Sturmian trees

Jean Berstel[1], Luc Boasson[2], Olivier Carton[2] and Isabelle Fagnot[1]

[1] Institut Gaspard-Monge (IGM), Université de Marne-la-Vallée and CNRS, Marne-la-Vallée
[2] Laboratoire d'informatique algorithmique: fondements et applications (LIAFA), Université Denis-Diderot (Paris VII) and CNRS, Paris

This is a preliminary version of a work in progress.

Abstract. We consider Sturmian trees as a natural generalization of Sturmian words. A Sturmian tree is a tree having $n+1$ distinct subtrees of height n for each n. As for the case of words, Sturmian trees are irrational trees of minimal complexity. We give various examples of Sturmian trees, and we characterize one family of Sturmian trees by means of a structural property of their automata.

1 Introduction

A *Sturmian tree* is a complete labeled binary tree having exactly $n + 1$ distinct subtrees of height n for each n. Thus Sturmian trees are defined by extending to trees one of the numerous equivalent definitions of Sturmian words. Sturmian trees share the same property of minimal complexity than Sturmian words: indeed, if a tree has at most n distinct subtrees of height n for some n, then the tree is rational, i.e. it has only finitely many distinct infinite subtrees.

This paper presents many examples and some results on Sturmian trees. The simplest method to construct a Sturmian tree is to choose a Sturmian word and to repeat it on all branches of the tree. We call this a uniform tree, see Figure 1. However, many other categories of Sturmian trees exist.

Contrary to the case of Sturmian words, and similarly to the case of episturmian words, there seems not to exist equivalent definitions for the family of Sturmian words. This is due of course to the fact that, in our case, each node in a tree has two children, and only one of them needs to be the root of a Sturmian tree to make the whole tree Sturmian. Each tree labeled with two symbols can be described by the set of words labeling paths from the root to nodes sharing a distinguished symbol. The (infinite) minimal automaton of the language has quite interesting properties when the tree is Sturmian. The most useful is that the Moore equivalence algorithm produces just one additional equivalence class at each step. We call these automata *slow*.

We have observed that two parameters make sense in studying Sturmian words: the *degree* of a Sturmian tree is the number of disjoint infinite paths composed of nodes which are all roots of Sturmian trees. The *rank* of a tree is the number of distinct rational subtrees it contains. Both parameters may be finite or infinite.

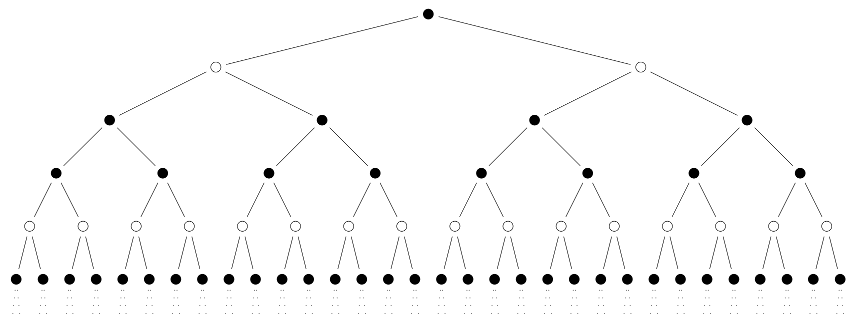

Fig. 1. The top of a uniform tree for the word $abaaba\cdots$. Node label a is represented by $\bullet$, and label b is represented by $\circ$. This tree will be seen to have infinite index and rank 0.

The main result of this paper is that the class of Sturmian trees of degree one and with finite rank can be described by infinite automata of a rather special form. The automata are obtained by repeating infinitely many often a distinguished path in some finite slow automaton, and intertwining consecutive copies of this path by letters taken from some Sturmian infinite word. Another property is that a Sturmian tree with finite degree at least 2 always has infinite rank.

The class of Sturmian trees seems to be quite rich. We found several rather different techniques to construct Sturmian trees. To the best of our knowledge, there is only one paper on Sturmian trees prior to the present one, by Carpi, De Luca and Varricchio [1].

2 Sturmian Trees

We are interested in complete labeled infinite binary trees, and we consider finite trees insofar as they appear inside infinite trees.

In the sequel, D denotes the alphabet $\{0,1\}$. A *tree domain* is a prefix-closed subset P of D^*. Any element of a tree domain is called a *node*. Let A be an alphabet. A *tree over* A is a map t from a tree domain P into A. The domain of the tree t is denoted $\mathrm{dom}(t)$. For each node w of t, the letter $t(w)$ is called the *label* of the node w. A *complete* tree is a tree whose domain is D^*. The *empty tree* is the tree whose domain is the empty set. A (finite or infinite) *branch* of a tree t is a (finite or infinite) word x over D such that each prefix of x is a node of t.

Example 1. (*Dyck tree*) Let A be the alphabet $\{a,b\}$. Let L be the set of Dyck words over $D=\{0,1\}$, that is the set of words generated by the context-free grammar with productions $S \to 0S1S+\varepsilon$. The *Dyck tree* is the complete tree defined by

$$t(w) = \begin{cases} a & \text{if } w \in L, \\ b & \text{otherwise.} \end{cases} \tag{1}$$

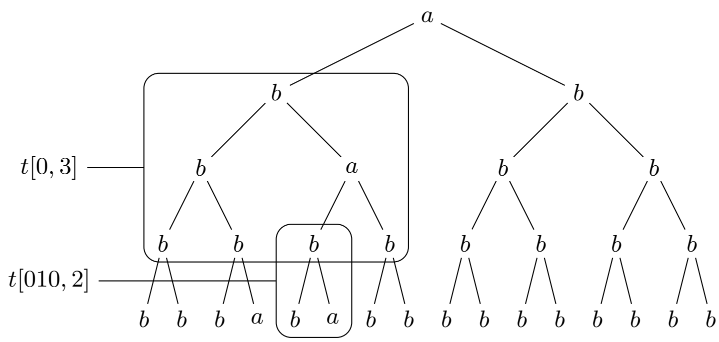

Fig. 2. The top of the Dyck tree of Example 1 and two of its factors, of height 3 and 2, respectively.

