Moore Automata and Epichristoffel Words

G. Castiglione and M. Sciortino

Dipartimento di Matematica e Informatica
Università di Palermo, via Archirafi, 34 - 90123 Palermo, Italy
{giusi, mari}@math.unipa.it.

In recent years (cf. [1, 3, 4]) a bridge between combinatorics on words and the study of complexity of algorithms for the minimization of finite state automata (*acceptors*) has aroused great interest. In particular, the study of combinatorial properties of *christoffel classes* (or *circular sturmian words*) allowed to prove that Hopcroft's minimization algorithm becomes not ambiguous when applied to the family of cyclic unary acceptors constructed by circular sturmian word. Furthermore, for particular subfamilies the tightness is obtained.

In the field of combinatorics on words, increasing the cardinality of the letters alphabet can give rise to new problematic questions. For example, for *episturmian words* (cf. [5]) representing an extension to a larger alphabet of the notion of infinite sturmian word, many of the crucial properties of such a family of words, as the balancing, are lost. These considerations also hold when finite combinatorial objects as circular words are considered.

We denote by (w) the circular word over the k-ary alphabet A corresponding to all the conjugates of the word w. Given a circular word (w) a factor u is called m-special if there exist exactly m distinct characters $a_1, a_2, \dots, a_m$ in the alphabet A, such that all ua_i are factors of (w) for each $i = 1, \dots, m$. Some families of circular words are able to capture many properties of classes of infinite words. For instance, in the binary case, circular sturmian words inherits the balancing from infinite sturmian words. Moreover each circular sturmian word (w) admits a unique 2-special factor for each length up to $|w| - 2$. In [4, 2] other structural characterizations have been also investigated.

Circular epichristoffel words, introduced in [8], are circular words that maintain some structural properties of episturmian words. More formally, we say that (w) is a circular epichristoffel word if it is the image of a letter by an episturmian morphism. One can prove that, for each length up to $|w| - 2$, there exists a unique special factor. In case of k-ary alphabet the problem of determining for each $2 \leq m \leq k$ the maximal length of all m-special factors can be investigated. Several properties of circular sturmian words can not be extended to circular epichristoffel words and there are a lot of open problems connected to such a family. Furthermore, the study of such a class seems to be connected to Fraenkel conjecture.

In this paper we deal with the question of how the process of minimization of a *Moore automaton* (cf. [7]) is influenced by the problems arising in combinatorics on words when alphabets of size greater than 2 are considered. Note that for such automata, differently from acceptors, the output alphabet is not binary. In particular, we analyze the behavior of a variant of Hopcroft's algorithm on a family of unary cyclic Moore automata associated to circular epichristoffel words and we relate the minimization process with particular factorization properties, here introduced, of such words.

Given $p = (p_1, p_2, \ldots, p_k)$ a k-tuple of non-negative integers, in [8] the author gives an algorithm to determine whether a circular epichristoffel word, having p as vector of occurrences of the letters, there exists and a construction is shown. All the steps of the construction determine a sequence of letters, called *directive sequence*, used to construct the circular epichristoffel word.

We prove that each letter a_i of a k-ary alphabet A uniquely determines a circular factorization of a circular epichristoffel word (w) defined over A in a set X_{a_i} containing k circular epichristoffel words. Such a factorization is induced by the directive sequence. Let $z_{a_i}(w)$ be the circular word obtained from (w) by encoding by $a_1, a_2, \ldots, a_k$ the occurrences of the correspondent elements of X_{a_i}. Let us denote by $(i(w))$ the circular epichristoffel word obtained by permuting the letters of (w) such that the associated k-tuple is not increasing, i.e. $p_1 \geq p_2 \geq \ldots \geq p_k$. We prove that the circular word $(i(z_{a_i}(w))$, denoted by $L_{a_i}(w)$, is a circular epichristoffel word. Therefore, we can associate to each epichristoffel word (w) a k-ary tree $\tau(w)$, called *reduction tree*, defined as follows.

- If w is a single letter a_i, $\tau(w)$ is a single node labeled by $(i(a_i)) = (a_1)$.
- If $|w| > 1$, $\tau(w)$ is a tree with root labeled by $(i(w))$ and at most k subtrees. The i-th subtree is $\tau(L_{a_i}(w))$.

