivatives of regular expressions. J. ACM **11** (1964) 4, 481—494.

[14] *Choueka, Y.*: Theories of automata on ω-tapes. J. Comput. System Sci. 8 (1974), 117—141.

[15] *Cohen, R. S., A. Y. Gold*: Theory of ω-languages; I: Characterizations of ω-context-free languages. J. Comput. System Sci. **15** (1977) 2, 169—184.

[16] *Cohen, R. S., A. Y. Gold*: Theory of ω-languages; II: A study of various models of ω-type generation and recognition. J. Comput. System Sci. **15** (1977) 2, 185—208.

[17] *Cohen, R. S., A. Y. Gold*: ω-computations on deterministic push down machines. J. Comput. System Sci. **16** (1978) 3, 257—300.

[18] *Cohen, R. S., A. Y. Gold*: ω-computations on Turing machines. Theoret. Comput. Sci. **6** (1978) 1, 1—23.

[19] *Cohen, R. S., A. Y. Gold*: On the complexity of ω-type Turing acceptors. Theoret. Comput. Sci. **10** (1980), 249—272.

[20] *Creutzburg, E., L. Staiger*: Zur Erkennung regelmäßiger Gesetzmäßigkeiten. In: Seminarber. 82 (Teil II), Humboldt-Univ. Berlin, Sekt. Mathematik; Berlin 1986; pp. 342—383.

[21] *Darondeau, Ph., L. Kott*: Towards a formal proof system for ω-rational expressions. Inform. Process. Letters **19** (1984), 173—177.

[22] *Davis, M.*: Infinitary games of perfect information. In: Advances in Game Theory; Princeton Univ. Press, Princeton N.J. 1964; pp. 89—101.

[23] *Eilenberg, S.*: Automata, Languages, and Machines; Vol. A. Academic Press, 1974.

[24] *Elgot, C. C.*: Decision problems of finite automata design and related arithmetics. Trans. Amer. Math. Soc. **98** (1961), 21—51.

[25] *Freund, R.*: Init and Anf operating on ω-languages. Inform. Process. Letters **16** (1983), 265—269.

[26] *Ginsburg, S.*: Algebraic and Automata-theoretic Properties of Formal Languages. North Holland, 1975.

[27] *Ginsburg, S., Sh. Greibach*: Deterministic contextfree languages. Inform. and Control **9** (1966), 620—648.

[28] *Ginsburg, S., Sh. Greibach*: Abstract families of languages. Mem. Amer. Math. Soc. **87** (1969), 1—32.

[29] *Ginsburg, S., T. Hibbard, J. Ullian*: Sequences in context-free languages. Ill. J. Math. **9** (1965), 321—337.

[30] *Girault-Beauquier, D.*: Some results about finite and infinite behaviours of a push-down automaton. In: Automata, Languages and Programming; ed.: *J. Paradaens*; Lect. Notes Comput. Sci., 172; Springer, 1984; pp. 187—195.

[31] *Hartmanis, J., R. Stearns*: Sets of numbers defined by finite automata. Amer. Math. Monthly **74** (1967), 539—542.

[32] *Head, T.*: The topological structure of adherences of regular languages. RAIRO Infor. Théor. Appl. **20** (1986) 1, 31—41.

[33] *Hossley, R.*: Finite tree automata and ω-automata. Ph. D. Diss. MIT, Cambridge, Mass., 1970.

[34] *Istrail, S.*: Some remarks on non-algebraic adherences. Theoret. Comput. Sci. **21** (1982), 341—349.

[35] *Izumi, H., Y. Inagaki, N. Honda*: A complete axiom system of algebra of closed rational expressions. In: Automata, Languages and Programming; ed.: *J. Paradaens*; Lect. Notes Comput. Sci, 172; Springer, 1984; pp. 260—269.

[36] *Johnson, H. R.*: Infinite strings over finite machines. Ph. D. Diss., Univ. of Illinois, Urbana, Ill., 1970.

[37] *Jürgensen, H., H. J. Shyr, G. Thierrin*: Disjunctive ω-languages. Elektron. Inf.-verarb. Kybern. EIK **19** (1983) 6, 267—278.

[38] *Jürgensen, H., G. Thierrin*: On ω-languages whose syntactic monoid is trivial. Intern. J. Comput. Inform. Sci. **12** (1983), 359—365.

[39] *Jürgensen, H., G. Thierrin*: Varieties of monoids and classes of ω-languages. In: Theory of Semigroups; Proc. of the 1984 Conf. Math. Gesellsch. DDR; Berlin 1985; pp. 62—67.

