On the Power of Reading the Whole Infinite Input Tape

Ludwig Staiger*
Martin-Luther-Universität Halle-Wittenberg
Institut für Informatik
Kurt-Mothes-Str. 1
D-06120 Halle (Saale)
Germany

*email: `staiger@informatik.uni-halle.de`

Contents

Finite devices accepting infinite strings are the topic of several papers (see the recent surveys [EH93], [St97] or [Th90, 97]). Most of these papers deal with finite automata. Thus finite automata as devices accepting infinite strings are well-understood. The situation is a little bit more involved if one considers more complicated accepting devices like, e.g., pushdown automata or Turing machines.

A thorough treatment of a general class of accepting devices, so-called X-automata, on infinite words has been given in [EH93]. If one compares the results on the acceptance of infinite strings by Turing machines obtained in the papers [CG78,80], [EH93] and [WS77,SW78] one observes slight differences in the manner how the machines under consideration take into account their behaviour on the input tape.

Type 1 The approach of [WS77,SW78] (cf. also [St86,97]) does not take into consideration the behaviour of the Turing machine on its input tape. Acceptance is based solely on the infinite sequence of internal states the machine runs through during its infinite computation.
Thus the machine may base its decision on a finite part of the whole infinite input.

Type 2 For X-automata Engelfriet and Hoogeboom [EH93] require that, in addition to the fulfillment of certain conditions on the infinite sequence of internal states in order to accept an input, the machine has to read the whole infinite input tape. Cohen and Gold [CG77] considered this same type by for pushdown automata.
Thus, besides blocking[1] as for Type 1, machines have a further possibility to reject inputs.

Type 3 The most complicated type of acceptance for Turing machines was introduced by Cohen and Gold [CG78,80]. They require in addition to Type 2 that the machine scans every cell of the input tape only finitely many times. This behaviour is termed as having a complete non-oscillating run.

The aim of our paper is to explore those sets of infinite words accepted by deterministic Turing machines with respect to Type 2

[1] Most of the six state acceptance conditions considered in Section 2.2 can simulate blocking.

and compare them to the ones accepted with respect to Type 1 which are widely investigated (cf. [CG78][2],[WS77,SW78],[St86,97]). Following the lines of [WS77,SW78],[St86] we provide also characterizations of Type 2 Turing accepted ω-languages in terms of the arithmetical hierarchy and automaton-free descriptions using recursively enumerable languages and limit operations.

Utilizing these characterizations, we show that in case of deterministic Turing machines Type 2-acceptance is strictly more powerful than Type 1-acceptance. The stronger acceptance power of Type 2, however, contrasts with two elegant properties of Type 1-acceptance:
First, the classes of accepted ω-languages coincide fully with classes of the arithmetical hierarchy [WS77,SW78] (cf. also [St86,97]),and secondly, the hierarchy of classes accepted by nondeterministic Turing machines does not collapse.

1 Notation

We start with some necessary notation. By $\mathbb{N} = \{0, 1, 2, \ldots\}$ we denote the set of natural numbers. We consider the space X^ω of infinite strings (sequences, ω-words) on a finite alphabet of cardinality $r := \operatorname{card} X \geq 2$. By X^* we denote the set (monoid) of finite strings (words) on X, including the *empty* word e. For $w \in X^*$ and $b \in X^* \cup X^\omega$ let $w \cdot b$ be their *concatenation*. This concatenation product extends in an obvious way to subsets $W \subseteq X^*$ and $B \subseteq X^* \cup X^\omega$. As usual we denote subsets of X^* as languages and subsets of X^ω as ω-languages.

Furthermore $|w|$ is the *length* of the word $w \in X^*$, hence $X^n = \{w : w \in X^* \wedge |w| = n\}$. By $b[m..n] := b(m) \cdots b(n)$ we denote the *infix* extending from the mth to the nth letter[3] of a string $b \in X^*$, $|b| \geq n$, or $b \in X^\omega$, and $\mathbf{A}(b) := \{b[1..n] : n \in \mathbb{N} \wedge n \leq |b|\}$ and $\mathbf{A}(B) := \bigcup_{b \in B} \mathbf{A}(b)$ are the sets of all finite prefixes of $b \in X^* \cup X^\omega$ and $B \subseteq X^* \cup X^\omega$, respectively.

[2]Although Cohen and Gold [CG78] started their consideration of Turing machines on ω-words with considering the more powerful Type 2-acceptance when turning to the deterministic case they weakened their model stipulating the so-called Property C (continuity property) which requires that the Turing machine under consideration *a priori* has to read the whole input tape in a non-oscillating manner.

[3]If $m > n$ or $m > |b|$ then $b[m..n] := e$, if $n > |b|$ then $b[m..n] := b[m..|b|]$.

