Sturmian Trees

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

[1] Laboratoire d'Informatique de l'Institut Gaspard-Monge (LABINFO-IGM) UMR CNRS 8049, Université Paris-Est, Marne-la-Vallée
[2] Laboratoire d'informatique algorithmique: fondements et applications (LIAFA), Université Denis-Diderot (Paris VII) and CNRS, Paris

January 11, 2008

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 prove that a tree is Sturmian if and only if the minimal automaton associated to its language is slow, that is if the Moore minimization algorithm splits exactly one equivalence class at each step. We give various examples of Sturmian trees, and we introduce two parameters on Sturmian trees, called the degree and the rank. We show that there is no Sturmian tree of finite degree at least 2 and having finite rank. We characterize the family of Sturmian trees of degree 1 and having finite rank by means of a structural property of their minimal automata.

1 Introduction

Sturmian words have been extensively studied for many years (see e.g. [5, 6] for recent surveys). We propose here an extension to trees.

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 trees. This is due to the fact that, in our case, each node in a tree has two children, which provides more degrees of freedom. In particular, only one of the children of a node needs to be the root of a Sturmian tree to make the whole tree Sturmian.

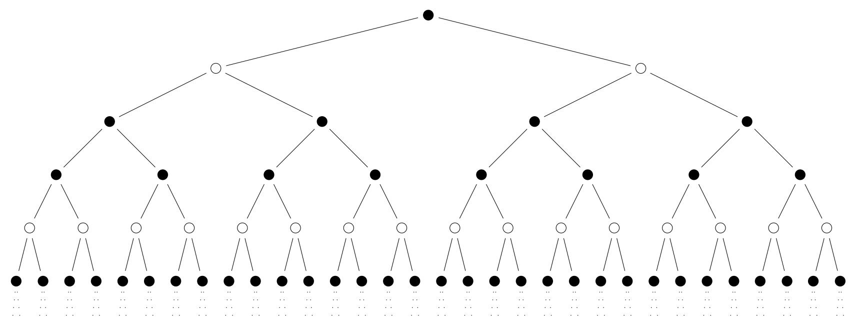

Fig. 1: The top of a uniform tree for the word $abaaba\cdots$. Node label a is represented by $\bullet$, and label b is represented by $\circ$. This tree will be seen to have infinite degree and rank 0.

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 trees: 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.

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.

Here is a table summarizing the relations between degree and rank for Sturmian trees. 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 14
≥ 2, finite	empty by Proposition 6	example 15
infinite	example of Dyck tree	example 14(a)

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 [2].

The paper is organized as follows. Section 2 contains the basic definitions, and the proof that a tree with fewer subtrees is rational. This was already known before, in particular in [2]. We give the easy proof for sake of completeness. Section 3 recalls several basic properties of automata, just observing that these properties also hold for infinite automata. In the case of slow automata, the way classes are split in the Moore equivalence algorithm is analyzed, for later use. The next section is devoted to Rauzy graphs. These are a useful construct for Sturmian words, and we show how they characterize Sturmian trees. As a byproduct, we obtain new insight in the nature of Rauzy graphs for Sturmian words: they are the graphs obtained in the Moore minimization process of an infinite automaton over a single letter alphabet.

In Section 5, we define the rank and the degree of a tree. In Section 6, we consider Sturmian trees with finite rank, that is with finitely many rational subtrees. We prove the main result, namely that Sturmian trees with finite rank and of degree one correspond precisely to extensions of automata obtained as an extension of a finite slow automaton by repeating some finite path infinitely many often, and by intertwining consecutive copies of this path with the successive letters taken from some Sturmian infinite word.

Section 7 is concerned with trees with infinite rank. It contains examples of Sturmian trees and of the corresponding automata showing that the general case might be quite intricate.

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

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

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.

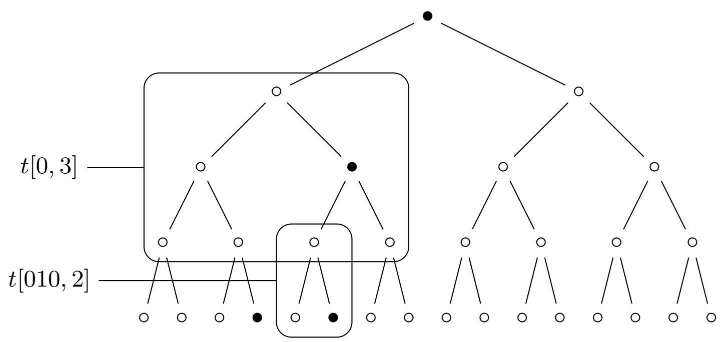

Fig. 2: The top of the Dyck tree of Example 1 and two of its factors, of height 3 and 2, respectively. Again, a is represented by $\bullet$ and b by $\circ$.

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 deal with the two-letter alphabet $A = \{a, b\}$, and we fix the letter a, and we simply call *language of t* its a-language.

We shall see that the language of a tree t is 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 $w^{-1}\,\mathrm{dom}(t)$ defined by $t[w](u) = t(wu)$ for each u in $w^{-1}\,\mathrm{dom}(t)$. The tree $t[w]$ is sometimes written as $w^{-1}t$, for instance in [2]. 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 $w^{-1}\,\mathrm{dom}(t) \cap D^{<h}$ 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. [2]). 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:

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

A tree is *rational* if it has finitely many distinct suffixes. Recall (see e.g. [3]) 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 [7]. The following proposition gives a characterization of complete rational trees using factors. It extends to trees the characterization of ultimately periodic words by means of their subword complexity [4]. This statement appears also in [2], we give the proof for sake of completeness.

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

Proof. It is clear that if t has k distinct suffixes, it has at most k distinct factors of height h for any h.

Conversely, we prove by induction on h that if t has at most h factors of height h, then t is rational. If $h = 1$, all nodes in t have the same label and clearly t is rational. Suppose now that $h > 1$ and that t has h factors $f_1, \ldots, f_h$ of height h. Each factor of height $h-1$ of t is a prefix $f_i[\varepsilon, h-1]$ of some factor f_i. If there are at most $h-1$ such factors, then t is rational by the induction hypothesis. Otherwise, each factor of height $h-1$ is the prefix of exactly one factor of height h. But this means that each suffix is determined by its prefix of height $h-1$. Consequently, t has h suffixes and therefore is rational. □

A tree is *Sturmian* if it is complete and if it has $h+1$ factors of height h for any integer 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 $\{a, b\}$.

Remark 1. In a Sturmian tree t, each subtree of height h has infinitely many occurrences. This does not mean that all infinite subtrees of t are Sturmian, see for instance the Dyck tree (Figure 2).

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 $x = x_0x_1x_2\cdots$ be an infinite word over an alphabet A, where $x_0, x_1, x_2, \ldots$ 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 the Fibonacci word $x = abaaba\cdots$.

Example 3. (*Left branch tree*) Let $x = x_0 x_1 x_2 \cdots$ be an infinite word over A, where $x_0, x_1, x_2, \ldots$ 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 leftmost branch of the tree, and all nodes on a rightmost branch share the same label. Figure 3 shows the top of the left branch tree of the Fibonacci word $x = abaaba \cdots$.