The top of this tree is depicted in Figure 2. The first four words ε, 01, 0101 and 0011 of L correspond to the four occurrences of the letter a as label on the top of the tree.

More generally, the *characteristic tree* of any language L over D is defined to be the tree t given by Equation (1). Conversely, for any tree t over some alphabet A, and for any letter a in A, there is a language $L = t^{-1}(a)$ of words labeled with the letter a. The language $L = t^{-1}(a)$ is called the *a-language of t*. In the sequel, we usually deal with the two-letter alphabet $A = \{a, b\}$, and we fix the letter a. We then say the *language of t* instead of the a-language.

We shall see that the a-languages of a tree t are regular if and only if the tree t is rational. For any word w and any language L, the expression $w^{-1}L$ denotes the set $w^{-1}L = \{x \mid wx \in L\}$. Let t be a tree over A and w be a word over D. We denote by $t[w]$ the tree with domain $w^{-1}\operatorname{dom}(t)$ defined by $t[w](u) = t(wu)$ for each u in $w^{-1}\operatorname{dom}(t)$. The tree $t[w]$ is sometimes written as $w^{-1}t$, for instance in [1]. If w is not a node of t, the tree $t[w]$ is empty. A tree of the form $t[w]$ is the *suffix* of t rooted at w. Suffixes are also called quotients or subtrees in the literature.

Let t be a tree over A and let w be a word over D. For a positive integer h, we denote by $D^{<h}$ the set $(\varepsilon + D)^{h-1}$ of words over D of length at most $h - 1$. We set $D^{<0} = \emptyset$.

Let h be a nonnegative integer. The *truncation* of a tree t at height h is the restriction of t to the domain $D^{<h}$. Any tree obtained by truncation is called a *prefix* of t. A *factor* of t is a prefix of a suffix of t. More precisely, for any word w and any nonnegative integer h, we denote by $t[w, h]$ the factor of height h rooted at w, that is the tree of domain $w^{-1}\operatorname{dom}(t) \cap D^{<h}$ and defined by $t[w, h](u) = t(wu)$. A factor of height 0 is always the empty tree. A factor $t[w, 1]$ of height 1 can be identified with the letter $t(w)$ of A that labels its root. A prefix is a tree of the form $t[\varepsilon, h]$.

Factors of height h are sometimes considered to have height $h - 1$ in the literature (e.g. [1]). In this paper, the height of a finite tree is the number of nodes along a maximal branch and not the number of steps in-between. Our convention will be justified by Proposition 1 which extends a similar result for words in similar terms.

The following equation for any words w and w' over D and any positive integers h and h' holds:

$$t[w,h][w',h'] = t[ww', \min(h - |w'|, h')] \qquad \text{for } |w'| \leq h.$$

A tree is *rational* if it has finitely many distinct suffixes. Recall (see e.g. [2]) that a tree over an alphabet A is rational if and only if $t^{-1}(a) = \{w \in D^* \mid t(w) = a\}$ is a regular subset of D^* for each letter a of A. For instance the Dyck tree t of Example 1 is not rational since $t^{-1}(a)$ is the Dyck language which is not regular [4]. The following proposition gives a characterization of complete rational trees using factors. It extends to trees the characterization of ultimately periodic words by mean of their subword complexity [3]. This statement appears in [1].

Proposition 1. *A complete tree t is rational if and only there is an integer h such that t has at most h distinct factors of height h.*

A complete tree is *Sturmian* if for any integer h, it has $h+1$ factors of height h. Since the factors of height 1 are the letters $t(w)$ a Sturmian tree is defined over a two letter alphabet. In what follows, we always assume that this alphabet is $\{a,b\}$.

We will prove later that the Dyck tree given in Example 1 is indeed Sturmian. We start with some simpler examples of Sturmian trees.

In the first of these examples, the same infinite word is repeated along each branch of the tree.

Example 2. (*Uniform trees*) Let $x = x_0x_1x_2\ldots$ be an infinite word over an alphabet A, where $x_0, x_1, \ldots,$ are letters. The *uniform tree* of x is the complete tree t defined by $t(w) = x_{|w|}$. This means of course that all nodes of the same level n in the tree are labeled with the same symbol x_n. If x is a Sturmian word, then its uniform tree t is a Sturmian tree. Figure 1 shows the top of the uniform tree of word $x = abaabb\cdots$.

Example 3. (*Left branch tree*) Let $x = x_0x_1x_2\ldots$ be an infinite word over A, where $x_0, \ldots,$ are letters. We define a complete tree t by $t(w) = x_{|w|_0}$. (Recall that $|w|_d$ is the number of occurrences of d in w).

The label of each node w is the letter x_n of x, where n is the number of symbols 0 occurring on the path from the root to w. The label of the root node is x_0. If the label of w is x_n, the labels of $w0$ and $w1$ are respectively x_{n+1} and x_n.

In particular, the letters of the word x label the nodes of the left branch of the tree, and all nodes on a right branch share the same label. Figure 3 shows the top of the left branch tree of word $x = abaabb\cdots$.