[40] *Jürgensen, H., G. Thierrin*: Which Monoids are Syntactic Monoids of ω-Languages? Elektron. Inf.verarb. Kybern. EIK **22** (1986) 10/11, 513—526.

[41] *Kaminski, M.*: A classification of ω-regular languages. Theoret. Comput. Sci. **36** (1985), 217—229.

[42] *Karpinski, M.*: Almost deterministic ω-automata with existential output condition. Proc. Amer. Math. Soc. **53** (1975), 449—452.

[43] *Kobayashi, K., M. Takahashi, H. Yamasaki*: Characterization of ω-regular languages by first order formulas. Theoret. Comput. Sci. **28** (1984) 3, 315—327.

[44] *Korenjak, A. J., J. E. Hopcroft*: Simple deterministic languages. In: Proc. 7th Ann. IEEE Symp. Switch. and Automata Theory; 1966; pp. 36—46.

[45] *Kuratowski, K.*: Topology; I. Academic Press, 1966.

[46] *Landweber, L. H.*: Decision problems for ω-automata. Math. Syst. Theory **3** (1969) 4, 376—384.

[47] *Lindner, R.*: On the theory of inference operators. In: Proc. Int. Congr. Math.; Vancouver, 1974; pp. 471—475.

[48] *Lindner, R., L. Staiger*: Algebraische Codierungstheorie — Theorie der sequentiellen Codierungen. Akademie-Verlag, Berlin 1977.

[49] *Lindner, R., L. Staiger*: Erkennungs-, maß- und informationstheoretische Eigenschaften regulärer Folgenmengen. Z. Math. Logik Grundl. Math. **23** (1977), 283 to 287.

[50] *Lindsay, P. A.*: Alternation and ω-type Turing acceptors. Theoret. Comput. Sci. **43** (1986) 1, 107—115.

[51] *Linna, M.*: On ω-words and ω-computations. Ann. Univ. Turku, Ser. AI 168, 1975.

[52] *Linna, M.*: On ω-sets associated with context-free languages. Inform. and Control **31** (1976), 272—293.

[53] *Linna, M.*: A decidability result for deterministic ω-context-free languages. Theoret. Comput. Sci. **4** (1977), 83—98.

[54] *McNaughton, R.*: Testing and generating infinite sequences by a finite automaton. Inform. and Control **9** (1966), 521—530.

[55] *Miyano, S., T. Hayashi*: Alternating finite automata on ω-words. Theoret. Comput. Sci. **32** (1984) 3, 321—330.

[56] *Müller, D. E.*: Infinite sequences and finite machines. In: AIEE Proc. 4th Ann. Symp. Switching Theory and Logical Design, Chicago 1963; pp. 3—16.

[57] *Mostowski, A. W.*: Operations on ω-regular languages. In: Fund. Computation Theory; ed.: *M. Karpiński*; Lect. Notes Comput. Sci., 56; Springer, 1977, pp. 135 to 141.

[58] *Nivat, M.*: Mots infinis engendres par une grammaire algebrique. RAIRO Infor. Théor. **11** (1977) ,311—327.

[59] *Nivat, M.*: Sur les ensembles de mots infinis engendres par une grammaire algebrique. RAIRO Infor. Théor. **12** (1978), 259—278.

[60] *Nivat, M.*: Infinite words, infinite trees, infinite computations. Math. Centre Tracts 109, 1979, 1—52.

[61] Automata on Infinite Words. Eds.: *M. Nivat, D. Perrin*. Lect. Notes Comput. Sci., 192. Springer, 1985.

[62] *Park, D.*, Concurrency and automata on infinite sequences. In: Theoretical Computer Science; ed.: *P. Deussen*; Lect. Notes Comput. Sci., 104; Springer, 1981; pp. 167—183.

[63] *Perrin, D.*: Recent results on automata on infinite words. In: Mathematical Foundation of Computer Science; eds.: *M. Chytil, V. Koubek*; Lect. Notes Comput. Sci., 176; Springer, 1984; pp. 134—148.

[64] *Prodinger, M., F. J. Urbanek*: Language operators related to Init. Theoret. Comput. Sci. 8 (1979), 161—165.

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

[66] *Redziejowski, R. R.*: The theory of general events and its application to parallel programming. Tech. Paper TP 18.220, IBM Nordic Lab., Lidingö, Sweden, 1972.

[67] *Redziejowski, R. R.*: Infinite word languages and continuous mappings. Theoret. Comput. Sci. **43** (1986) 1, 59—79.

[68] *Rogers, H.*: Theory of Recursive Functions and Effective Computability. McGraw Hill, 1967.

