ON A POSSIBLE CLASSIFICATION OF REAL-TIME CONSTRUCTED SEQUENCES

Noëlle BLEUZEN-GUERNALEC
Département de Mathématiques, Faculté des Sciences de Luminy, 70 Route Léon-Lachamp, 13288 Marseille Cedex 9, France

Received October 1988

Abstract. We consider two approaches of infinite sequences over a finite alphabet:
- concerning complexity of production, real-time productibility is a “good” criterion;
- concerning complexity of structure, iterative productibility is particularly interesting.

Different modes of iteration, all of them being: first an automatic “fixed-point generation” and then some “encoding”, allow us to build a hierarchy in real-time constructible sequences.

Contents

1. Introduction ................................................ 143
2. Real-time sequences ........................................ 144
3. Iterative production ....................................... 144
   3.1. Example of a fixed point obtained by the action of a transducer 145
   3.2. Some results ......................................... 146
4. Conclusion ................................................ 146
References .................................................... 147

1. Introduction

The study of infinite sequences over a finite alphabet is interesting not only in Number Theory but also in Theoretical Computer Science, in Biology. Therefore it seems important to try and find an easy way
- of producing such sequences,
- of studying their structure.

We shall consider two approaches to these sequences:
- complexity of production (computational time): from this point of view, simplicity is real-time productibility;
- complexity of structure: in that case an easy approach is the iterative nature of production.
Different modes of iteration will allow us to build a hierarchy in real-time constructible sequences.

2. Real-time sequences

The notion of "complexity of algorithms" partially comes from problems in Number Theory. The main concepts were formalized in 1963 by Hartmanis and Stearns [1] and these concepts made new types of conjectures possible in Number Theory. For example,
- we know that rational numbers are real-time constructible,
- it has been proved that algebraic numbers are quadratic-time constructible,
- some transcendental numbers are known to be real-time constructible.

**Conjecture.** No algebraic irrational number is real-time constructible.

The following can easily be formalized in terms of Turing programs:
- For a constructible (i.e. automatic) sequence \( s \), and a program \( P \) which successively produces the digits of \( s \), we write \( P \rightarrow s \).
- For \( P \rightarrow s \), let \( T_P(n) \) be the number of steps up to the \( P \)-computation of the \( n \)th symbol of \( s \) (\( n \in \mathbb{N} - \{0\} \)).
- \( \mathbb{D} \)-Time(\( f \)) = \{ \( s \) \mid \exists P: P \rightarrow s; T_P(n) \leq f(n) \}.

The most intuitive control seems to be the one of time between the production of two consecutive symbols. Hence, define the class:

\[ \mathbb{B} = \{ s \mid \exists P, \exists K > 0: P \rightarrow s, T_P(n + 1) - T_P(n) \leq K \} \]

of "uniformly bounded production" sequences. The following result must be pointed out:

\[ \mathbb{R} \text{-Time} = \mathbb{D} \text{-Time}(n) = \mathbb{B} = \bigcup_{\alpha} \mathbb{D} \text{-Time}(an) = \mathbb{L} \text{-Time}. \]

The importance of \( \mathbb{B} \) is first of all, obviously, a practical one and secondly a theoretical one, since every automatic sequence is the homomorphic image of an element of \( \mathbb{B} \). (Results and comments on real-time sequences are essentially from [1].)

3. Iterative production

The basic idea is the ability of expansion contained in iterative processes: D0L-languages, iterated morphisms . . . ([2, 3]). Let \( A \) be a finite alphabet, \( h: A \rightarrow A^* \) a morphism (naturally extended to \( A^* \) and to \( A^\omega \)). \( \alpha \in A^\omega \) is a fixed point for the \( h \)-iteration iff \( \alpha = h(\alpha) = h^\omega(\alpha) \), where \( \alpha \in a.A^\omega \) and \( h(a) \in a.A^+ \).
Notations

PF: The class of infinite sequences which can be obtained as fixed points of morphism-iteration,

PFₐ: the subclass of PF obtained by using the ε-free morphisms only,

PFₖ: the subclass of PFₐ obtained with the growing morphisms,

PFₜ: the subclass of PFₖ obtained with the uniform morphisms.

Since these classes are not closed under elementary operations such as “finite modifications”, we consider their closure under literal morphisms (tag sequences) [4, 5]. Then we have “good” closure properties, and the following inclusions:

\[
\begin{align*}
&\text{PF} \supseteq \text{PF}ₐ \supseteq \text{PF}ₖ \supseteq \text{PF}ₜ \supseteq \text{PF} \\
&\text{TS} \supseteq \text{TS}ₐ \supseteq \text{TS}ₖ \supseteq \text{TS}ₜ = \text{TS}
\end{align*}
\]

And now, let us introduce a general idea for iterative production:

1. a fixed point “generation”,
2. an “encoding”.