In the previous examples, two factors $t[w,h]$ and $t[w',h]$ of height h are equal if and only if $x[|w|,h] = x[|w'|,h]$ in Example 2 (resp. if and only if $x[|w|_0,h] =$

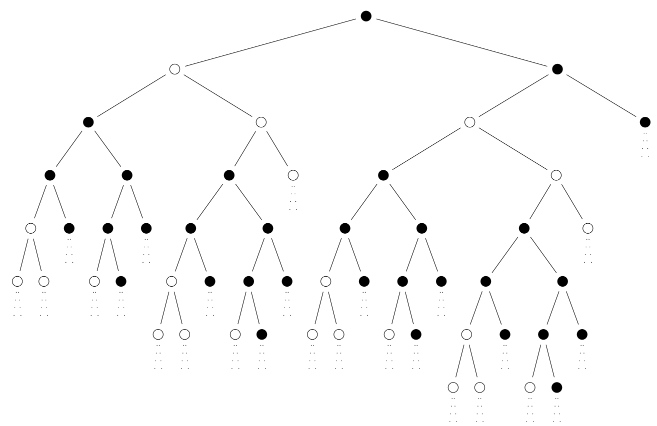

Fig. 3. The top of a left branch tree for the word $abaabb\cdots$.

$x[|w'|_0, h]$ in Example 3). (We write $x[n, h]$ for the factor $x_n x_{n+1} \cdots x_{n+h-1}$ of the word x.) It follows that in these examples, the tree t is Sturmian if and only if the word x is Sturmian.

Example 4. (*Indicator tree*) Let $x = x_0x_1x_2\cdots$ be an infinite word over D. The *indicator tree* of x is the complete tree t defined by

$$t(w) = \begin{cases} a & \text{if } w \text{ is a prefix of } x, \\ b & \text{otherwise.} \end{cases}$$

In other terms, there is exactly one infinite path in t with all its nodes labeled by the letter a. The letters of this path are the letters of the word x. Equivalently, the indicator tree of the infinite word x is the characteristic tree of the language composed of its (finite) prefixes. Figure 4 shows the indicator tree of the Fibonacci word. It can be easily proved that x is a Sturmian word if and only if its indicator tree t is a Sturmian tree.

An indicator tree has rank 1 and degree 1.

The following example is a variation on Example 4. For a finite word w and an infinite word x, we denote by $d(w, x)$ the integer $|w| - |u|$ where u is the longest common prefix of w and x.

Example 5. (*Band indicator tree*) Let $x = x_0x_1x_2\dots$ be an infinite word over D and let k be a non-negative integer. The *band indicator tree of width* k is the complete tree t defined by

$$t(w) = \begin{cases} a & \text{if } d(w, x) \le k, \\ b & \text{otherwise.} \end{cases}$$

Again, x is a Sturmian word if and only if t is a Sturmian tree. A band indicator tree has degree 1. The band indicator tree of width 0 is the indicator tree defined in Example 4, since $d(w, x) \le 0$ if and only if w is a prefix of x.

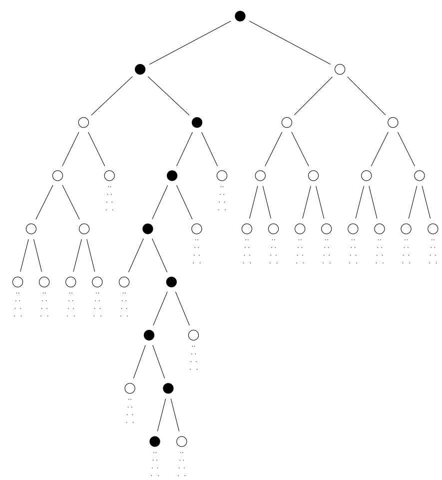

Fig. 4. The top of the indicator tree for the Fibonacci word $01001010 \cdots$. The only nodes labeled a are on the Fibonacci path.

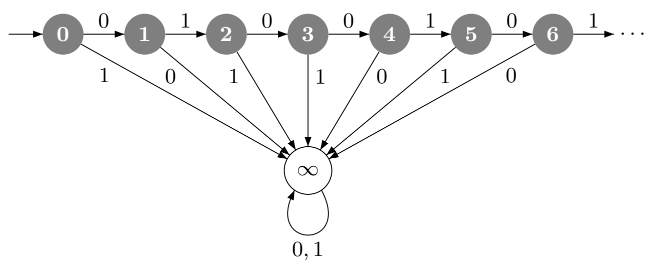

Fig. 5. Automaton accepting the prefixes of $01001010 \cdots$. All states are final excepted $+\infty$.

3 Rank and degree

Recall that a *branch* of a tree is a (finite or infinite) word x over D such that each prefix of x is a node of the tree.

A node w of a tree t is called *rational* if the suffix $t[w]$ is a rational tree. The *rank* of a tree t is the number of distinct rational suffixes of t. This number is either a nonnegative integer or infinite.

If w is an irrational node, then its prefixes also are irrational. Furthermore, at least one of the two words $w0$ and $w1$ also is irrational. The set of irrational nodes of a tree is a tree domain in which any finite branch is the prefix of an infinite branch.

The *degree* of a tree t is the number of infinite branches composed of irrational nodes. This number is either a nonnegative integer or infinite.

As a first example, consider the Dyck tree defined in Example 1. It has rank 1 and has infinite degree. A node w of this tree is rational if it is not a prefix of some Dyck word. The set of rational nodes is thus the set $L1D^*$ where L is the set of Dyck words. On the contrary, each branch in 00^*10^ω only contains irrational nodes. The degree of the Dyck tree is thus infinite.