As usual we define Π_1*-definable* ω*-languages* $E \subseteq X^\omega$ as

$$E = \{\xi : \forall n(n \in \mathbb{N} \to \xi/n \in W_E)\} \tag{1}$$

where $W_E \subseteq X^*$ is a recursive language, and we define Σ_2*-definable* ω*-languages* $F \subseteq X^\omega$ as

$$F = \{\xi : \exists i(i \in \mathbb{N} \wedge \forall n(n \in \mathbb{N} \to (i, \xi/n) \in M_F))\} \tag{2}$$

where M_F is a recursive subset of $\mathbb{N} \times X^*$. Σ_1-definable and Π_2-definable ω-languages are defined accordingly. As usual, we call a language Σ_1*-definable* iff it is recursively enumerable, and we call a language $W \subseteq X^*$ Π_2*-definable* iff

$$W = \{w : w \in X^* \wedge \forall i(i \in \mathbb{N} \to \exists n(n \in \mathbb{N} \wedge (i, n, w) \in M_W))\}$$

for some recursive subset $M_W \subseteq \mathbb{N} \times \mathbb{N} \times X^*$.

2 Turing machines

In order to be in accordance with the X-automata of [EH93] we consider Turing machines $\mathcal{M} = (X, \Gamma, Z, z_0, R)$ with a separate input tape on which the read-only-head moves only to right, n working tapes, X as its input alphabet, Γ as its worktape alphabet, Z the finite set of internal states, z_0 the initial state, and the relation

$$R \subseteq Z \times X \times \Gamma^n \times Z \times \{0, +1\} \times (\Gamma \times \{-1, 0, +1\})^n$$

defining the next configuration.

Here $(z, x_0, x_1, \ldots, x_n; z', y_0, y_1, \ldots, y_n) \in R$ means that when $\mathcal{M}$ is in state $z \in Z$, reads $x_0 \in X$ on its input tape and $x_i \in \Gamma$ on its worktapes $(i \in \{1, \ldots, n\})$, $\mathcal{M}$ changes its state to $z' \in Z$, moves its head on the input tape to the right if $y_0 = +1$ or if $y_0 = 0$ does not move the head, and for $(i \in \{1, \ldots, n\})$ and $y_i = (x_i', m_i)$ with $x_i' \in \Gamma$ and $m_i \in \{-1, 0, +1\}$ the machine $\mathcal{M}$ writes x_i' instead of x_i in its i-th worktape and moves the head on this tape to the left, if $m_i = -1$, to the right, if $m_i = +1$, or does not move it, if $m_i = 0$.

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 $(z, x_0, x_1, \ldots, x_n)$ in $Z \times X^{n+1}$ at least one[4] move $(z, x_0, x_1, \ldots, x_n; z, y_0, y_1, \ldots, y_n)$.

[4]exactly one, if the device is deterministic

For a detailed description of configurations (instantaneous descriptions) and the behaviour (computing process) of Turing machines the reader is referred to the literature (e.g. [Ro67]).

To avoid clumsy notation, we will describe the construction of machines and their behaviour only in an informal manner leaving details to the reader.

Let the input of the Turing machine be some sequence $\xi \in X^\omega$. We call a sequence $\zeta \in Z^\omega$ of states a *run* of $\mathcal{M}$ on ξ if ζ is the sequence of states the Turing machine runs through in its (some of its, if the machine is nondeterministic) computation with input ξ.

2.1 Acceptance without state conditions

In this section we are interested in the question which ω-languages can be accepted by Turing machines when we disregard the internal behaviour (states) and consider only the reading behaviour of the head on the input tape.

We say that an input word $w \in X^*$ is *read in full length* by a Turing machine $\mathcal{T}$ provided w is the empty word e or if $w = x_1 \cdot \ldots \cdot x_\ell$ then $\mathcal{T}$ starting on the leftmost letter x_1 moves its input head at least once onto the last letter x_ℓ.

Let $\mathcal{T}$ be a Turing machine. Define the function $\varphi : X^* \times \mathbb{N} \to X^*$ in the following way

$$\varphi(w,n) := \begin{cases} w[1..n], & \text{if } |w| \geq n \text{ and } \mathcal{T} \text{ reads at least the first} \\ & n \text{ letters of } w, \text{ and} \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

Obviously, φ is a partial recursive function. Hence $\varphi(X^* \times \mathbb{N})$ is a recursively enumerable language.

Thus we have the following relationship between prefix-closed, that is, $W = \mathbf{A}(W)$, recursively enumerable languages and Turing machines reading words in full length.

Lemma 1 *A language $W \subseteq X^*$ is prefix-closed and recursively enumerable iff there is a Turing machine $\mathcal{T}$ such that W is the set of all input words which $\mathcal{T}$ reads in full length.*

We say that an ω-word $\xi \in X^\omega$ is $\lim$-*accepted* by a Turing machine $\mathcal{M}$ if an only if there is a run of $\mathcal{M}$ on input ξ such that $\mathcal{M}$ reads every symbol of ξ, and we refer to an ω-language $F \subseteq X^\omega$ as

$\lim$-*accepted* provided there is a deterministic Turing machine $\mathcal{M}$ such that $F = \{\xi : \xi \in X^\omega \wedge \xi$ is $\lim$-accepted by $\mathcal{M}\}$.

We obtain the following characterization of the of the class of $\lim$-accepted ω-languages.