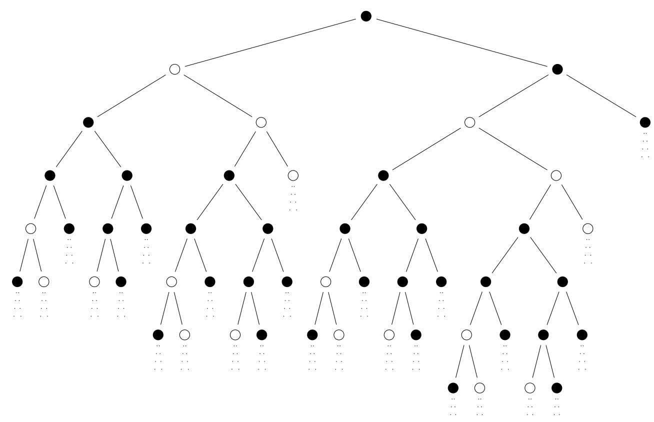

Fig. 3: The top of a left branch tree for the word $abaaba \cdots$.

We write $x[n, h]$ for the factor $x_n x_{n+1} \cdots x_{n+h-1}$ of the word x. In Example 2, 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 3, $t[w, h]$ and $t[w', h]$ are equal if and only if $x[|w|_0, h] = x[|w'|_0, h]$. 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 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.

The following example 5 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.

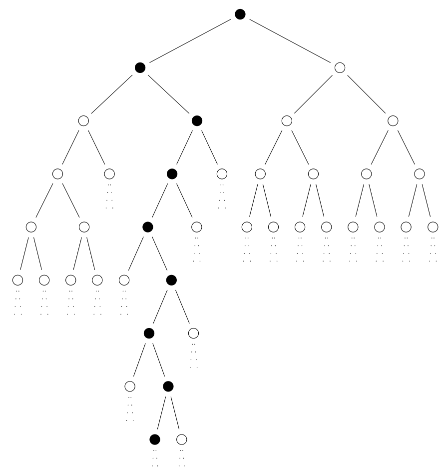

Fig. 4: The top of the indicator tree for the Fibonacci word $01001010\cdots$. The only nodes labeled a are on the Fibonacci path.

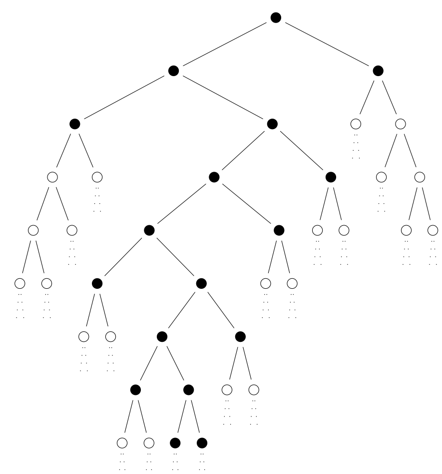

Fig. 5: The top of the band indicator tree of width 1 for the Fibonacci word $01001010\cdots$. The nodes labeled a are at distance at most 1 from the Fibonacci path.

Example 5. (*Band indicator tree*) Let $x = x_0x_1x_2\ldots$ 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

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

The same argument shows that x is a Sturmian word if and only if t is a Sturmian tree. An example is given in Figure 5. The band indicator tree of width 0 is the indicator tree defined in Example 4, since $d(w, x) \leq 0$ if and only if w is a prefix of x.

3 Slow automata

Let t be a complete tree over $\{a, b\}$. The *language* of t is the set $t^{-1}(a)$. We study properties of trees by considering automata recognizing their language. In particular, minimization of automata will play a central role.

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 $q \cdot d$. The next state function is extended to a function from $Q \times D^*$ to Q by setting $q \cdot \varepsilon = q$ and $q \cdot wd = (q \cdot w) \cdot d$ for a word $w \in D^*$ and a letter $d \in D$. 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.

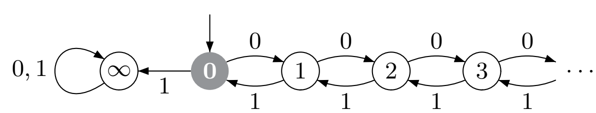

Fig. 6: Automaton of the Dyck language. State 0 is both the initial and the unique final state.

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

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

These are well-known in the case of finite automata, and many properties extend to general automata. We call $\sim_h$ the *Moore equivalence* of order h. The *index* of $\sim_h$ is the number of equivalence classes of $\sim_h$. The Moore minimization algorithm consists in computing inductively the Moore equivalences.

Example 7. We illustrate this definition on the Dyck automaton given in Example 6.

By definition the relation $\sim_1$ has two classes: one contains all the final states and the other contains all other states. Here, these classes are $\{0\}$ and $Q \setminus \{0\}$. One easily checks that the classes of $\sim_2$ are the three sets $\{0\}, \{1\}$ and $Q \setminus \{0, 1\}$. It is easily shown by induction that the classes of $\sim_h$ are the $h + 1$ sets $\{0\}, \{1\}, \ldots, \{h - 1\}$ and $Q \setminus \{0, 1, \ldots, h - 1\}$.

We define the *Nerode equivalence* relation $\sim$ by $\sim = \bigcap_{h \geq 1} \sim_h$. It is clear that $q \sim q'$ holds if and only if $L_{\mathcal{A}}(q) = L_{\mathcal{A}}(q')$ where $L_{\mathcal{A}}(q)$ denotes the set of words accepted by $\mathcal{A}$ with initial state q. We state here a few properties of the relations $\sim_h$ and $\sim$. They are well-known for finite automata but hold also in the general case

- Each relation $\sim_h$ has finite index.
- The relation $\sim$ has finite index iff $\sim$ equals $\sim_h$ for some integer h.
- If $\sim_{h+1} = \sim_h$ for some integer h, then $\sim_{h+k} = \sim_h$ for each $k \geq 0$ and $\sim = \sim_h$.
- If $\sim_h$ has index at most h, then $\sim_h = \sim$.
- The relation $\sim$ is a congruence. This means that for any states q and q' and any letter $d \in D$, $q \sim q'$ implies $q \cdot d \sim q' \cdot d$.

An automaton is *minimal* if its Nerode equivalence $\sim$ is the equality relation. Many results concerning minimal finite automaton remain valid for infinite ones. For each subset L of D^*, there is a unique minimal automaton accepting it. This minimal automaton is equal to $\mathcal{A}/\sim$ for any automaton $\mathcal{A}$ accepting L. In particular, it can be obtained from the automaton based on the complete binary tree as described above.

A subautomaton $\mathcal{B}$ of a minimal automaton $\mathcal{A}$ is itself minimal. The Nerode equivalence of $\mathcal{B}$ is the restriction of the Nerode equivalence of $\mathcal{A}$ to the set of states of $\mathcal{B}$.

Lemma 1. *Let $\mathcal{A}$ be an automaton. For any states q, q' and any positive integer h, one has*

$$q \sim_h q' \iff (\forall u \in D^{<h} \;\; q \cdot u \in F \iff q' \cdot u \in F)\,.$$

Proof. The proof is an easy induction on h. Since $D^{<1} = \{\varepsilon\}$, the case $h = 1$ follows from the definition of $\sim_1$. The induction step follows then from the equality $D^{<h+1} = D^{<h} \cup DD^{<h}$. □

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 Dyck automaton of example 7, the state ∞ is the only rational state. 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.

In a slow automaton, the index of each $\sim_h$ is exactly $h+1$, as long as $\sim_h$ is different from $\sim_{h-1}$. As soon as these relations are equal, they are the identity relation.

Given an infinite slow automaton $\mathcal{A}$, we denote, for $h \geq 0$, by c_h the class of $\sim_h$ that is split by $\sim_{h+1}$ and, for $h > 0$, by c'_h and c''_h the classes of $\sim_h$ obtained by splitting c_{h-1} in $\sim_h$.

The following easy property of slow automata is useful.