Next, let t be the indicator tree of a Sturmian word x, as defined in Example 4. A node w of t is irrational if and only if it is a prefix of x. Thus, the word x itself is the only infinite branch composed of irrational nodes, and therefore the degree of this tree is 1. All rational subtrees are the same, so this tree has rank 1.

These examples show that there are Sturmian trees of degree 1 or of infinite degree. It turns out that there exist also Sturmian trees of finite degree greater than 1. In the final section, we construct a Sturmian tree of degree 2 but this construction is rather involved.

Here is a table containing the result of discussing the relation between degree and rank. A tree with rank 0 always has infinite degree since there is no rational node.

	rank	
degree	finite	infinite
1	characterized in Theorem 1	example 9 (appendix)
≥ 2, finite	empty by Proposition 4	example 10 (appendix)
infinite	example of Dyck tree	example not given here

The main result of the paper is the characterization of Sturmian trees of degree 1 and with finite rank by a structural property of the minimal automaton of its language.

4 Automata

We recall elementary properties of automata, just observing that they hold also when the set of states is infinite. We only use deterministic and complete automata. An *automaton* $\mathcal{A}$ over a finite alphabet D is composed of a state set Q, a set $F \subseteq Q$ of *final states*, and of a *next-state function* $Q \times D \to Q$ that maps (q, d) to a state denoted by $q \cdot d$. Given a distinguished state i, a word w over D is *accepted* by the automaton if the state $i \cdot w$ is final. When we emphasize the existence of state i, we call it the initial state as usual.

An automaton $\mathcal{B}$ is a *subautomaton* of an automaton $\mathcal{A}$ if its set of states is a subset of the set of states of $\mathcal{A}$ which is closed under the next-state function of $\mathcal{A}$.

Example 6. (*Dyck automaton*) The following automaton accepts the Dyck language. The set of states is $Q = \mathbb{N} \cup \{\infty\}$. The initial and unique final state is 0. The next state function is given by $n \cdot 0 = n+1$ for $n \geq 0$, $n \cdot 1 = n-1$ for $n \geq 1$, $0 \cdot 1 = \infty$ and $\infty \cdot 0 = \infty \cdot 1 = \infty$. This automaton is depicted in Figure 6. We call it the *Dyck automaton*. The singleton $\{\infty\}$ is the unique proper subautomaton of the Dyck automaton.

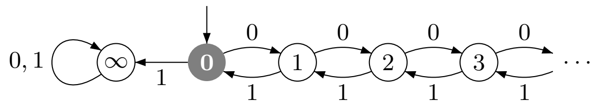

Fig. 6. Automaton of the Dyck language. State 0 is both the initial and the unique terminal state.

Given an arbitrary automaton $\mathcal{A}$, we define inductively a sequence $(\sim_n)_{n\geq 1}$ of equivalence relations on Q as follows.

$$q \sim_1 q' \iff (q \in F \iff q' \in F)$$
$$q \sim_{n+1} q' \iff (q \sim_n q' \text{ and } \forall d \in D \;\; q \cdot d \sim_n q' \cdot d)$$

These are well-known in the case of finite automata, and many properties extend to general automata. We call $\sim_n$ the *Moore equivalence* of order n. The *index* of $\sim_n$ is the number of equivalence classes of $\sim_n$.

The equivalence $\sim_{h+1}$ is a refinement of the equivalence $\sim_h$. Thus the index of $\sim_{h+1}$ is at least the index of $\sim_h$. An automaton is called *slow* if it is minimal and if the index of $\sim_h$ is at most $h+1$ for all $h \geq 1$. If $\sim_h$ and $\sim_{h+1}$ are different, that there is one class c of $\sim_h$ which gives raise to two classes in $\sim_{h+1}$. We say that $\sim_{h+1}$ *splits* class c, or that class c is *split* by $\sim_{h+1}$.

It is sometimes useful to distinguish, in a minimal automaton, two kinds of states. A state p is *rational* if it has finitely many descendants in the automaton, viewed as a graph or, equivalently, if it generates a finite subautomaton. States which are not rational are called *irrational*. In the minimal automaton associated to the language of a tree, a state is rational if and only if it corresponds to the root of a rational tree.

Let t be a complete tree over $\{a, b\}$. The *language* of t is the set $t^{-1}(a)$. Let $\mathcal{A}$ be an automaton accepting $t^{-1}(a)$. The following proposition shows that the classes of $\sim_h$ are in a one to one correspondence with the factors of t of height h.

Proposition 2. *Let t be a complete tree over $\{a, b\}$ and let $\mathcal{A}$ be an automaton over D accepting the language of t, with initial state i. For any words $w, w' \in D^*$ and any positive integer h, one has*

$$i \cdot w \sim_h i \cdot w' \iff t[w, h] = t[w', h]\,.$$

Corollary 1. *Let t be a complete tree over $\{a, b\}$ and let $\mathcal{A}$ be an automaton over D accepting the language of t. The tree t is Sturmian iff each equivalence relation $\sim_h$ has $h + 1$ classes.*

5 Trees with finite rank

5.1 A tree of degree one

In this section, we give an example of a family of Sturmian trees with finite rank and of degree 1 by describing the family of automata accepting their languages. These (infinite) automata are based on a finite slow automaton. In this automaton, a path is distinguished (called a lazy path). The infinite automaton is obtained by repeating the lazy path and intertwining the copies with symbols taken from an infinite Sturmian word.

In the next section, we show that any Sturmian tree of degree 1 and with finite rank can be obtained in this way.