Theorem 2 *An ω-language $F \subseteq X^\omega$ is $\lim$-accepted iff there is a recursively enumerable language $W \subseteq X^*$ such that $F = \{\zeta : \mathbf{A}(\zeta) \subseteq \mathbf{A}(W)\}$.*

Proof. Let F be $\lim$-accepted by a Turing machine $\mathcal{M}$. Then from the above lemma we know that the language $L_{\mathcal{M}}$ of all words $w \in X^*$ which are read in full length by $\mathcal{M}$ is recursively enumerable. Thus $F = \{\xi : \mathbf{A}(\xi) \subseteq \mathbf{A}(L_{\mathcal{M}})\}$.

Conversely, let $F = \{\xi : \mathbf{A}(\xi) \subseteq \mathbf{A}(W)\}$ for some recursively enumerable language W. Then $\mathbf{A}(W)$ is also recursively enumerable, and utilizing a Turing machine $\mathcal{T}$ which enumerates the language $\mathbf{A}(W)$ it is easy to construct a Turing machine $\mathcal{M}$ which for an input word $v \in X^*$ reads only the longest prefix $w \sqsubseteq v,\ w \in \mathbf{A}(W)$. From the behaviour of $\mathcal{M}$ it follows that $\mathcal{M}$ $\lim$-accepts F. ❑

In [St86] it is shown that the condition derived in Theorem 2 has several equivalent formulations in terms of the arithmetical hierarchy and topology.

To this aim we consider X^ω as a topological space with the basis $(w \cdot X^\omega)_{w \in X^*}$. Since X is finite, this topological space is homeomorphic to the Cantor discontinuum, hence compact. We mention here still that a subset F in X^ω is closed if and only if $F = \{\xi : \mathbf{A}(\xi) \subseteq \mathbf{A}(F)\}$. For more details the interested reader is referred to [St97, Section 2].

Lemma 3 ([St86]) *Let $F \subseteq X^\omega$. Then the following conditions are equivalent.*

1. *There is a recursively enumerable language $W' \subseteq X^*$ such that $F = \{\zeta : \mathbf{A}(\zeta) \subseteq W'\}$,*
2. *There is a recursively enumerable language $W \subseteq X^*$ such that $F = \{\zeta : \mathbf{A}(\zeta) \subseteq \mathbf{A}(W)\}$,*
3. *There is a recursive language $V \subseteq X^*$ such that $F = \{\zeta : \mathbf{A}(\zeta) \subseteq \mathbf{A}(V)\}$,*
4. *There is a Π_2-definable language U such that $F = \{\zeta : \mathbf{A}(\zeta) \subseteq U\}$, and*

5. *F is closed in X^ω and $\mathbf{A}(F)$ is Π_2-definable.*

In the sequel we will refer to the class of $\lim$-accepted ω-languages as

$$\mathfrak{P} := \{F : F \subseteq X^\omega \wedge F \text{ is closed } \wedge \mathbf{A}(F) \text{ is } \Pi_2\text{-definable}\} . \qquad (3)$$

We mention still some closure properties of the class $\mathfrak{P}$. To this end we consider homomorphisms $h : X \to X^*$. We extend the homomorphism h in the usual way to words. For ω-words $\xi \in X^\omega$ we define the extension of h, $\overline{h}$, in the following way:
Let $\overline{X} := \{x : x \in X \wedge h(x) = e\}$ be the set of letters erased by the homomorphism h.

$$\overline{h}(\xi) := \begin{cases} \text{undefined}, & \text{if } \xi \in X^* \cdot \overline{X}^\omega; \\ \lim_{w \to \xi} h(w), & \text{otherwise.} \end{cases}$$

Lemma 4 ([St86]) *The class $\mathfrak{P}$ is closed under union, intersection, non-erasing homomorphisms and their inverse mappings.*

2.2 State-acceptance conditions for Turing machines

In this part we define the six conditions imposed on the runs of Turing machines in order to specify a set of input ω-words (cf. [CG78][5], [EH93], [St97], [SW78]).

We say that an input sequence $\xi \in X^\omega$ is accepted by $\mathcal{M}$ according to condition (mode) Ξ if there is a run ζ of $\mathcal{M}$ on ξ such that ζ satisfies Ξ. In the sequel we shall consider the following conditions using the notation of [EH93]:

Let $\alpha : Z^\omega \to 2^Z$ be a mapping which assigns to every ω-word $\zeta \in Z^\omega$ a subset $Z' \subseteq Z$, and let $R \subseteq 2^Z \times 2^Z$ be a relation between subsets of Z. We say that a pair $(\mathcal{M}, \mathcal{Z})$ where $\mathcal{Z} \subseteq 2^Z$ accepts an ω-word $\xi \in X^\omega$ if and only if

$$\exists Z' \exists \zeta (Z' \in \mathcal{Z} \wedge \zeta \text{ is a run of } \mathcal{M} \text{ on } \xi \wedge (\alpha(\zeta), Z') \in R) .$$

Here we shall be concerned with the following mappings α and relations R: For an ω-word $\zeta \in Z^\omega$ let $ran(\zeta) := \{z : z \in Z \wedge \exists i (i \in \mathbb{N} \setminus \{0\} \wedge \zeta(i) = z)\}$ be the *range* of ζ (considered as a mapping $\zeta : \mathbb{N} \setminus \{0\} \to Z$), that is, the set of all letters occurring in ζ, and let

[5] Cohen and Gold consider only the five conditions introduced by Landweber [La69].