Lemma 2. *Let $\mathcal{A}$ be an infinite slow automaton. For $h > 0$, there exists a letter a_{h+1} that maps all states of c'_{h+1} into c'_h (or all states into c''_h) and all states of c'_{h+1} into the other of the classes c'_h or c''_h.*

Proof. Consider indeed $c_h = c'_{h+1} \cup c''_{h+1}$. There exist states $q' \in c'_{h+1}$ and $q'' \in c''_{h+1}$ which are equivalent for $\sim_h$ but which are not equivalent for $\sim_{h+1}$. By definition, this means that there is a letter $a = a_{h+1}$ such that $q' \cdot a \not\sim_h q'' \cdot a$, but $q' \cdot a \sim_{h-1} q'' \cdot a$. In other terms, the class of $\sim_{h-1}$ containing both states $q' \cdot a$ and $q'' \cdot a$ is split by $\sim_h$ into classes c'_h and c''_h. Furthermore, the letter a maps all states of c'_{h+1} into c'_h (or into c''_h) and all states of c''_{h+1} into the other of these two classes c'_h and c''_h, as claimed. □

Observe that the lemma does not say anything about the letters distinct from a_{h+1}. However, since the automaton is slow, they also map the classes c'_{h+1} and c''_{h+1} into classes of $\sim_h$.

Corollary 1. *Let $\mathcal{A}$ be an infinite slow automaton. There exists an infinite sequence of pairs (c'_h, c''_h) of classes of $\sim_h$ such that each $c'_h \cup c''_h$ is the equivalence class of $\sim_{h-1}$ split by $\sim_h$, and letters a_h, such that each a_h maps all states of c'_h into states of c'_{h-1}, and all states of c''_h into states of c''_{h-1}.*

Proof. This is a direct consequence of the previous lemma. □

Remark that for this corollary also the other letters of the alphabet do not split other classes.

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

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

Proof. Set $q = i \cdot w$ and $q' = i \cdot w'$.

$$\begin{aligned} q \sim_h q' &\iff (\forall u \in D^{<h} \;\; q \cdot u \in F \iff q' \cdot u \in F) \\ &\iff (\forall u \in D^{<h} \;\; t(wu) = a \iff t(w'u) = a) \\ &\iff t[w,h] = t[w',h] \,. \end{aligned}$$

□

Corollary 2. *Let t be a complete tree and let $\mathcal{A}$ be an automaton over D accepting the language of t. The tree t is Sturmian if and only if the minimal automaton of its language is infinite and slow.*

Example 8. We have seen in Example 7 that the relation $\sim_h$ for the Dyck automaton of Example 6 has $h+1$ classes. Thus this automaton is slow and the Dyck tree given in Example 1 is Sturmian. The automaton of Figure 6 is just one special case of an uncountable family of slow automata defined in a quite similar way and which provide an uncountable family of Sturmian trees.

These automata differ from the Dyck automaton by the behavior of the next-state function for the letter 0: instead of having $n \cdot 0 = n+1$ for $n \geq 1$, we require only that $n \cdot 0 \geq n$, and in addition that each state n is accessible from the initial state 0. An example is given in Figure 7. It is easily checked that again that the classes of $\sim_h$ are the $h+1$ sets $\{0\}, \{1\}, \ldots, \{h-1\}$ and $Q \setminus \{0, 1, \ldots, h-1\}$. So these automata are all slow.

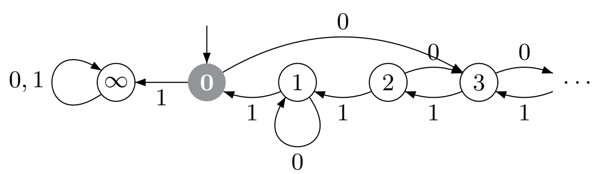

Fig. 7: Automaton of an extension of the Dyck language.

We consider now the automata for the examples of Sturmian trees of the previous section.

Example 9. (*Indicator tree*) We consider the automaton of an indicator tree, as defined in Example 4. 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

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

Recall that for $d \in D$, we write $\bar{d}$ for the opposite letter, that is $\bar{0} = 1$ and $\bar{1} = 0$. Let $x = x_0x_1x_2\ldots$ be an infinite words over D and let us consider the tree t_x defined in the previous example. The minimal automaton accepting the language $t_x^{-1}(a)$ has set of states $Q = \mathbb{N} \cup \{\infty\}$, with initial state $i = 0$ and set of final states $F = \mathbb{N}$. The next-state function is given by $n \cdot x_n = n+1$, $n \cdot \bar{x}_n = \infty$ and $\infty \cdot 0 = \infty \cdot 1 = \infty$. Figure 8 shows the automaton accepting the prefixes of the Fibonacci word $x = 01001010\cdots$

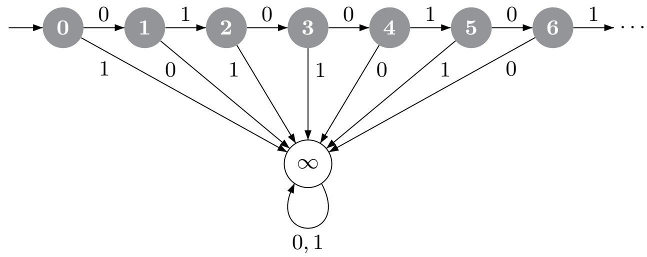

Fig. 8: Automaton accepting the prefixes of $01001010\cdots$. All states excepted ∞ are final.

By Theorem 1 below, indicator trees of Example 4 are Sturmian for Sturmian words. However, indicator trees are sufficiently simple to allow a direct proof. By Corollary 2, it suffices to show that each equivalence relation $\sim_h$ on the automaton of the previous example has $h+1$ classes. The classes of $\sim_1$ are the two sets $\mathbb{N}$ and $\{\infty\}$. Let h be a positive integer. We claim that two states n and n' satisfy $n \sim_h n'$ if and only if $x_n \cdots x_{n+h-2} = x_{n'} \cdots x_{n'+h-2}$. Indeed, a word $u \in D^{<h}$ satisfy $n \cdot u \in F$ if and only if u is a prefix of $x_n \cdots x_{n+h-2}$. For a word w of length k over D, we denote by N_w the set $\{n \mid x_n \cdots x_{n+k-1} = w\}$. Note that N_w is non empty if w is a factor of x. It follows then that the classes of $\sim_h$ are the set $\{\infty\}$ and the h sets of the form N_w where w is a factor of length $h-1$ of x. Therefore, the tree t_x of Example 4 is Sturmian if and only if x is Sturmian.

Example 10. We consider the automaton for a band indicator tree, as defined in Example 5.

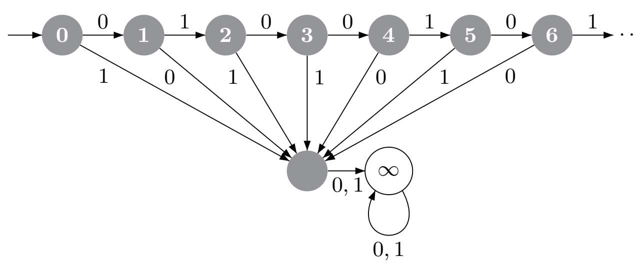

Fig. 9: Automaton of the band indicator tree of width 1 for the Fibonacci word $01001010\cdots$. Only one state is not final.

Example 11. 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. Indeed, it is known that the set of Sturmian words having exactly the same fixed set of factors is a minimal subshift, and since it is infinite, it is uncountable. On the other side, for each (of the countably many) suffix of an infinite word, there are only countably many infinite words sharing this suffix.

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 ∞.