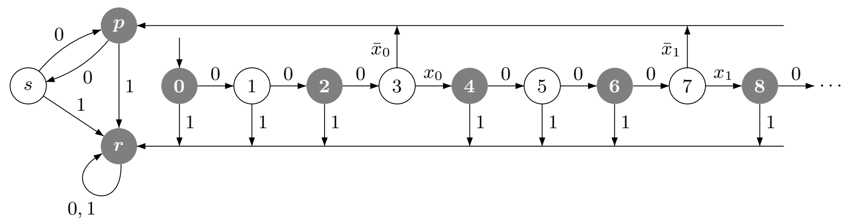

Fig. 7. A slow automaton $\hat{\mathcal{A}}$ for the Fibonacci word $x_0x_1\cdots = 01001010\cdots$. The final states are $p, r, 0, 2, 4, \cdots$.

Let $\mathcal{A} = (Q, \{i\}, F)$ be a finite deterministic automaton over the alphabet D with N states. We assume that $\mathcal{A}$ has the two following properties. First, $\mathcal{A}$ is *slow*. Recall that by definition, this means that the automaton is minimal and that Moore's minimization algorithm splits just one equivalence class into two new classes at each step.

Next, we suppose that there is a *lazy path* in $\mathcal{A}$. This is a path

$$\pi : q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_1 \cdots q_{h-1} \xrightarrow{a_{h-1}} q_h$$

of length h, where q_0 and q_h are the two states which are separated in the last step in Moore's algorithm together with the condition that

$$q_{h-1} \cdot \bar{a}_{h-1} = q_0 \text{ or } q_h .$$

The first of these conditions means that $q_0 \sim_{N-2} q_h$ and $q_0 \not\sim_{N-1} q_h$. As a consequence, the second property means that $q_{h-1} \cdot \bar{a}_{h-1}$ cannot be separated from $q_{h-1} \cdot a_{h-1}$ before the very last step of the Moore algorithm.

Example 7. The automaton $\hat{\mathcal{A}}$ given in Figure 7 has a subautomaton $\mathcal{A}$ composed of the states $\{p, s, r\}$. This subautomaton is slow: the first partition is into $\{p, r\}$ and $\{s\}$, and the second partition is equality. The finite subautomaton $\mathcal{A}$ in Figure 7 admits for example the lazy path $\pi : p \xrightarrow{0} s \xrightarrow{0} p \xrightarrow{0} s \xrightarrow{1} r$, and indeed $s \xrightarrow{0} p$. Here $h = 4$.

Given the finite slow automaton $\mathcal{A}$, the lazy path π and an infinite word $x = x_0x_1x_2\cdots$ over D, we now define an infinite minimal automaton $\hat{\mathcal{A}}$ which accepts the set of nodes labeled a of a tree t. We will show that if x is a Sturmian word, then t is a Sturmian tree of degree 1. This automaton is the *extension* of $\mathcal{A}$ by π and x, and is denoted by $\hat{\mathcal{A}} = \mathcal{A}(\pi, x)$.

The set of states of $\hat{\mathcal{A}}$ is $Q \cup \mathbb{N}$. For convenience, we use a mapping $q : \mathbb{N} \to Q$ defined by $q(n) = q_{n \bmod h}$ for any $n \in \mathbb{N}$. Here $q_0, \dots q_h$ are the states of the lazy path of $\mathcal{A}$. The initial state of $\hat{\mathcal{A}}$ is 0 and its set of final states is $F \cup q^{-1}(F)$. The next-state function of $\mathcal{A}$ is extended to $\hat{\mathcal{A}}$ by setting, for $n \in \mathbb{N}$,

(α) if $n \not\equiv h-1 \mod h$, then

$$n \cdot a_{n \bmod h} = n+1, \qquad n \cdot \bar{a}_{n \bmod h} = q(n) \cdot \bar{a}_{n \bmod h}$$

(β) if $n = ih + h - 1$ for some $i \geq 0$, then

$$n \cdot x_i = n+1, \qquad n \cdot \bar{x}_i = q_0$$

The infinite path through the integer states of the automaton $\hat{\mathcal{A}}$ is composed of an infinite sequence of copies of the lazy path of $\mathcal{A}$. For each state $q(n)$ inside each of the copies of the lazy path, the next-state for the "other" letter, that is the letter $\bar{a}_{n \bmod h}$, maps $q(n)$ back into $\mathcal{A}$. Two consecutive copies of the lazy path, say the ith and $i+1$th, are linked together by the letter x_i of the infinite word x driving the automaton.

Proposition 3. *Let $\hat{\mathcal{A}} = \mathcal{A}(\pi, x)$ be the extension of the finite slow automaton $\mathcal{A}$ by a lazy path π and an infinite word x. If the word x is Sturmian, then $\hat{\mathcal{A}}$ defines a tree t which is Sturmian, of degree 1, and having finite rank.*

The tree defined by this automaton has degree 1 since the only irrational states are the integer states n and they all lie on a single branch. We claim that this tree is also Sturmian. The proof is through three lemmas.

We denote by $\sim_k$ the Moore equivalence on the states. The next three lemmas just prove that the automaton has the required properties. We fix the automaton $\mathcal{A}$, set N to the number of its states, and we fix the lazy path $\pi : q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} \cdots \xrightarrow{a_{h-1}} q_h$. We also use the notation $q(n) = q_{n \bmod h}$. However, x is not required to be Sturmian in the three following lemmas.

5.2 Characterization

In this section, we give a characterization of Sturmian trees of degree 1 which have finite rank by describing the family of automata accepting their languages. These (infinite) automata are extensions of a finite automaton by a lazy path and a Sturmian word.