$inf(\zeta) := \{z : z \in Z \wedge \zeta^{-1}(z)$ is infinite$\}$ be the *infinity set* of ζ, that is, the set of all letters occurring infinitely often in ζ.

As relations R we shall use $=$, $\subseteq$ and $\sqcap$ where $Z' \sqcap Z'' :\Leftrightarrow Z' \cap Z'' \neq \emptyset$.

We obtain the six types of acceptance presented in the following table. For the sake of completeness we add a simple description and their correspondence to the five types originally defined by Landweber [La69] and used in [CG78,80].

$(ran, \sqcap)$	1-acceptance	at least once
$(ran, \subseteq)$	$1'$-acceptance	everywhere
$(ran, =)$		
$(inf, \sqcap)$	2-acceptance	infinitely often
$(inf, \subseteq)$	$2'$-acceptance	almost everywhere
$(inf, =)$	3-acceptance	

3 Classes of ω-languages accepted by deterministic Turing machines

In this section we consider the ω-languages defined by Turing machines according to Types 1 and 2 defined above and the six accepting conditions introduced in the previous section. First we give a brief review on the results concerning the six classes accepted according to Type 1 obtained in [WS77,SW78] or [St86]. Then we derive a relationship between the Type 1- and Type 2-classes based on the results of Section 2.1. Here it turns out that the class of $\lim$-accepted ω-languages plays a crucial role. Finally, we use this relationship to obtain automata-free characterizations of the classes of ω-languages accepted according to Type 2. These descriptions yield immediately that Type 2-acceptance is, in some cases, strictly more powerful than Type 1-acceptance, whereas the proper Type 2-classes neither do coincide with classes of the Arithmetical hierarchy, as the Type 1 classes do, nor do they enjoy the same closure properties.

3.1 ω-languages accepted according to Type 1

We start with some characterizations in terms of recursive or recursively enumerable languages of the low level classes of the Arithmetical hierarchy of ω-languages (cf. [St86,97]).

Lemma 5

1. *A set $E \subseteq X^\omega$ is in Σ_1-definable iff there is a recursive (or, equivalently, a recursively enumerable) language $W \subseteq X^*$ such that $E = W \cdot X^\omega$.*

2. *A set $F \subseteq X^\omega$ is in Π_1-definable if and only if F is closed in X^ω and $X^* \setminus \mathbf{A}(F)$ is recursively enumerable, that is, $\mathbf{A}(F) \subseteq X^*$ is Π_1-definable.*

3. *A set $E \subseteq X^\omega$ is in Π_2-definable iff there is a recursive (or, equivalently, a recursively enumerable) language $W \subseteq X^*$ such that $E = \{\xi : \xi \in X^\omega \wedge \mathbf{A}(\xi) \cap W$ is infinite$\}$.*

We mention still the following well-known closure properties of the classes of the Arithmetical hierarchy, analogously to the ones in Lemma 4.

Lemma 6

1. *Each one of the classes $\mathbf{\Sigma}_1, \mathbf{\Pi}_1$ and $\mathbf{\Sigma}_2$ of Σ_1-definable, Π_1-definable, or Σ_2-definable subsets of X^ω, respectively, is closed under union, intersection, non-erasing homomorphisms and inverse non-erasing homomorphisms.*

2. *The class $\mathbf{\Pi}_2$ of Π_2-definable subsets of X^ω is closed under union, intersection, and inverse non-erasing homomorphisms, but not under non-erasing homomorphisms.*

The class of all ω-languages $F \subseteq X^\omega$ Type 1-accepted by deterministic (nondeterministic) Turing machines with respect to mode (α, R) is denoted by $\mathbf{DT}^{(X)}(\alpha, R)$ (or $\mathbf{NT}^{(X)}(\alpha, R)$, respectively). The following characterization of the six **DT**-classes is shown in [WS77,SW78] (cf. also [St86,97]). We start with the simple modes $\mathbf{DT}^{(X)}(\alpha, \subseteq)$ and $\mathbf{DT}^{(X)}(\alpha, \sqcap)$.

Theorem 7

$$
\begin{array}{rcl}
\mathbf{DT}^{(X)}(ran, \subseteq) & = & \mathbf{\Pi}_1 = \{F : F \subseteq X^\omega \wedge F \text{ is } \Pi_1\text{-definable}\} \\
\mathbf{DT}^{(X)}(ran, \sqcap) & = & \mathbf{\Sigma}_1 = \{F : F \subseteq X^\omega \wedge F \text{ is } \Sigma_1\text{-definable}\} \\
\mathbf{DT}^{(X)}(inf, \subseteq) & = & \mathbf{\Sigma}_2 = \{F : F \subseteq X^\omega \wedge F \text{ is } \Sigma_2\text{-definable}\} \\
\mathbf{DT}^{(X)}(inf, \sqcap) & = & \mathbf{\Pi}_2 = \{F : F \subseteq X^\omega \wedge F \text{ is } \Pi_2\text{-definable}\}
\end{array}
$$

Since Turing machines allow for a parallel composition, we obtain a characterization for the classes $\mathbf{DT}^{(X)}(\alpha, =)$. Here $\mathcal{B}(\mathcal{S})$ denotes the closure of the set $\mathcal{S} \subseteq 2^{X^\omega}$ under Boolean operations.