[69] *Salomaa, A.*: Two complete axiom systems for the algebra of regular events. J. ACM **13** (1966) 1, 158—169.

[70] *Salomaa, A.*: Theory of Automata. Pergamon Press, 1969.
[71] *Salomaa, A.*: Computation and Automata. (Encyclopedia of Mathematics and its Applications, 25). Cambridge Univ. Press, 1985.
[72] *Schnorr, C. P.*: Zufälligkeit und Wahrscheinlichkeit. (Lect. Notes Math., 218). Springer, 1971.
[73] *Siefkes, D.*: Büchi's Monadic Second Order Successor Arithmetic. (Lect. Notes Math., 120). Springer, 1970.
[74] *Siefkes, D.*: The work of *J. Richard Büchi*. In: Proc. AMS Meeting Spring 1985 Chicago (to appear)
[75] *Sipser, M.*: Borel sets and circuit complexity. In: Proc. 15th Ann. Symp. Theory of Computing; 1983; pp. 61–69.
[76] *Staiger, L.*: Über ein Analogon des Satzes von Ginsburg-Rose für sequentielle Folgenoperatoren und reguläre Folgenmengen. Diploma thesis, Friedrich-Schiller-Univ. Jena, 1970.
[77] *Staiger, L.*: Eine Bemerkung zur Charakterisierung von Folgenmengen durch Wortmengen. Elektron. Inf.verarb. Kybern. EIK **8** (1972) 10, 589–592.
[78] *Staiger, L.*: Zur Topologie der regulären Mengen. Diss. A, Friedrich-Schiller-Univ. Jena, 1976.
[79] *Staiger, L.*: Reguläre Nullmengen. Elektron. Inf.verarb. Kybern. EIK **12** (1976) 6, 307–311.
[80] *Staiger, L.*: Empty-storage acceptance of ω-languages. In: Fundamentals of Computation Theory; ed.: *M. Karpiński*; Lect. Notes Comput. Sci., 56; Springer, 1977; pp. 516–521.
[81] *Staiger, L.*: Measure and category in X^ω. In: Proc. 1977 Conf. Topology and Measure II; eds.: *J. Flachsmeyer, Z. Frolik, F. Terpe*; Part 2; Wiss. Beitr. Univ. Greifswald; 1980; pp. 129–136.
[82] *Staiger, L.*: A Note on Connected ω-Languages. Elektron. Inf. verarb. Kybern. EIK **16** (1980) 5/6, 245–251.
[83] *Staiger, L.*: Complexity and entropy. In: Mathematical Foundations of Computer Science; eds.: *J. Gruska, M. Chytil*; Lect. Notes Comput. Sci., 118; Springer, 1981; pp. 508–514.
[84] *Staiger, L.*: Finite-state-ω-languages. J. Comput. System Sci. **27** (1983) 3, 434–448
[85] *Staiger, L.*: Subspaces of $GF(q)^\omega$ and convolutional codes. Inform. and Control **59** (1983) 1–3, 148–183.
[86] *Staiger, L.*: Projection lemmas for ω-languages. Theoret. Comput. Sci. **32** (1984) 3, 331–337.
[87] *Staiger, L.*: The entropy of finite-state ω-languages. Probl. Control and Inform. Theory **14** (1985) 5, 383–392.
[88] *Staiger, L.*: Hierarchies of Recursive ω-languages. Elektron. Inf.verarb. Kybern. EIK **22** (1986) 5/6, 219–241.
[89] *Staiger, L.*: ω-computations on Turing machines and the accepted languages. In: Theory of Algorithms; eds.: *L. Lovász, E. Szemerédi*; Coll. Math. Soc. Janos Bolyai, 44; Budapest 1986; pp. 393–403.
[90] *Staiger, L.*: On infinitary finite length codes. RAIRO Infor. Théor. Appl. **30** (1986) 4, 483–494.
[91] *Staiger, L.*: Sequential mappings of ω-languages. RAIRO Infor. Théor. Appl. **21** (1987) 2, 147–173.
[92] *Staiger, L., W. Nehrlich*: The centers of context-sensitive languages. In: Mathematical Foundations of Computer Science; eds.: *J. Gruska, B. Rovan, J. Wiedermann*; Lect. Notes Comput. Sci., 233; Springer, 1986; pp. 594–601.
[93] *Staiger, L., K. Wagner*: Automatentheoretische und automatenfreie Charakterisierungen topologischer Klassen regulärer Folgenmengen. Elektron. Inf.verarb. Kybern. EIK **10** (1974) 7, 379–392.
[94] *Staiger, L., K. Wagner*: Zur Theorie der abstrakten Familien von ω-Sprachen (ω-AFL). In: Algorithm. Kompl., Lern- und Erkennungsproz. (Proceedings); Jena 1976; pp. 79–91.