Figure 1 shows an instance of reduction tree of a circular epichristoffel word and its correspondent factorizations. It is possible to prove that each circular epichristoffel word is uniquely determined by its reduction tree, as stated in the following theorem.

Theorem 1. *Let (w) and (w') be two circular epichristoffel words over the alphabet $A = \{a_1, \ldots, a_k\}$. Then, $\tau(w) = \tau(w')$ if and only if $(w') = (w)$ (up to a permutation of the letters).*

Let $(w) = (a_1 a_2 \ldots a_n)$ be a circular word over the alphabet A. The *cyclic automaton associated to* (w), denoted by $\mathcal{A}_w$, is a particular deterministic Moore automaton (DMA) $\mathcal{A} = (\Sigma, A, Q, q_0, \delta, \lambda)$ in which $Q = \{1, 2, \ldots, n\}$ is the set of states, $\Sigma = \{0\}$ is the input alphabet, A is the output alphabet, δ is the transition function defined as $\delta(i, 0) = (i + 1)$, $\forall\ i \in Q \setminus \{n\}$ and $\delta(n, 0) = 1$. The choice of q_0 does not affect the minimization process. Moreover, $\lambda : Q \mapsto \Gamma$ is a *output function* that assigns an output to the states of the automaton here defined as $\lambda(i) = a_i$ for each $i \in Q$. See Figure 2(a) for an example.

In this paper we propose a minimization strategy (called L-Minimization algorithm) for DMA that is variant of Hopcroft's minimization algorithm, the most efficient known minimization algorithm for acceptors (cf. [6]) that runs in time $O(n \log n)$. It can operate on a generic deterministic Moore automaton and it is based on two main ingredients. The first one is the notion of m-split operation, defined as follows. Given a partition Π of Q, let $\mathbb{C} \subset \Pi$, we say that $(\mathbb{C}, a)$ *m-splits* the class B if there exist $\{Q_1, Q_2, \ldots, Q_{m-1}\} \subset \mathbb{C}$ such

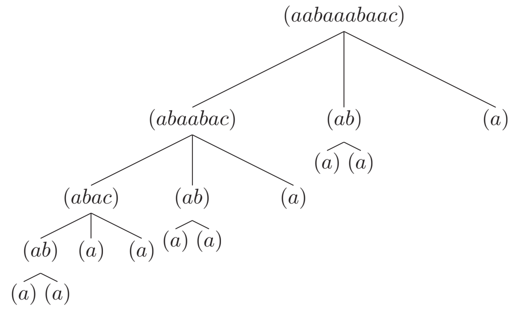


Fig. 1: The reduction tree $\tau(aabaaabaac)$. The circular word $(aabaaabaac)$ can be circularly factorized into circular epichristoffel words, as follows: $(w) = (a)(ab)(a)(a)(ab)(a)(ac)$, $(w) = (aacaab)(aaab)$, $(w) = (aabaaabaac)$. Such factorizations are coded by the circular epichristoffel words $(abaabac)$, (ca) and (c), respectively. Consequently, $L_a(w) = (abaabac)$, $L_b(w) = (ab)$, $L_c(w) = (a)$. Analogously, the other factorizations can be determined.