Corollary 8

$$\begin{aligned}\mathbf{DT}^{(X)}(ran, =) &= \mathcal{B}(\mathbf{\Pi}_1) &= \mathcal{B}(\mathbf{\Sigma}_1)\\ \mathbf{DT}^{(X)}(inf, =) &= \mathcal{B}(\mathbf{\Sigma}_2) &= \mathcal{B}(\mathbf{\Pi}_2)\end{aligned}$$

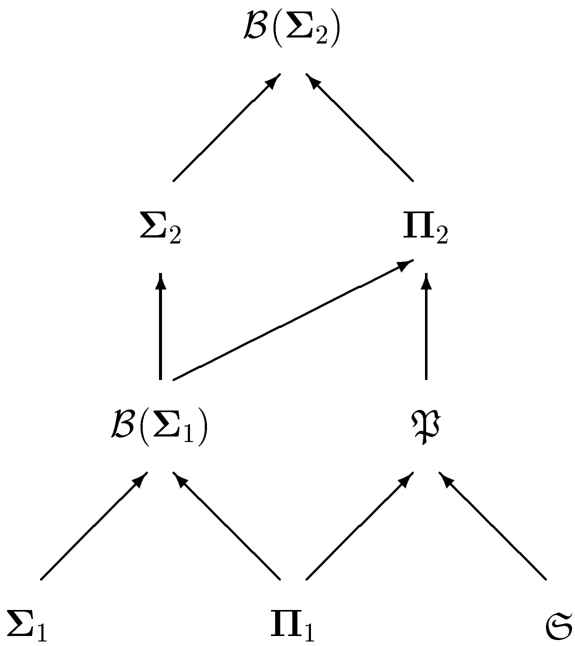

Figure 1: Inclusion relations between various classes of ω-languages accepted by Turing machines

Figure 1 shows the relations between the six classes of Theorem 7 and Corollary 8, the class $\mathfrak{P}$ (cf. Eq. (3)) and its related subclass

$$\mathfrak{S} := \{F : F \subseteq X^\omega \wedge F \text{ is closed } \wedge \mathbf{A}(F) \text{ is recursively enumerable}\}\ .$$

Example 1.15 of [St93] shows that $\mathfrak{S}$ is incomparable to the classes of Π_1-definable, Σ_1-definable and Σ_2-definable subsets of X^ω. Thus, the inclusion properties presented in Figure 1 are proper and other inclusions than the ones represented do not exist.

3.2 Composition and Decomposition Theorem

In this section we analyze the behaviour of Turing machines having the additional possibility to reject by not reading the whole input. We obtain a characterization in terms of the class of lim-accepted ω-languages and the classes accepted by Turing machines not regarding the moving of the input head.

We agree on the following notation: By $T(\mathcal{M}', \mathcal{Z})$ we denote the ω-language accepted by the pair $(\mathcal{M}, \mathcal{Z})$ according to Type 1, and by $\lim\text{-}T(\mathcal{M}, \mathcal{Z})$ the ω-language accepted by the pair $(\mathcal{M}, \mathcal{Z})$ according to Type 2, and $\lim\text{-}T(\mathcal{M})$ is the ω-language lim-accepted by $\mathcal{M}$..

We obtain the following easily verified decomposition theorem.

Theorem 9 (Decomposition Theorem) *Let $\mathcal{M} = (X, Z, \Gamma, z_0, f)$ be a deterministic Turing machine and let $\mathcal{Z} \subseteq 2^Z$. Then for $\alpha \in \{ran, inf\}$ and $R \in \{\subseteq, \sqcap, =\}$ we have $\lim\text{-}T^{\alpha}_{R}(\mathcal{M}, \mathcal{Z}) = T^{\alpha}_{R}(\mathcal{M}, \mathcal{Z}) \cap \lim\text{-}T(\mathcal{M})$.*

In some sense we can also prove the converse of the Decomposition Theorem.

Theorem 10 (Composition Theorem) *Let $\mathcal{M} = (X, Z, \Gamma, z_0, f), \mathcal{Z} \subseteq 2^Z$, and $\mathcal{M}' = (X, S, \Gamma', s_0, g)$ be deterministic Turing machines. Then there is a deterministic Turing machine $\mathcal{M}''$ such that for $\alpha \in \{ran, inf\}$ and $R \in \{\subseteq, \sqcap, =\}$ it holds $\lim\text{-}T^{\alpha}_{R}(\mathcal{M}'', \mathcal{Z}_{\alpha,R}) = T^{\alpha}_{R}(\mathcal{M}, \mathcal{Z}) \cap \lim\text{-}T(\mathcal{M}')$ for suitable $\mathcal{Z}_{\alpha,R}$.*

Proof. We give an informal description of how to construct an appropriate Turing machine $\mathcal{M}''$. $\mathcal{M}''$ is a parallel composition of $\mathcal{M}$ and $\mathcal{M}'$ in the following sense:
$\mathcal{M}''$ consists of one copy of $\mathcal{M}$ and $\mathcal{M}'$ each and an additional copying and step counting control.