Theorem 1. *Let t be a Sturmian tree of degree one having finite rank, and let $\hat{\mathcal{A}}$ be the minimal automaton of the language of t. Then $\hat{\mathcal{A}}$ is the extension of a slow finite automaton $\mathcal{A}$ by a lazy path π and a Sturmian word x, i.e. $\hat{\mathcal{A}} = \mathcal{A}(\pi, x)$.*

Given a tree t and some Moore equivalence $\sim_h$ on its minimal automaton, it is convenient to call an quivalence class of $\sim_h$ an *irrational class* if it is entirely composed of irrational states. It is a *rational class* otherwise. A rational class contains at least one rational state, and may contain even infinitely may irrational states.

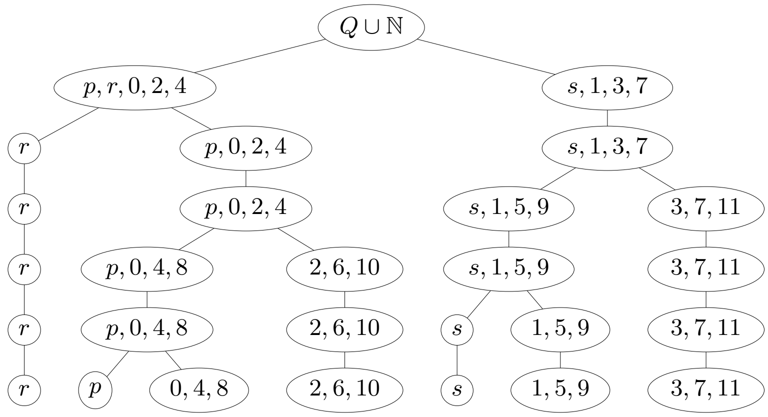

Fig. 8. The tree showing the evolution of the Moore equivalence relations on the automaton given in the previous picture. Each level describes a partition. Each level has one class splitting into two classes at the next level.

Lemma 1. *Let t be a Sturmian tree with finite rank. Either there is an integer n such that all rational classes of $\sim_n$ are singletons, or there is an integer n such that all irrational obtained by splitting a class of $\sim_n'$ for $n' \geq n$ are singleton classes.*

We will see later examples of Sturmian trees of degree greater than one. In these examples, the trees have finite rank. This is due to the following properties of Sturmian trees.

Proposition 4. *The degree of a Sturmian tree with finite rank is either one or infinite.*

Proof. Let t be a Sturmian tree with finite rank, and assume it has finite degree $d > 1$. A node w of t is a *fork* if both $w0$ and $w1$ are irrational nodes. Since t has degree d, it has exactly $d - 1$ fork nodes.

A state of the minimal automaton of t is a fork state if it is the state of a fork node. The automaton has at most $d - 1$ fork states. We want to show that for large enough n, an equivalence class of $\sim_n$ containing a fork state is a singleton.

In view of Lemma 1, there are two cases. Either there is an integer n such that all rational states are singletons for $\sim_n$. Then a class of $\sim_n$ containing a fork state contains only fork states since indeed a state that is not a fork state maps to a rational state by at least one letter, whereas a fork state does not. So any class containing a fork state is finite, and therefore will be split eventually into singleton classes.

In the other case described by Lemma 1, irrational states will be in singleton classes for large enough n. Again, since there are only finitely many fork states, each of these will be constructed at some step in the Moore algorithm.

Consider now an integer H such that each fork state is a singleton class of $\sim_H$. This means that the Nerode equivalence and the equivalence $\sim_H$ coincide for these states, and consequently two fork nodes in the tree define the same state in the automaton if and only if they are the roots of the same subtree of height H. Since there are infinitely many occurrences of any subtree of height H in Sturmian tree, there are infinitely many nodes that correspond to the same fork state, so there are infinitely many fork nodes. This yields the desired contradiction.

5.3 Another example

Example 8. Let x and x' be two Sturmian words over D that have exactly the same factors but share no common suffix: for each factorization $x = uy$, $x' = u'y'$, one has $y \neq y'$. Such words do exist and can even be constructed explicitly. We define a tree $t_{x,x'}$ by giving a (minimal) automaton accepting $t_{x,x'}^{-1}(a)$. Let Q be the set $\mathbb{N} \cup \{n' \mid n \in \mathbb{N}\} \cup \{\infty\}$. The only final state is ∞.

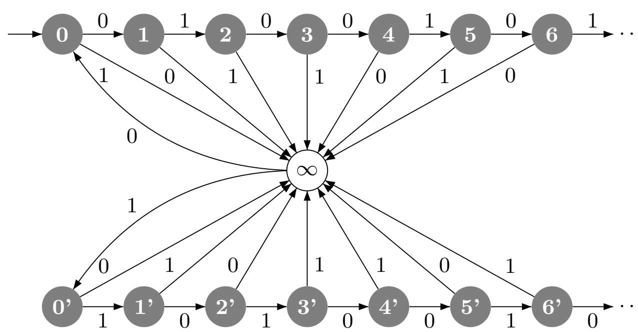

Fig. 9. Automaton of $t_{x,x'}$ for $x = 01001010\ldots$ and $x' = 10100100\ldots$

To prove that $t_{x,x'}$ is Sturmian, one shows that the automaton is slow and minimal. The automaton is indeed minimal because the two words share no common suffix, and it is slow because the two words are Sturmian *and* have the same set of factors. This tree has rank 0 and infinite degree.

6 Trees with infinite rank

There exist Sturmian trees with infinite rank given in the Appendix.

References

1. A. Carpi, A. de Luca, and S. Varricchio. Special factors and uniqueness conditions in rational trees. *Theory of Computing Systems*, 34:375–395, 2001.
2. Bruno Courcelle. Fundamental properties of infinite trees. *Theoretical Computer Science*, 25:95–165, 1983.
3. E. M. Coven and G. A. Hedlund. Sequences with minimal block growth. *Mathematical Systems Theory*, 7:138–153, 1973.
4. J. Sakarovitch. *Élements de théorie des automates*. Vuibert, 2003.