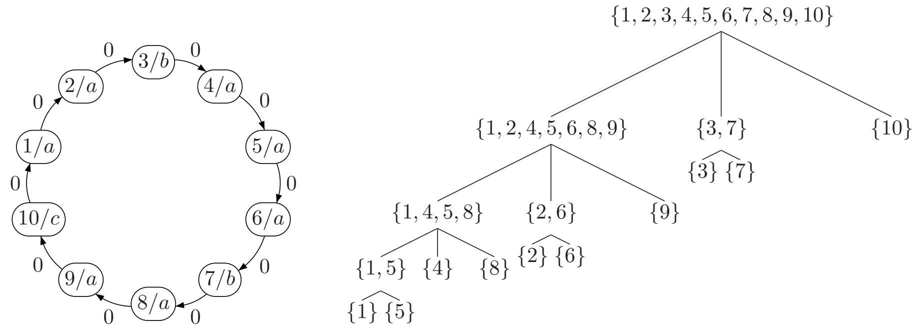
that $\delta_a^{-1}(Q_i) \cap B \neq \emptyset$ and $B \nsubseteq \delta_a^{-1}(Q_i)$, with $i = 1, \ldots, m-1$. In this case the set B can be, obviously, split into $B_i = \delta_a^{-1}(Q_i) \cap B$, with $i = 1, \ldots, m-1$ and $B_m = B \setminus \bigcup_{i=1,\ldots,m-1} B_i$. The pairs $(\mathbb{C}, a)$ are stored and successively extracted from an auxiliary data structure $\mathcal{W}$, called *waiting set*. The second ingredient is the *all but not the largest* strategy instead of *smaller half* strategy of the classical Hopcroft's minimization algorithm. In particular, we store into the waiting set $\mathcal{W}$ all but not the largest sets obtained from the m − *split* as the $(m-1)$-tuple $(\mathbb{C}, a)$. Such a strategy is applied throughout the algorithm, starting with the first step in which the set Q of states is split into classes of states having the same output function λ and this is fundamental in order to obtain the minimal automaton. The successive split operations and insertions into $\mathcal{W}$ could be execute by the smaller-half strategy where 2-splits can occur. Also in this case the minimal automaton is produced. Although, in general, L-minimization is not deterministic, we show that there is an infinite family of automata for which, differently from smaller-half strategy, it has a unique execution, as stated in the following theorem.

Theorem 2. *The execution of* L-Minimization *algorithm on cyclic automata associated to circular epichristoffel words is unique.*

Such a result can be proved by using the fact that a m-split occurs in correspondence with m-special factor of the circular word.

However, it is possible to verify that there exist infinite families of Moore automata for which the executions of our algorithm are better than some executions of the smaller-half minimization strategy and vice-versa. It could be interesting to study the tightness and the average time complexity of the two methods.

The refinement process produced during each execution of the algorithm on a generic Moore automaton $\mathcal{A}$, can be represented by a k-ary tree $\mathcal{T}(\mathcal{A})$, also



(a) (b)

Fig. 2: (a) Cyclic automaton $\mathcal{A}_w$ for $(w) = (aabaaabaac)$; $\Sigma = \{0\}$, $A = \{a, b, c\}$, $\lambda(1) = \lambda(2) = \lambda(4) = \lambda(5) = \lambda(6) = \lambda(8) = \lambda(9) = a$, $\lambda(3) = \lambda(7) = b$, $\lambda(10) = c$. (b) The derivation tree $\mathcal{T}(\mathcal{A}_w)$.

called *derivation tree*, whose nodes are labeled by classes of the partitions and their descendants are the classes produced by the m-split operations. See Figure 2(b) for an example. Note that, in general, the shape of the derivation tree is strongly affected from the non-deterministic choices of minimization algorithm, although the leaves are the same.

The following theorem establishes a relationship between derivation trees and reduction trees.

Theorem 3. *If (w) is a circular epichristoffel word then $\mathcal{T}(\mathcal{A}_w)$ and $\tau(w)$ are isomorphic.*

References

1. J. Berstel, L. Boasson, and O. Carton. Continuant polynomials and worst-case behavior of Hopcroft's minimization algorithm. *Theor. Comput. Sci.*, 410:2811–2822, 2009.
2. J.-P. Borel and C. Reutenauer. On christoffel classes. *ITA*, 40(1):15–27, 2006.
3. G. Castiglione, A. Restivo, and M. Sciortino. Hopcroft's algorithm and cyclic automata. In C. Martín-Vide, F. Otto, and H. Fernau, editors, *LATA*, volume 5196 of *Lecture Notes in Computer Science*, pages 172–183. Springer, 2008.
4. G. Castiglione, A. Restivo, and M. Sciortino. Circular sturmian words and Hopcroft's algorithm. *Theor. Comput. Sci.*, 410(43):4372–4381, 2009.
5. A. Glen and J. Justin. Episturmian words: a survey. *ITA*, 43(3):403–442, 2009.
6. J.E. Hopcroft. An $n \log n$ algorithm for mimimizing the states in a finite automaton. In *Theory of machines and computations (Proc. Internat. Sympos. Technion, Haifa, 1971)*, pages 189–196. Academic Press, New York, 1971.
7. E.F. Moore. *Gedaken experiments on sequential machines*, pages 129–153. Princeton University Press, 1956.
8. G. Paquin. On a generalization of christoffel words: epichristoffel words. *Theor. Comput. Sci.*, 410(38-40):3782–3791, 2009.