$\mathcal{M}''$ starts working as $\mathcal{M}'$. As soon as $\mathcal{M}'$ reads a new letter from the input tape this letter is copied to an internal tape serving as the input tape of $\mathcal{M}$. Then $\mathcal{M}''$ switches to $\mathcal{M}$, resets the input head of $\mathcal{M}$ to the situation it had before copying and then simulates the work of $\mathcal{M}$ until $\mathcal{M}$ requires an input symbol not yet copied to its input tape or otherwise the number of steps given by the step counting control is exhausted. This step counting control ensures $\mathcal{M}''$ to switch to simulate $\mathcal{M}'$ even in case $\mathcal{M}$ does not require further input letters. ❑

As an immediate consequence of Theorems 9 and 10 we obtain that the classes $\overline{\mathbf{DT}}^{(X)}(\alpha, R)$ of ω-languages accepted by fulfilling the acceptance condition (α, R) *and* reading the whole input can be described as follows.

Theorem 11

$$\begin{array}{lcl}
\overline{\mathbf{DT}}^{(X)}(ran, \subseteq) & = & \mathfrak{P} = \{F : F \textit{ is closed } \wedge \mathbf{A}(F) \textit{ is } \Pi_2\textit{-definable}\} \\
\overline{\mathbf{DT}}^{(X)}(ran, \sqcap) & = & \{E \cap F : E \textit{ is } \Sigma_1\textit{-definable} \wedge F \in \mathfrak{P}\} \\
\overline{\mathbf{DT}}^{(X)}(ran, =) & = & \{E \cap F : E \in \mathbf{DT}^{(X)}(ran, =) \wedge F \in \mathfrak{P}\} \\
\overline{\mathbf{DT}}^{(X)}(inf, \subseteq) & = & \{E \cap F : E \textit{ is } \Sigma_2\textit{-definable} \wedge F \in \mathfrak{P}\} \\
\overline{\mathbf{DT}}^{(X)}(inf, \sqcap) & = & \mathbf{DT}^{(X)}(inf, \sqcap) = \{E : E \textit{ is } \Pi_2\textit{-definable}\} \\
\overline{\mathbf{DT}}^{(X)}(inf, =) & = & \mathbf{DT}^{(X)}(inf, =)
\end{array}$$

Proof. The first four identities are obvious, and the last two follow from the fact that every ω-language $F \in \mathfrak{P}$ is Π_2-definable, and the class of Π_2-definable ω-languages is closed under intersection and union. ❑

From the results presented in Figure 1 we obtain the following inclusion relations involving the new classes and the ones described above.

Lemma 12

$$\begin{array}{ccccccc}
\mathfrak{P} & \subset & \overline{\mathbf{DT}}^{(X)}(ran, \sqcap) & \subset & \overline{\mathbf{DT}}^{(X)}(ran, =) & \subset & \mathbf{\Pi}_2 \\
\mathbf{\Sigma}_2 & \subset & \overline{\mathbf{DT}}^{(X)}(inf, \subseteq) & \subset & \mathcal{B}(\mathbf{\Sigma}_2) & &
\end{array}$$

In contrast to Lemma 6 the new classes $\overline{\mathbf{DT}}^{(X)}(ran, \sqcap)$, $\overline{\mathbf{DT}}^{(X)}(ran, =)$ and $\overline{\mathbf{DT}}^{(X)}(inf, \subseteq)$, however, are not closed under union, they are only closed under intersection. We provide an example.

Example 1 Let $X := \{a, b, c\}$. Take an ω-language $F_1 \subset \{a, b\}^\omega$, $F_1 \in \mathfrak{P}$ which is not Σ_2-definable and let $F_2 := \{a, b\}^* \cdot c \cdot \{a, b, c\}^\omega$ which is Σ_2-definable, even Σ_1-definable, but not closed in X^ω. Then $F_1 \in \mathfrak{P} \subseteq \overline{\mathbf{DT}}^{(X)}(ran, \sqcap)$ and, since F_2 is Σ_1-definable, we have also $F_2 \in \overline{\mathbf{DT}}^{(X)}(ran, \sqcap)$. Assume now $F_1 \cup F_2 \in \overline{\mathbf{DT}}^{(X)}(inf, \subseteq)$.

The smallest closed subset $F \subseteq \{a, b, c\}^\omega$ containing $F_1 \cup F_2$ is $\{a, b, c\}^\omega$ itself. Hence, $F_1 \cup F_2 = E \cap F$ with $E \in \mathbf{\Sigma}_2$ and F closed in X^ω implies $F = \{a, b, c\}^\omega$, whence $F_1 \cup F_2 = E \in \mathbf{\Sigma}_2$. But this contradicts the fact that $(F_1 \cup F_2) \cap \{a, b\}^\omega = F_1 \notin \mathbf{\Sigma}_2$. ❑

4 Nondeterministic Turing machines

The classes of ω-languages accepted by nondeterministic Turing machines can be obtained from the deterministic classes via Projection lemmas for Type 1-acceptance and via the Homomorphism lemma [EH93, Lemma 3.11].