Appendix

There exist Sturmian trees with infinite rank. The following example gives a Sturmian tree with infinite rank and of degree 1.

Example 9. We define a tree by giving a (minimal) automaton accepting its language. The set of states of the automaton is $Q = \{n \in \mathbb{N} \mid n \geq 3\} \times \{0, 1\}$. The set of final states is the set $\{(n, b) \in Q \mid n \equiv 0 \mod 2\}$. The set E of transitions is defined as follows. Let $n = 2^k m$ where $m \geq 1$ and $m \not\equiv 0 \mod 2$. The integer 2^k is then the greatest power of 2 which divides n.

$$(n,b)\cdot 0 = \begin{cases} (2^{k-1}+1, 0) & \text{if } m = 1 \text{ and } b = 0 \\ (n+1, b) & \text{otherwise} \end{cases}$$

$$(n,b)\cdot 1 = \begin{cases} (3,0) & \text{if } k = 0 \\ (4,0) & \text{if } k = 1 \\ (4,0) & \text{if } k = 2,\ m = 1 \text{ and } b = 0 \\ (2^{k-2}+1, 0) & \text{if } k > 2,\ m = 1 \text{ and } b = 0 \\ (2^{k-1}+1, 0) & \text{otherwise} \end{cases}$$

In Figure 10 we give a picture of this automaton; states of the form $(n, 0)$ are drawn as circles n and states of the form $(n, 1)$ as squares $\boxed{n}$.

The previous example of a Sturmian tree of degree one having infinite rank can be extended to an example of a Sturmian tree of degree two which must have infinite rank in view of Proposition 4.

Example 10. We define a tree by giving a (minimal) automaton accepting its language. The set of states of the automaton is the following set Q

$$\begin{aligned} Q = {} & \{(n,0) \mid n \geq 2\} \\ & \cup \{(n,1) \mid n = 3 \text{ or } 3^{k+2} - 4\cdot 3^k + 1 \leq n \leq 3^{k+2} \text{ with } k \geq 0\} \\ & \cup \{(n,2) \mid n \in \{2,3\} \text{ or } 3^{k+2} - 3^k + 1 \leq n \leq 3^{k+2} \text{ with } k \geq 0\}. \end{aligned}$$

The set of final states is the set $\{(n, b) \in Q \mid n \equiv 0 \mod 2\}$. The set E of transitions is defined as follows. Let $n = 2^k 3^l m$ where $m \not\equiv 0 \mod 2$ and $m \not\equiv 0 \mod 3$. The integers 2^k and 3^l are the greatest powers of 2 and 3 which divide n.

$$(n,b)\cdot 0 = \begin{cases} (3^{l-1}+1, 0) & \text{if } k = 0,\ m = 1 \text{ and } b = 0 \\ (3^{l+1} - 4\cdot 3^{l-1}+1, 1) & \text{if } k = 0,\ m = 1 \text{ and } b = 1 \\ (3^{l-1} - 2\cdot 3^{l-1}+1, 2) & \text{if } k = 0,\ m = 1 \text{ and } b = 2 \\ (n+1, b) & \text{otherwise} \end{cases}$$

$$(n,b)\cdot 1 = \begin{cases} (3,1) & \text{if } n = 2 \text{ and } b = 2 \\ (3,0) & \text{if } n = 3 \text{ or } l = 0 \\ (3^{l-2}+1, 0) & \text{if } k = 0,\ l \geq 2 \text{ and } m = 1 \\ (3^{l-1}+1, 0) & \text{if } k = 0,\ l \geq 1 \text{ and } m \geq 2 \\ (3^l+2, 0) & \text{otherwise} \end{cases}$$

In Figure 11 we give a picture of this automaton; states of the form $(n, 0)$ are drawn as circles n and states of the form $(n, 1)$ as squares $\boxed{n}$ and states of the form $(n, 2)$ as lozenges n .