In each of the two phases, we consider either a morphism or a transduction (the action of a D.G.S.M. [6]), with the aim of taking advantage of the following remarks [7]:

- each fixed point (PF for a morphism, PFT for a transduction) is in \( \mathbb{B} \),
- every automatic sequence is the homomorphic image of an element in PFT,
- \( \mathbb{B} \) is stable whatever the ε-free “encoding”.

3.1. Example of a fixed point obtained by the action of a transducer

\[ A = \{0, 1, S\}, \quad Q = \{q₀, q₁, q₂\}. \]

The transition functions are described by the diagram in Fig. 1.

\[ αₕ = \text{PFT}(q₀, 0) = 0S₁S \ldots ₁S₁₁S \ldots \]

- \( αₕ \in \text{PFT₉} \)
- the subword complexity of \( αₕ \) is \( p_{αₕ}(n) = O(n \cdot 2ⁿ) \). Figure 2 shows that PF and PFT are of quite different natures: the generating power is not contained in the “encoding” process, but in the fixed-point construction.

![Fig. 1.](image_url)
3.2. Some results

Table 1. Subword complexity

<table>
<thead>
<tr>
<th>PFu</th>
<th>PFc</th>
<th>PFc+PF</th>
<th>PFTu, PFTc, PFT</th>
<th>PIF, PF</th>
</tr>
</thead>
<tbody>
<tr>
<td>O(n)</td>
<td>O(n log n)</td>
<td>O(n^2)</td>
<td>O(n^3)</td>
<td>O(γ&quot;)**</td>
</tr>
</tbody>
</table>

* The subword complexity of the PFTu and PFTc sequences covers and does not exceed the polynomial growth.
** The subword complexity of the PFT and PFT sequences reaches the exponential growth.

Table 2. Main inclusions

PFu \subseteq PFTu \subseteq PFTc \subseteq PFTc \subseteq PFT \subseteq \mathcal{B}

Proofs either use subword complexity or structure criterions [7, 10] such as, for example, the following lemma.

Lemma. Let a ∈ A, α ∈ A^ω, α ∈ PFT and α not degenerated. Then,

\[ \lim_{n \to +\infty} \frac{C'_{\alpha}(n+1)}{C'_{\alpha}(n)} \leq k = \max_{b \in A, \gamma \in \Omega} |\lambda(q, b)|, \]

k being the exact bound.

C'_{\alpha}(m) is the index of the mth occurrence of a symbol in A - {a}. This criterion gives the Liouville sequence as an element of \( \mathbb{B} \)-PFT.

4. Conclusion

4.1.

- One can prove that PFT is not closed by “encoding”, not even by literal encoding (the proof uses a gap-criterion: [7]).
Under general "encoding", the two distinct classes $\mathbb{B}$ and PFT yield the same class $\mathcal{A}$ of all automatic sequences.

We have a partial result in the form of the following lemma.

**Lemma.** Let $\alpha \in A^\omega$, $\alpha \in \text{PFT}$; $h$ is a morphism such that $\{ a \in A \mid h(a) \neq \varepsilon \}$ has a density $d$ in $\alpha$ verifying $d \neq 0$. Then $h(\alpha) \in \mathbb{B}$.

**Question.** Is it possible to find a class of morphisms (respectively of "encoding" processes) so that $\mathbb{B}$ may be exactly deduced from PFT?

4.2.

What kind of control over subword complexity of PFT, (respectively PFT) is possible? We do not know if the maximal subword complexity can be reached. Concerning that question, Rauzy gave some information during the session, as follows.

**Proposition.** For each $\varepsilon > 0$ there exists an $\alpha \in \text{PFT}(\{0,1\})$ and $K > 0$ satisfying

$$K \cdot 2^{(1-\varepsilon)n} < p_\alpha(n) \leq 2^n.$$

We gave the example of a PFT$_r$-sequence $\alpha_S$ over $\{0,1,S\}$ in which a particular symbol $S$ separates $\bar{n}$ from $\bar{n+1}$. Similarly, we can construct a PFT$_s$-sequence $\alpha_T$ over $\{0,1\}$ where $\bar{n}$ is replaced by $\bar{n}$ and $S$ by $T$, $\bar{n}$ being the representation of the integer $n$ in a Fibonacci numeration system of order $m$ [11] and $T$ being the block $0.1^{n}$,

$$n = \sum_{i=0}^{m-1} \varepsilon_i u_i \quad (\varepsilon_i \in \{0,1\}; \varepsilon_1 \varepsilon_2 \ldots \varepsilon_{i+m-1} \neq 1^n),$$

$$u_{p+m} = \sum_{i=0}^{m} u_{p+i} \quad \forall p \geq 1. \quad (\ast)$$

Then $p_{\alpha_S}(n) \geq u_{n+1} - u_n$ and, using $(\ast)$, we can state $u_{n+1} - u_n = O(\theta^n)$ where $\theta \in [2m/(m+1), 2[$.

References