Let X, Y be finite alphabets, and let $\overline{pr}_1 : X \times Y \to X$ be a projection.[6]

Lemma 13 (Projection lemma for Turing machines [WS77,SW78])

$F \in \mathbf{NT}^{(X\times Y)}(\alpha, R)$ implies that $\overline{pr}_1(F) \in \mathbf{NT}^{(X)}(\alpha, R)$.
Conversely, for every $E \in \mathbf{NT}^{(X)}(\alpha, R)$ there is an $F \in \mathbf{DT}^{(X\times Y)}(\alpha, R)$ such that $E = \overline{pr}_1(F)$.

This lemma yields in an elegant way an immediate characterization of the ω-languages Type 1 accepted by nondeterministic Turing machines. Because Lemma 6 shows that the classes $\mathbf{DT}^{(X)}(ran, \subseteq)$, $\mathbf{DT}^{(X)}(ran, \sqcap)$ and $\mathbf{DT}^{(X)}(inf, \subseteq)$ are closed under projections, according to the projection lemma they coincide with their nondeterministic counterparts.

The nondeterministic counterparts of the other classes are obtained as follows.

Theorem 14

$$\begin{array}{lclcl}\mathbf{NT}^{(X)}(ran, =) & = & \mathbf{NT}^{(X)}(inf, \subseteq) & = & \mathbf{DT}^{(X)}(inf, \subseteq) \\ \mathbf{NT}^{(X)}(inf, \sqcap) & = & \mathbf{NT}^{(X)}(inf, =) & = & \{F : F \subseteq X^\omega \wedge F \textit{ is } \Sigma_1^1\textit{-definable}\}\end{array}$$

Here, as usual, an ω-language is referred to as Σ_1^1-definable provided

$$F = \{\xi : \exists\eta(\eta \in X^\omega \wedge \forall n\exists m\,((n, \eta[1..m], \xi[1..m]) \in M_F))\}\ ,$$

for some recursive relation $M_F \subseteq \mathbb{N} \times X^* \times X^*$. Observe that the class of Σ_1^1-definable ω-languages equals the class of projections of Π_2-definable ω-languages.

Turning to Type 2 acceptance we have the following general relationship between the determninistic and nondeterministic classes.

[6]It is essential here that the mapping $\overline{pr}_1$ is a non-erasing homomorphism, whereas in Lemma 15 the homomorphism may erase letters.

Lemma 15 (Homomorphism lemma for Turing machines [EH93])

$F \in \overline{\mathbf{NT}}^{(X)}(\alpha, R)$ implies that $\overline{h}(F) \in \overline{\mathbf{NT}}^{(Y)}(\alpha, R)$.
Conversely, for every $E \in \overline{\mathbf{NT}}^{(Y)}(\alpha, R)$ there is an $F \in \overline{\mathbf{DT}}^{(X)}(\alpha, R)$ such that $E = \overline{h}(F)$.

Since a Turing machine can simulate even a counter, Theorem 3.11 of [EH93] proves that the hierarchy of classes $\overline{\mathbf{NT}}^{(X)}(\alpha, R)$ collapses. Using recursion-theoretic arguments one obtains.

Theorem 16 *For all $\alpha \in \{ran, inf\}$ and all $R \in \{\subseteq, \sqcap, =\}$ the class $\overline{\mathbf{NT}}^{(X)}(\alpha, R)$ coincides with the class of Σ_1^1-definable ω-languages over X.*

Thus, similar to the case of projections, where the class of Σ_1^1-definable ω-languages over X can be obtained from the class of of Π_2-definable ω-languages over X via projections, all Σ_1^1-definable ω-languages over X are homomorphic images of (closed) ω-languages in $\mathfrak{P}$.

We can prove even more.

Lemma 17 *For every Σ_1^1-definable $F \subseteq X^\omega$ there are a Π_1-definable $E \subseteq Y^\omega$ and a homomorphism $h : Y \to X$ such that $F = \overline{h}(E)$.*

Proof. Since every Σ_1^1-definable ω-language is the projection (image under non-erasing homomorphism) of a Π_2-definable ω-language, it suffices to prove that every Π_2-definable $F \subseteq X^\omega$ is the image under arbitrary homomorphism of a Π_1-definable $E \subseteq Y^\omega$.

To this aim let $Y := X \cup \{d\}$ where d is a letter not in X. We define $h : X \cup \{d\} \to X \cup \{e\}$ to be the homomorphism erasing the letter d, that is,

$$h(y) = \begin{cases} y & \text{, if } y \in X \text{ , and} \\ e & \text{, if } y = d \text{ .} \end{cases}$$

We use Lemma 5 to obtain e recursive language $W \subseteq X^*$ such that $F = \{\xi : \xi \in X^\omega \wedge \mathbf{A}(\xi) \cap W$ is infinite$\}$.