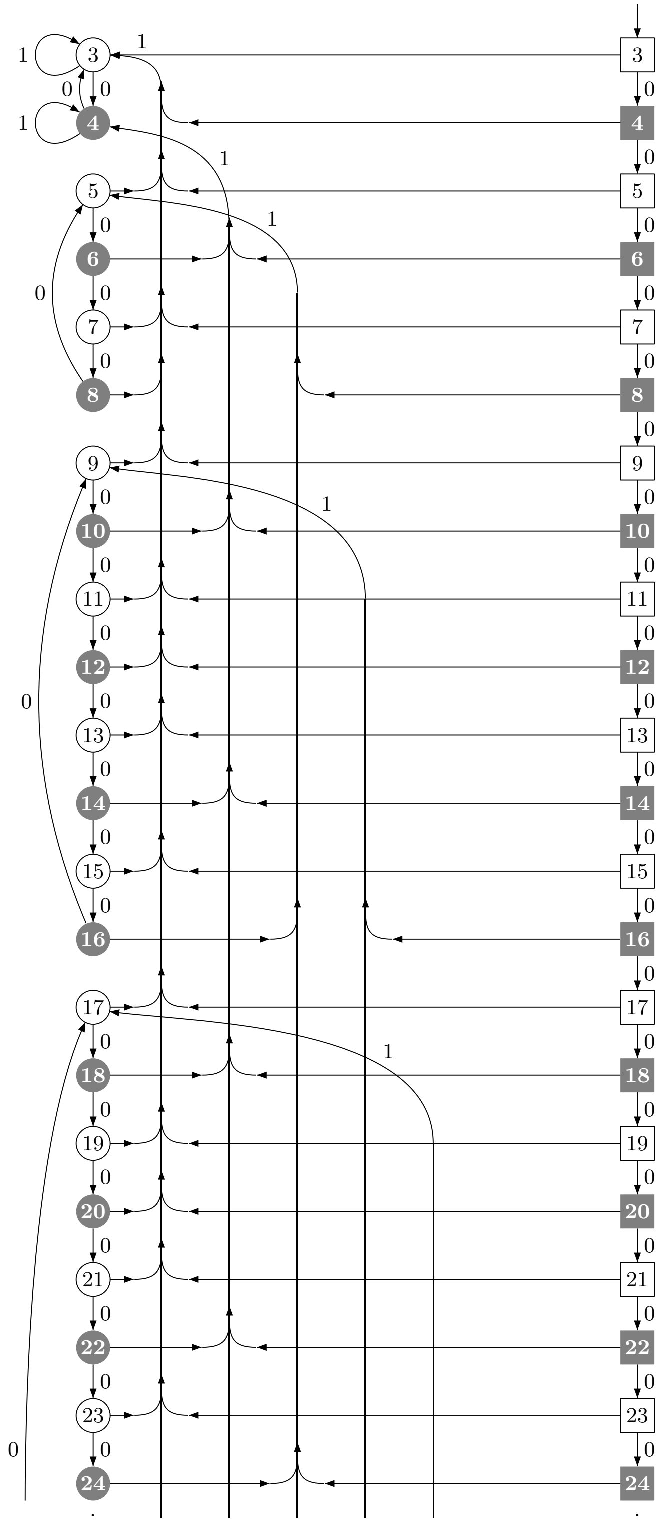

Fig. 10. Final states are dark. Observe the fractal-like structure, with a doubling of the size of each block.

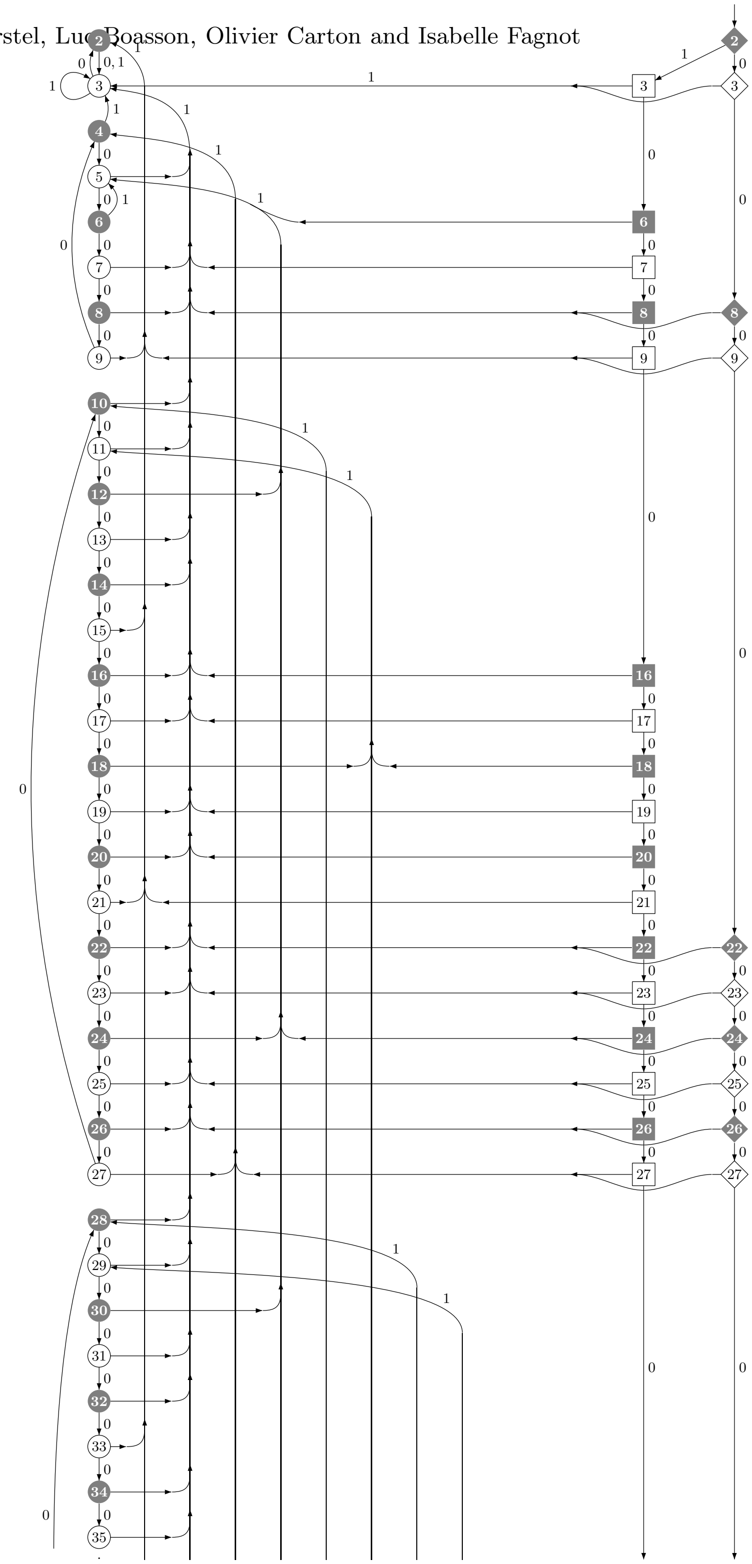

Fig. 11. Sturmian Tree of degree 2