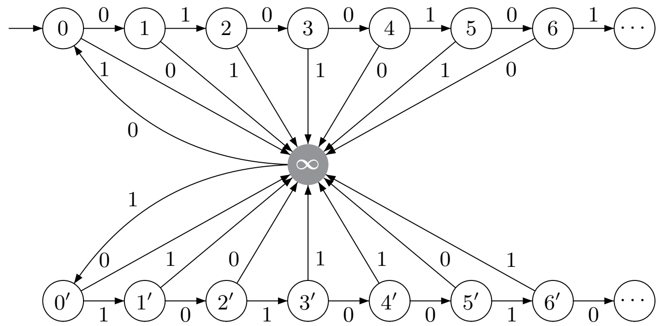

Fig. 10: Automaton of $t_{x,x'}$ for $x = 01001010\cdots$ and $x' = 10100100\cdots$

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, in the sense defined later.

4 Rauzy graphs

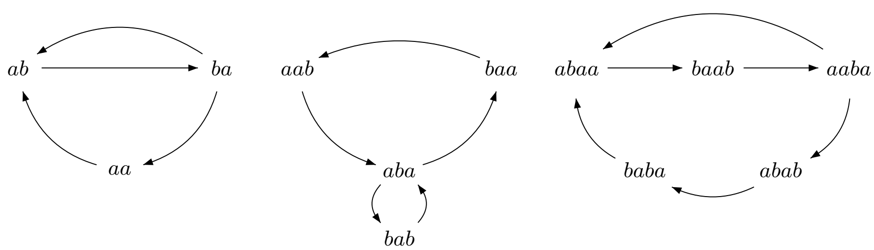

Fig. 11: Three Rauzy graphs for the Fibonacci word.

We define here the Rauzy graphs of a tree. This notion extends to trees the Rauzy graphs of words [1]. Recall that, for an infinite word x, the *Rauzy graph* of order h, or the *Rauzy graph* for short, is a graph whose vertices are the factors of length h of x, and whose edges are in bijection with the factors of length $h+1$ of x as follows. The pair (u, v) is an edge if $u = cw$, $v = wd$ for some letters c, d, and $ud = cv$ is a factor of x, see Figure 11. Rauzy graphs are special de Bruijn graphs.

The h-Rauzy graph of an infinite word x can be viewed as the quotient of an infinite automaton. Define indeed an infinite automaton $\mathcal{A}(x) = (\mathbb{N}, 0, F)$ over a

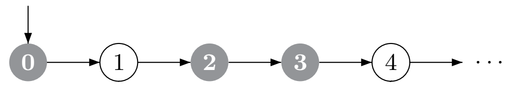

Fig. 12: Automaton for the Fibonacci infinite word f. A state i is final if $f_i = a$. The state 0 is initial. Final states are gray.

one letter alphabet with next-state function $n \mapsto n+1$, and set of final states $F = \{n \mid x_n = a\}$. This automaton recognizes the set of prefixes of x ending with the letter a, see Figure 12. The Moore equivalence $\sim_h$ of order h of $\mathcal{A}(x)$ satisfies

$$n \sim_h n' \quad \text{if and only if} \quad [n+i \in F \Leftrightarrow n'+i \in F \ (0 \le i < h)]$$

thus $n \sim_h n'$ if and only if the factors $x(n, n+h)$ and $x(n', n'+h)$ of x are equal. The quotient $\mathcal{A}(x)/{\sim_h}$ has its vertices in bijection with the factors of length h of x, and there is an edge (u, v) precisely if $u = x(n, n+h)$ and $v = x(n+1, n+1+h)$ for some n. This shows that the quotient $\mathcal{A}(x)/{\sim_h}$ is precisely the Rauzy graph of x. Observe that if x is Sturmian, there is exactly one class of $\sim_h$ that splits into two classes of $\sim_{h+1}$. This corresponds to the unique right special factor of length h, i. e. to the state in $\mathcal{A}(x)/{\sim_h}$ which has two outgoing states

The situation for trees is more involved. Roughly speaking, the vertices of the h-Rauzy graph of a tree are the factors of height h and its edges are the factors of height $h+1$. It turns out that these graphs are in fact hypergraphs since each edge is a triple of vertices.

Let t be a complete tree. For an integer h, the *h-Rauzy graph* of t is a graph whose vertices are the factors of height h of t. Its edges are triples of the form $(t_\varepsilon, t_0, t_1)$ where t_ε, t_0 and t_1 are three vertices, that is, three factors of height h of t. Each factor f of height $h+1$ of t, define an edge $(t_\varepsilon, t_0, t_1)$ by $t_\varepsilon = f[\varepsilon, h]$, $t_0 = f[0, h]$ and $t_1 = f[1, h]$. The edges of the h-Rauzy graph are in bijection with the factors of height $h+1$ of t, since a factor of height $h+1$ is entirely defined by its three factors of height h.

An edge $(t_\varepsilon, t_0, t_1)$ of a Rauzy graph is drawn as in Figure 13. If the tree t is

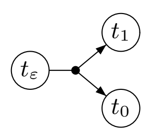

Fig. 13: An edge $(t_\varepsilon, t_0, t_1)$ of a Rauzy graph. The factor t_ε is the prefix of some factor f of height $h+1$, and t_0 and t_1 are the left and right subtrees of height h of f.

Sturmian, its h-Rauzy graph has $h+1$ vertices and $h+2$ edges. Furthermore, since each factor of height h is the prefix of at least one factor of height $h+1$, there is at least one edge starting in each vertex. Consequently, there is precisely one vertex which has two edges starting in it.

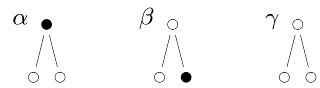

Fig. 14: The three factors of height 2 of the Dyck tree, named α, β and γ.

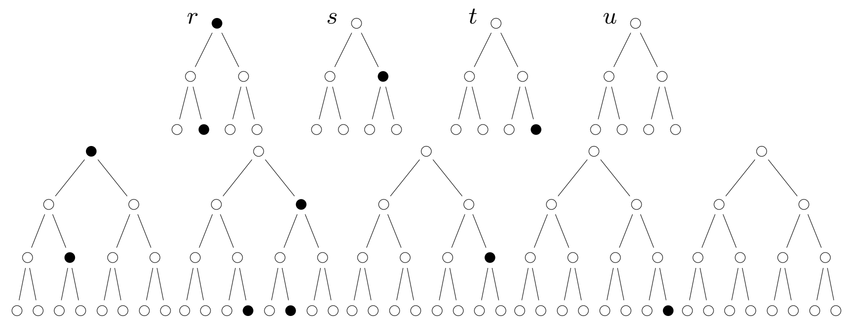

Fig. 15: The four factors of height 3, and the five factors of height 4 of the Dyck tree.

Example 12. Let us consider the Dyck tree given in Example 1. Its factors of height 2, 3 and 4 are shown in Figure 14 and 15. Its 1, 2 and 3-Rauzy graphs are given in Figure 16.

The states of the 1-Rauzy graphs are the two factors of height 1, that is the letters a and b. The edges of the 2-Rauzy graph are the three factors of height 2 named α, β and γ for convenience.

The states of the 2-Rauzy graph are the three factors α, β and γ, and the edges are the four factors of height 3 are named r, s, t, and u.

The states of the 3-Rauzy graph are the four factors r, s, t, and u, and its edges are the five factors given in the second row of Figure 15

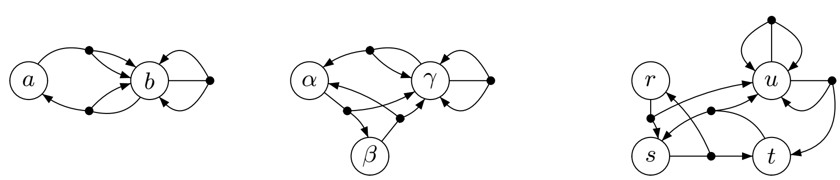

Fig. 16: The 1, 2, and 3-Rauzy graphs of the Dyck tree. There is exactly one edge starting in each vertex, excepted for b, γ and u.