[95] *Staiger, L., K. Wagner*: Rekursive Folgenmengen I. Z. Math. Logik. Grundl. Math. **24** (1978), 523—538.

[96] *Takahashi, M., H. Yamasaki*: A note on ω-regular languages. Theoret. Comput. Sci. **23** (1983), 217—225.

[97] *Thomas, W.*: Star-free regular sets of ω-sequences. Inform. and Control **42** (1979), 148—156.

[98] *Thomas, W.*: A combinatorial approach to the theory of ω-automata. Inform. and Control **48** (1981), 261—283.

[99] *Trakhtenbrot, B. A.*: Finite automata and monadic second order logic. Siberian Math. J. **3** (1962), 103—131. (Russian)
English translation in: AMS Transl. **59** (1966), 23—55.

[100] *Trakhtenbrot, B. A., Ja. M. Barzdin*: Finite Automata, Behaviour and Synthesis. Mir Publishers, Moscow 1970. (Russian)
English translation: North Holland, 1973.

[101] *Valk, R.*: Infinite behaviour of Petri nets. Theoret. Comput. Sci. **25** (1983), 311—341.

[102] *Wagner, K.*: Akzeptierbarkeitsgrade regulärer Folgenmengen. Elektron. Inf.verarb. Kybern. EIK **11** (1975) 10/12, 626—630.

[103] *Wagner, K.*: A hierarchy of regular sequence sets. In: Mathematical Foundations of Computer Science; ed.: *J. Bečvář*; Lect. Notes Comput. Sci., 32; Springer, 1975; pp. 445—449.

[104] *Wagner, K.*: Eine Axiomatisierung der Theorie der regulären Folgenmengen. Elektron. Inf.verarb. Kybern. EIK **12** (1976) 7, 337—354.

[105] *Wagner, K.*: Zur Theorie der regulären Folgenmengen. Diss. B, Friedrich-Schiller-Univ. Jena, 1976.

[106] *Wagner, K.*: Eine topologische Charakterisierung einiger Klassen regulärer Folgenmengen. Elektron. Inf.verarb. Kybern. EIK **13** (1977) 9, 473—487.

[107] *Wagner, K.*: Arithmetische Operatoren. Z. Math. Logik Grundl. Math. **22** (1976), 553—570.

[108] *Wagner, K.*: Arithmetische und Bairesche Operatoren. Z. Math. Logik Grundl. Math. **23** (1977), 181—191.

[109] *Wagner, K.*: On ω-regular sets. Inform. and Control **43** (1979), 123—177.

[110] *Wagner, K., L. Staiger*: Recursive ω-languages. In: Fundamentals of Computation Theory; ed.: *M. Karpiński*; Lect. Notes Comput. Sci., 56; Springer, 1977; pp. 532 to 537.

[111] *Wagner, K., G. Wechsung*: Computational Complexity. Deutscher Verlag der Wissenschaften, Berlin 1986.

[112] *Yamasaki, H., M. Takahashi, K. Kobayashi*: Characterization of ω-regular languages by monadic second-order formulas. Theoret. Comput. Sci. **46** (1986) 1, 91—99.

[113] *Latteux, M., E. Timmerman*: Two characterizations of rational adherences. Theoret. Comput. Sci. **46** (1986) 1, 101—106.

[114] *Pecuchet, J.-P.*: On the complementation of Büchi automata. Theoret. Comput. Sci. **47** (1986) 1, 95—98.

[115] *Lindsay, P. A.*: On alternating ω-automata. J. Comput. System Sci. (to appear)

[116] *Latteux, M., E. Timmerman*: Finitely generated ω-languages. Inform. Process. Letters **23** (1986), 171—175.

[117] *Litovsky, I., E. Timmerman*: On generators of rational ω-power languages. Theoret. Comput. Sci. (to appear)

Kurzfassung

Es wird ein Überblick über einige Resultate der Theorie der ω-Sprachen gegeben, wobei besonders topologische und algebraische Aspekte berücksichtigt werden.

Резюме

Дается обзор некоторых результатов теории ω-языков, причем особое внимание уделено топологическим и алгебраическим аспектам.

(Received: first version September 18, 1986,
revised version December 18, 1986)

Author's address:

Dr. L. Staiger
Zentralinstitut für Kybernetik und
Informationsprozesse der AdW
Postfach 1298
1086 Berlin
German Democratic Republic