Next we pad the language W by inserting strings of ds in such a way that we can reconstruct from every padded word all (proper) prefixes of the original word:

Define for $w = w_1 \cdot w_2 \cdots w_n \in W$ where $\mathbf{A}(w) \cap W \setminus \{e\} = \{w_1 \cdot w_2 \cdots w_i : 1 \leq i \leq n\}$ its *padding* as follows.

$$\begin{aligned} \pi(w) &:= \{d^{|w_1|} \cdot w_1 \cdot d^{|w_2|} \cdot w_2 \cdots w_{n-1} \cdot d^{|w_n|} \cdot w'_n : w'_n \sqsubseteq w_n\} \cup \\ &\quad \cup \{d^{|w_1|} \cdot w_1 \cdot d^{|w_2|} \cdot w_2 \cdots w_{n-1} \cdot d^l : l \leq |w_n|\}, \text{ and let} \\ \pi(W) &:= \bigcup_{w \in W} \pi(w) . \end{aligned}$$

Observe that $\pi(W)$ is a recursive language provided W is recursive, and, moreover, $\pi(W) = \mathbf{A}(\pi(W))$ if $W \neq \{e\}$.

Let $E := \{\eta : \mathbf{A}(\eta) \subseteq \pi(W)\} = \{\eta : \mathbf{A}(\eta) \cap \pi(W)$ is infinite$\}$.

By definition $\overline{h}(\eta)$ is defined if η contains infinitely many prefixes ending with a letter of X. Thus, $\overline{h}(\eta)$, $\eta \in E$ is defined iff η has infinitely many prefixes of the form $d^{|w_1|} \cdot w_1 \cdot d^{|w_2|} \cdot w_2 \cdots d^{|w_n|} \cdot w'_n$ where $e \neq w'_n \in X^*$. From this we can conclude that $\overline{h}(\eta)$ has infinitely many prefixes in W, that is, $\overline{h}(\eta) \in F$.

Conversely, let $\xi \in F$. Then $\xi = w_1 \cdot w_2 \cdots w_i \cdots$ where $\mathbf{A}(\xi) \cap W \setminus \{e\} = \{w_1 \cdot w_2 \cdots w_i : i \in \mathbb{N} \setminus \{0\}\}$. Then $\eta := d^{|w_1|} \cdot w_1 \cdot d^{|w_2|} \cdot w_2 \cdots d^{|w_i|} \cdot w_i \cdots \in E$ and $\overline{h}(\eta) = \xi$. ❑

Reading the previous proof more carefully, we observe that, since $\pi(W)$ is recursive, $E := \{\eta : \mathbf{A}(\eta) \subseteq \pi(W)\}$ is not only Π_1-definable, but also in $\mathfrak{S}$, and we obtain the following corollary showing the power of erasing.

Corollary 18 *For every Σ_1^1-definable $F \subseteq X^\omega$ there are an ω-language $E \subseteq Y^\omega$, $E \in \mathbf{\Pi}_1 \cap \mathfrak{S}$ and a homomorphism $h : Y \to X$ such that $F = \overline{h}(E)$.*

References

[CG77] R.S. Cohen and A.Y. Gold, Theory of ω-languages I: Characterizations of ω-context-free languages, and II: A study of various models of ω-type generation and recognition, J. Comput. System Sci. 15 (1977)2, 169–184 and 185–208.

[CG78] R.S. Cohen and A.Y. Gold, ω-computations on Turing machines, Theoret. Comput. Sci. 6 (1978), 1–23.

[CG80] R.S. Cohen and A.Y. Gold, On the complexity of ω-type Turing acceptors, Theoret. Comput. Sci. 10 (1980), 249–272.

[EH93] J.Engelfriet and H.J. Hoogeboom, X-automata on ω-words. Theoret. Comput. Sci. 110 (1993) 1, 1–51.

[La69] L.H. Landweber, Decision problems for ω-automata, Math. Syst. Theory 3 (1969) 4, 376–384.

[Ro67] H. Rogers, Theory of Recursive Functions and Effective Computability, McGraw Hill, New York 1967.

[St86] L. Staiger, Hierarchies of recursive ω-languages. J. Inform. Process. Cybernetics EIK 22 (1986) 5/6, 219–241.

[St97] L. Staiger, ω-languages. in: *Handbook of Formal Languages*, Vol. 3, G. Rozenberg and A. Salomaa (Eds.), Springer-Verlag, Berlin. 339–387.

[SW78] L. Staiger und K. Wagner, Rekursive Folgenmengen I. Zeitschr. Math. Logik u. Grundl. Mathematik 24 (1978) 6, 523–538.

[Th90] W. Thomas, Automata on Infinite Objects, in: *Handbook of Theoretical Computer Science*, Vol. B, J. Van Leeuwen (Ed.), Elsevier, Amsterdam, 1990. 133–191.

[Th97] W. Thomas, Languages, automata, and logic, in: *Handbook of Formal Languages*, Vol. 3, G. Rozenberg and A. Salomaa (Eds.), Springer-Verlag, Berlin. 389–455.

[WS77] K. Wagner and L. Staiger, Recursive ω-languages. in: *Fundamentals of Computation Theory*, M. Karpiński (Ed.), Lect. Notes Comput. Sci. 56, Springer Verlag, Berlin 1977, 532–537.