Let t be a complete tree over $\{a, b\}$ and let $\mathcal{A}$ be an automaton accepting $t^{-1}(a)$. Let h be an integer. We construct a hypergraph $\mathcal{A}/{\sim_h}$ as follows. The vertices of $\mathcal{A}/{\sim_h}$ are the equivalence classes of $\sim_h$. The edges of this graph are the triples of the form $([q], [q \cdot 0], [q \cdot 1])$ where q ranges over the states of $\mathcal{A}$ and $[q]$ denotes the class of q.

Proposition 3. *Let t be a complete tree. The graph $\mathcal{A}/{\sim_h}$ is the h-Rauzy graph of t.*

Proof. By Proposition 2, there is a one to one correspondence between vertices of $\mathcal{A}/{\sim_h}$ and vertices of the h-Rauzy graph. Furthermore, two states satisfy $[q] = [q']$, $[q \cdot 0] = [q' \cdot 0]$, and $[q \cdot 1] = [q' \cdot 1]$ iff they satisfy $q \sim_{h+1} q'$. Again by Proposition 2, there is a one to one correspondence between edges of $\mathcal{A}/{\sim_h}$ and edges of the h-Rauzy graph. □

5 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. It is called *irrational* otherwise. 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.

In a Sturmian tree t, a node w is irrational if and only if $t[w]$ is a Sturmian tree.

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. If the degree is 0, the tree t is rational.

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 Section 7, we construct a Sturmian tree of degree 2 but this construction is rather involved.

Here is a table summarizing the relations between degree and rank for Sturmian trees. A tree with rank 0 always has infinite degree since there is no rational node.

degree	rank: finite	rank: infinite
1	*characterized in Theorem 1* Indicator tree (rank 1) Band width tree (rank $d+1$)	example 14
≥ 2, finite	*empty by Proposition 6*	example 15
infinite	Uniform tree (rank 0) Left branch tree (rank 0) Dyck tree (rank 1)	example 14(a)

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.

6 Trees with finite rank

We start with an observation concerning the occurrences of subtrees of fixed height in a Sturmian tree of finite rank.

Let t be a Sturmian tree. For any $h \geq 0$, each of the $h+1$ factor trees of height h of t has infinitely many occurrences in t. Indeed, assume the contrary. Then there is an integer N such that some factor tree of height h does not occur in the subtrees $t[w]$ with $|w| \geq N$. But then these subtrees are all rational and t is rational, contradiction.

We now turn to the question of how these infinitely many occurrences are distibuted in a Sturmian tree. This question arises as a natural extension of the same question concerning Sturmian words: one knows that every factor of a Sturmian word appears infinitely times (words sharing this property are called *recurrent*), and moreover the difference between consecutive occurrences of the same factor in a Sturmian word is bounded, for each factor (words with this property are called *uniformly recurrent*). For Sturmian trees, it is clearly false that every factor tree of height h appears on every infinite branch, because there exist infinite branches of rational nodes, as soon as the rank of the tree is positive. However, we have the following property which shows that there exist branches which are in some sense recurrent.

Proposition 4. *In any Sturmian tree of finite rank, there exists an infinite branch composed of irrational nodes such that every factor tree having its root on this branch appears infinitely many times on this branch.*

Proof. Let r be the rank of the Sturmian tree t and let $t^{(1)}, \ldots, t^{(r)}$ be the distinct rational subtrees of t. For each irrational node w, the irrational tree $t[w]$ rooted at w is distinct from $t^{(1)}, \ldots, t^{(r)}$. Therefore, there exists an integer H_w such that the prefix $t[w, H_w]$ of height H_w of $t[w]$ differs from the prefixes of height H_w of $t^{(1)}, \ldots, t^{(r)}$.

We claim that for every pair v, v' of irrational nodes, and for every $h > 0$, there exists an irrational node v'' such that v'' is a proper descendent of v' and

$t[v, h] = t[v'', h]$. Indeed, let $H \geq \max(h, H_v)$, where H_v is defined as above. Since v' is irrational, there exists a proper descendent v'' of v' such that $t[v'', H] = t[v, H]$. Since $H \geq H_v$, the tree $t[v'', H]$ is not a prefix of one of the rational trees. Thus v'' is an irrational node. Since $H \geq h$, one has $t[v, h] = t[v'', h]$.

The proposition is derived from the claim by constructing an infinite branch as follows, for $h = 1, 2, \ldots$. For $h = 1$ the claim (with $v = v' = \varepsilon$) shows that there exist an irrational node $w \neq \varepsilon$ such that $t(\varepsilon) = t(w)$. Denote by $z_0 = \varepsilon, z_1, \ldots, z_n = w$ the path from ε to w. For $h = 1$, consider the factor trees $t[z_0, h]$ and $t[z_1, h]$. By the claim, there exists path $z_n, z_{n+1}, \ldots, z_{n+m}$ to an irrational node such that $t[z_0, h] = t[z_{n+m}, h]$, and a path $z_{n+m}, z_{n+m+1}, \ldots, z_{n+m+m'}$ to an irrational node $z_{n+m+m'}$ such that $t[z_1, h] = t[z_{n+m+m'}, h]$.

The general step is as follows. One considers the h factor trees $t[z_0, h], t[z_1, h], \ldots, t[z_{h-1}, h]$ that are rooted on the first h nodes of the irrational path $z_0, \ldots, z_{m_h}$ already constructed. For each of these factor trees, one extends the irrational path by paths $z_{m_h} \to z_{m_h+p_0}$, $z_{m_h+p_0} \to z_{m_h+p_1}, \ldots, z_{m_h+p_{h-2}} \to z_{m_h+p_{h-1}}$ in such a way that $t[z_0, h] = t[z_{m_h+p_0}, h]$, $t[z_1, h] = t[z_{m_h+p_1}, h], \ldots, t[z_{h-1}, h] = t[z_{m_h+p_{h-1}}, h]$. In this way, the irrational path has been extended from node z_{m_h} to node $z_{m_h+p_{h-1}}$, and on this path there is a copy of each of the factor trees of height h which are rooted on the first h nodes of the irrational path. This ensures that all factor trees of height at most h that are rooted on the the first h nodes of the path appear at least twice, and in fact infinitely often when this construction is repeated. This concludes the proof. □

Observe that we do not claim that for every infinite branch composed of irrational nodes, every subtree of height h appears infinitely many times (of course, at least one does so). Observe also that our proof does not show that subtrees of height h appear with bounded gaps. So we have shown that the subtrees appear as factors in a recurrent word, but we do not know if they may appear in some uniformly recurrent manner.

We shall see later (Example 16) an example showing that the proposition does not hold if the tree has infinite rank.

6.1 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. These trees are generalizations of the band indicator trees of Example 5. We describe the family of automata accepting their languages. These (infinite) automata are based on a finite slow automaton. In this automaton, a path (called a lazy path) is distinguished. The infinite automaton is obtained by repeating the lazy path and intertwining the copies with symbols taken from an infinite Sturmian word, as we will see below.

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

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

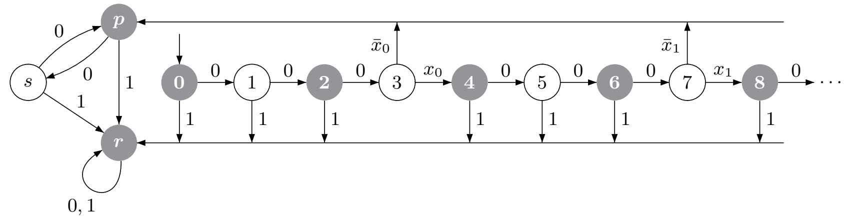

Fig. 17: A slow automaton $\hat{\mathcal{A}}$ for the Fibonacci word $x_0x_1\cdots = 01001010\cdots$. The final states are $p, r, 0, 2, 4, \cdots$.

that the Moore minimization algorithm splits just one equivalence class into two new classes at each step.

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

$$\pi : q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \cdots q_{h-1} \xrightarrow{a_{h-1}} q_h$$

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

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

where $\bar{a} = 1 - a$ for $a \in D$. If $N \geq 2$, 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 if the states $q_{h-1} \cdot \bar{a}_{h-1}$ and $q_{h-1} \cdot a_{h-1}$ are distinct, they cannot be separated before the very last step of the Moore algorithm.

Example 13. The automaton $\hat{\mathcal{A}}$ given in Figure 17 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 17 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 automaton $\hat{\mathcal{A}}$ which accepts the set of nodes labeled a of a tree t. The automaton $\hat{\mathcal{A}}$ will be minimal as soon as x is not eventually periodic. 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 $q : \mathbb{N} \to \{q_0, \ldots, q_{h-1}\}$ defined by $q(n) = q_{n \bmod h}$ for any $n \in \mathbb{N}$. Here and below $q_0, \ldots, q_h$ are the states of the lazy path of $\mathcal{A}$ and $a_0, \ldots, a_{h-1}$ are the letters labeling the path. 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 $n \cdot a_{n \bmod h} = n+1$ and $n \cdot \bar{a}_{n \bmod h} = q(n) \cdot \bar{a}_{n \bmod h}$,
(β) if $n = ih + h - 1$ for some $i \geq 0$, then $n \cdot x_i = n + 1$ and $n \cdot \bar{x}_i = q_0$.

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 (see Figure 17).

Proposition 5. *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. Its rank is the number of states of the automaton $\mathcal{A}$. 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 be 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, in the three following lemmas x is not required to be Sturmian.

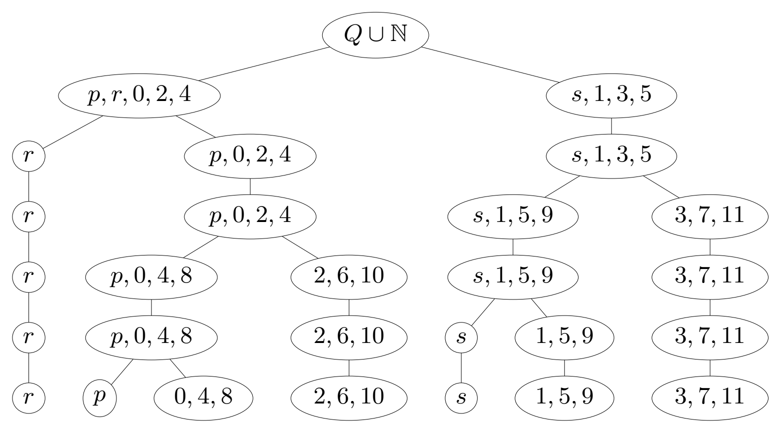

Fig. 18: The tree showing the evolution of the Moore equivalence relations on the automaton given in Figure 17. Each level describes a partition. Each level has one class splitting into two classes at the next level.

Lemma 3. *For $1 \le k \le N-1$ and any $n \ge 0$, one has $n \sim_k q(n)$.*

Proof. By induction on k. The result clearly holds for $k = 1$ since the set of final states is $F \cup q^{-1}(F)$. Assume that $n \sim_k q(n)$ for any $n \ge 0$ and $k \le N-2$. We prove that $n \sim_{k+1} q(n)$ holds for any $n \ge 0$.

If $n \not\equiv h-1 \mod h$, then setting $a = a_{n \bmod h}$, one has $n \cdot a = n+1 \sim_k q(n+1) = q(n) \cdot a$ and $n \cdot \bar{a} = q(n) \cdot \bar{a}$. From $n \sim_k q(n)$, it follows that $n \sim_{k+1} q(n)$.

Suppose now that $n = ih + h - 1$ for some $i \geq 0$. One has $n \cdot x_i = n + 1 \sim_k q_0$, $n \cdot \bar{x}_i = q_0$ and $q_{h-1} \cdot x_i, q_{h-1} \cdot \bar{x}_i \in \{q_0, q_h\}$. Furthermore, $q_0 \sim_k q_h$ because $k \leq N - 2$. This, together with $n \sim_k q(n)$, shows that $q(n) \sim_{k+1} q_{h-1}$. □

Example 13 (*continued*). In our example, one has $h = 4$ and $q(n) = p$ if $n \equiv 0, 2 \bmod h$, and $q(n) = s$ otherwise. Also, the first two Moore equivalence classes group together integer states with their images by the mapping q. Observe that each class in $\sim_2$ contains exactly one state of the $\{p, s, r\}$.

Lemma 4. *Let $k, \ell \geq 0$ be integers with $k + \ell = h - 1$.*

1. *If $(n \bmod h) < \ell$, then $n \sim_{N+k} q(n)$.*
2. *If $(n \bmod h) \geq \ell$, then $n \not\sim_{N+k} p$ for any $p \in Q$, and $n \sim_{N+k} n'$ iff $n \equiv n' \bmod h$.*

Example 13 (*continued*). In our example, the next h steps (with $h = 4$) split away classes of integer states, in the decreasing order of their labels mod 4: first the states $n \equiv 3 \bmod h$, then the states $n \equiv 2 \bmod h$ and so on. After these steps, the states p, s, r are singleton classes.

Proof. of Lemma 4. The proof is by induction on k.

First, assume $k = 0$, that is $\ell = h - 1$.

If $0 \leq n \bmod h \leq h - 2$, then $n \not\equiv h - 1 \mod h$. By the previous lemma, one has $n \sim_{N-1} q(n)$. Setting $a = a_{n \bmod h}$ for short, one has $n \cdot a = n + 1 \sim_{N-1} q(n+1) = q(n) \cdot a$ and $n \cdot \bar{a} = q(n) \cdot \bar{a}$. This proves that $n \sim_N q(n)$.

If $n = ih + h - 1$, then for the first claim, it suffices to prove that $n \not\sim_N q(n)$. To show this, observe that the letter a_{h-1} separates n and $q(n) = q_{h-1}$. Indeed, first one has $q_{h-1} \cdot a_{h-1} = q_h$ and second, $n \cdot x_i = n + 1 \sim_{N-1} q_0$ and $n \cdot \bar{x}_i = q_0$, Consequently, $n \cdot a_{h-1} \sim_{N-1} q_0 \not\sim_{N-1} q_h$. This proves the first claim.

For the second claim, consider $n \equiv n' \mod h$ and set $n' = h' + h - 1$. Then $n \sim_{N-1} n'$ by the previous lemma, and moreover $n \cdot x_i = n + 1 \sim_{N-1} q_0$, $n \cdot \bar{x}_i = q_0$, and $n' \cdot x_{i'} = n' + 1 \sim_{N-1} q_0$, $n' \cdot \bar{x}_{i'} = q_0$. This proves that $n \sim_N n'$. Suppose conversely that $n \not\equiv h - 1 \mod h$. One has $n' \sim_N q(n')$ by claim (1) and $n \not\sim_N p$ for any $p \in Q$. Thus $n \not\sim_N n'$. This proves that $n \sim_N n'$ iff $n \equiv n' \bmod h$.

Assume now that $k > 0$ and that the result holds for $k - 1$. If n is such that $n \bmod h < \ell$, then $n + 1 \bmod h < \ell + 1$ and by the induction hypothesis, one has $n + 1 \sim_{N+k-1} q(n+1)$. Setting $a = a_{n \bmod h}$ for short, one has $n \cdot a = n + 1 \sim_{N+k-1} q(n+1)$ and $n \cdot \bar{a} = q(n) \cdot \bar{a}$. Since $n \sim_{N+k-1} q(n)$ holds by induction, this shows that $n \sim_{N+k} q(n)$.

Let n be such that $\ell \leq n \bmod h$. If $\ell + 1 \leq n \bmod h$, then $n \not\sim_{N+k-1} p$ for each $p \in Q$ which implies that $n \not\sim_{N+k} p$ for each $p \in Q$. In the last case $n \bmod h = \ell$, it suffices to show that $n \not\sim_{N+k} q(n)$. For this, observe that by the induction hypothesis, one has $n + 1 \not\sim_{N+k-1} q(n+1)$. Next, $n \cdot a = n + 1$ and $q(n) \cdot a = q(n+1)$ where $a = a_{n \bmod h}$. This proves that $n \not\sim_{N+k} q(n)$.

Suppose now that $n \equiv n' \mod h$ and set again $n = ih + h - 1$, $n' = i'h + h - 1$. If $n \equiv h - 1 \mod h$, then $n \cdot x_i = n + 1$, $n \cdot \bar{x}_i = q_0$, $n' \cdot x_{i'} = n' + 1$ and $n \cdot \bar{x}_{i'} = q_0$, and it follows from $n \sim_{N+k-1} n'$ and $n+1 \sim_{N+k-1} n'+1 \sim_{N+k-1} q_0$ that $n \sim_{N+k} n'$.

If on the contrary $n \not\equiv h-1 \mod h$, then $n \cdot a = n+1$, $n' \cdot a = n'+1$ and $n \cdot \bar{a} = n' \cdot a = q(n+1)$, and it follows from $n \sim_{N+k-1} n'$ and $n+1 \sim_{N+k-1} n'+1$ that $n \sim_{N+k} n'$.

In order to prove the converse, suppose now that $n \sim_{N+k} n'$. Set $m = n \bmod h$ and $m' = n' \bmod h$. Assume that $m \neq m'$ and without loss of generality $m < m'$. If $m' < \ell$, then by claim (1), $n \sim_{N+k} q(n)$ and $n' \sim_{N+k} q(n')$, a contradiction because $q(n) \not\sim_N q(n')$. Thus $m' \geq \ell$. Set $\ell' = m'$ and $k' = h - \ell' - 1$. Then $k' \leq k$ and $m < \ell' = m'$. Therefore, $n \sim_{N+k'} q(n)$ and $n' \not\sim_{N+k'} p$ for any $p \in Q$. This implies $n \not\sim_{N+k'} n'$ and also $n \not\sim_{N+k} n'$ since $k' \leq k$. □

Lemma 5. *Let $k, \ell \geq 0$ be integers with $k + \ell = h - 1$. Let $n \equiv n' \mod h$ and set $i = \lfloor n/h \rfloor$ and $i' = \lfloor n'/h \rfloor$. For all integer $j \geq 1$,*

$$n \sim_{N+jh+k} n' \iff \begin{cases} x_i \cdots x_{i+j-2} = x_{i'} \cdots x_{i'+j-2} & \text{if } (n \bmod h) < \ell \\ x_i \cdots x_{i+j-1} = x_{i'} \cdots x_{i'+j-1} & \text{otherwise.} \end{cases}$$

The proof is very similar to that of the previous lemma and is therefore omitted.

Example 13 (*continued*). In our example, the infinite word is the Fibonacci word $x_0 x_1 \cdots = 01001 \cdots$ and $h = 4$. The first partition is into states $ih + h - 1$ with $x_i = 0$ and with $x_i = 1$. After 4 steps, the next partition is into states $ih + h - 1$ with $x_i x_{i+1} = 00$, with $x_i x_{i+1} = 01$ and with $x_i x_{i+1} = 10$.

Proof. of Proposition 5. It suffices to count the number of trees of a given height in t, and by Corollary 2, it suffices to show that $\hat{\mathcal{A}}$ is slow. For this purpose, we compute the index of $\sim_n$. By Lemma 3, the index of $\sim_{N-1}$ is N, by Lemma 4 the index of $\sim_{N+h-1}$ equals $N + h$, and by Lemma 5, the index of the equivalence $\sim_{N+jh+h-1}$ is equal to $N + hc_x(j)$ where $c_x(j)$ is the number of distinct factors of length j in x. If x is Sturmian, then $c_x(j) = j + 1$, and thus the index of $\sim_{N+jh+h-1}$ is $N + h(j+1)$. We have shown that the index of $\sim_n$ is $n+1$ for infinitely many n, and this implies immediately that the index is $n + 1$ for all n. □

6.2 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. $\hat{\mathcal{A}} = \mathcal{A}(\pi, x)$.*

We start with a general lemma which will also be of use later. Consider a tree t having finite rank. Let $\hat{\mathcal{A}}$ be the minimal automaton of the language of t. Each of the finitely many distinct rational subtrees of t contributes a state to this automaton. Let Q be the finite set of states corresponding to rational subtrees. Recall that a state in Q is what we called a *rational* state. The next-state function maps rational states into rational states, so that the restriction of $\hat{\mathcal{A}}$ to the set Q is a subautomaton that we denote by $\mathcal{A}$. This subautomaton is itself minimal.

Given a tree t and some Moore equivalence $\sim_h$ on its minimal automaton, it is convenient to call an equivalence 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 many irrational states.

Lemma 6. *Let $\hat{\mathcal{A}}$ be the minimal automaton of some (not necessarily Sturmian) tree t. Assume that, for some integer h, the equivalence $\sim_h$ only splits irrational classes. Then for all $h' \geq h$, $\sim_{h'}$ only splits irrational classes.*

Proof. Let Q be the set of rational states in $\hat{\mathcal{A}}$. Let P be the union of the rational classes of $\sim_{h-1}$. Then $P \supset Q$, and at least one class c that is split in $\sim_h$ is disjoint from P. The set P is stable for the next-state function: indeed, each state p in P is equivalent for $\sim_{h-1}$ to some state q in Q. The states p and q are mapped, by each letter a, into the same class for $\sim_{h-1}$ because $\sim_h$ does not split any class contained in P. Thus both states $q \cdot a$ and $p \cdot a$ are in P.

This shows that P is the set of states of a subautomaton, and since $\sim_{h-1}$ and $\sim_h$ are identical on this subautomaton, it is minimal and there will never be any splitting of one of its classes. So all splittings concern only classes that do not contain any element in Q. □

Note that if the tree t in Lemma 6 has infinite rank, then each $\sim_h$ also splits a rational class. Note also that if t is Sturmian and has infinite rank, there is no irrational class at all. Thus the lemma is of interest for trees with finite rank.

Corollary 3. *Let t be a Sturmian tree with finite rank. Let $\hat{\mathcal{A}}$ be the minimal automaton of the language of t. Assume that, for some integer h, the equivalence $\sim_h$ splits some irrational class. Then for all $h' \geq h$, $\sim_{h'}$ always splits an irrational class.* □

Lemma 7. *Let t be a Sturmian tree with finite rank. Let N be the number of rational states. The Moore equivalence $\sim_{N-1}$ has index N and each of these classes contains exactly one rational state.*

Proof. This is because the subautomaton composed of rational states is slow and minimal. □

Lemma 8. *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 exists, for each irrational state, an integer n such that the class of this state in $\sim_n$ is a singleton.*

Proof. Assume that the first of the two possibilities is not realized. Then for all n, there exists a class containing both rational and irrational states. We call here such a class a *mixed* class. Thus a mixed class is a rational class which contains at least one irrational state. We consider an $n \geq N$, where N is the number of rational states, so each mixed class contains exactly one rational state, and at least one irrational state. For each $n \geq N$ the class split in $\sim_n$ must be a mixed class, because otherwise in view of Lemma 6, this class will never be split. The same argument shows that every irrational class is a singleton. Let p be any irrational state in some mixed class c, containing a unique rational state, say r. Since $p \neq r$ and the automaton is minimal, there exists an integer m such that r and p are in different classes for $\sim_m$. The class of p is irrational because r is in another class for $\sim_m$. In view of Lemma 6, the class of p will never split, so it is a singleton class. □

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 property of Sturmian trees.

Proposition 6. *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 8, 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 8, every irrational state will be in singleton classes for large enough n. Since there are only finitely many fork states, each of these will be in a singleton irrational class of $\sim_n$ for some n.

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. According to Remark 1, 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. □

Assume now that the tree t has degree 1. Each of the subtrees of the infinite irrational branch contributes a state to $\hat{\mathcal{A}}$. Denote by $\mathbb{N}$ the set of states of the infinite branch. We number these states so that the next-state function of $\hat{\mathcal{A}}$ maps a state n in $\mathbb{N}$ to the state $n+1$ in $\mathbb{N}$ by one of the two letters, and to a rational state by the other.

The next statements show that after some additional steps, each of the rational states constitutes a singleton class of the corresponding Moore equivalence. Recall that an equivalence class is an *irrational* class if it is composed only of irrational states, and a *rational* class otherwise.

Lemma 9. *If the tree t is Sturmian and has degree one, then there is some integer n such that $\sim_n$ splits an irrational class.*

Proof. Assume the contrary: at each step, the class c_{n-1} split by $\sim_n$ contains a rational state. We assume $n \geq N$, so each class of these classes contains exactly one rational state.

In view of Corollary 1, there exists an infinite sequence of pairs (c'_n, c''_n) of classes of $\sim_n$ such that each $c_{n-1} = c'_n \cup c''_n$ is the equivalence class of $\sim_{n-1}$ which is split by $\sim_n$. There are also letters a_n, such that each a_n maps all states of c'_n into states of c'_{n-1}, and all states of c''_n into states of c''_{n-1}. One of the classes c'_n, c''_n contains the rational state. If this holds for c'_n, then it holds for c'_{n-1} since a rational state is mapped into a rational state. Next, the classes c''_n never split since they are irrational classes. Since the automaton is minimal, this implies that these classes are singleton classes. Set $c''_n = \{r_n\}$. Then there is an infinite path $\cdots r_n \xrightarrow{a_n} r_{n-1} \xrightarrow{a_{n-1}} \cdots \xrightarrow{a_{N+1}} r_N$, which is impossible since then $0 \leq r_n = r_{n-1} - 1 < r_{n-1}$ for all n, and the r_n are a strictly decreasing sequence of positive integers. □

From now on, the tree t is assumed to be a Sturmian tree of degree 1 and with finite rank. In particular, the previous lemma holds.

Corollary 4. *Let H be the smallest integer such that $\sim_H$ splits an irrational class. Then all rational equivalence classes of $\sim_{H-1}$ are singletons.*

Proof. Indeed consider a rational class c of $\sim_{H-1}$. By Lemma 3, the class c will never be split again. Since the automaton is minimal, it must be a singleton. □

Lemma 10. *Let H be the smallest integer such that $\sim_H$ splits an irrational class, and set $h = H - N$, where N is the number of rational states. Then there is a lazy path*

$$\pi : q_0 \xrightarrow{b_0} q_1 \xrightarrow{b_1} \cdots q_{h-1} \xrightarrow{b_{h-1}} q_h$$

with $q_{h-1} \cdot \bar{b}_h = q_0$ or q_h, where $q_0, \ldots, q_h$ are rational states, and $q_0 \sim_{N-2} q_h$.

Proof. According to the previous statements, the equivalence $\sim_{N-1}$ contains N classes, and each of its classes contains exactly one rational state. The equivalence $\sim_{H-1}$ contains H classes. N of theses classes are singleton classes composed of a rational state, the $H - N$ other classes are irrational classes. Each of the equivalence relations $\sim_n$, for $n = N, \ldots, H-1$, splits a rational class c_{n-1}

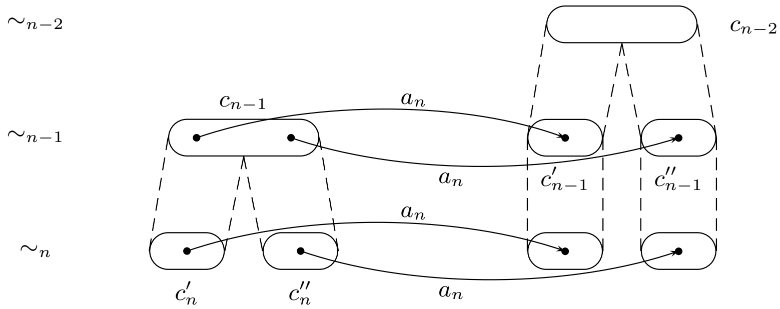

Fig. 19: The class c_{n-1} of $\sim_{n-1}$ is split into two classes c'_n and c''_n of $\sim_n$. The letter a_n maps states of c_{n-1} into states of the class c'_{n-1} and states of the class c''_{n-1} into states of the class c''_{n-1}. Observe that the classes c'_{n-1} and c''_{n-1} are those created at the preceding step.

containing a single rational state into a (smaller) class c'_n containing this state, and an irrational class c''_n. See Figure 19.

In view of Corollary 1, there are letters a_n such that states of c'_n are mapped into states of c'_{n-1} by a_n and states of c''_n are mapped by a_n into states of c''_{n-1}.

Each of the rational classes c'_n is mapped, by one letter, into the class c'_{n-1}, and by the other letter into some rational class, whereas each of the irrational classes c''_n is mapped, by one letter, into the irrational class c''_{n-1} and by the other letter into a rational class.

The $H-N$ irrational classes produced during these steps will not be changed before $\sim_H$: so all these irrational classes produced during these steps are also classes of $\sim_{H-1}$. Indeed, otherwise a relation splits an irrational class, and by Lemma 3 the classes containing rational states will not be split anymore. Denote by s_n the rational state in the class c'_n, for $n = N, \dots, H-1$. There is a path

$$s_{H-1} \xrightarrow{a_{H-1}} s_{H-2} \xrightarrow{a_{H-2}} \cdots \xrightarrow{a_{N+3}} s_{N+2} \xrightarrow{a_{N+2}} s_{N+1} \xrightarrow{a_{N+1}} s_N$$

Also, s_N is mapped by each letter to a rational state. see Figure 20.

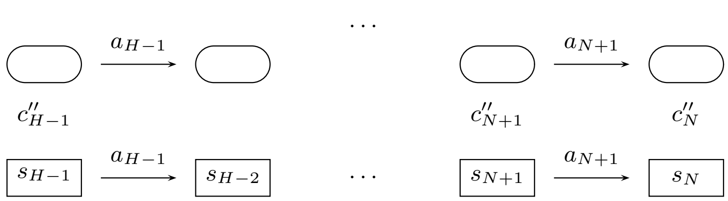

Fig. 20: The classes of $\sim_{H-1}$. The classes $c''_N, c''_{N+1}, \dots, c''_{H-1}$ are irrational, and the (rational) classes containing the states $s_N, s_{N+1}, \dots, s_{H-1}$ are singletons.

Consider now the equivalence $\sim_H$; it contains two fresh classes c'_H and c''_H, obtained by splitting a class c_{H-1} of $\sim_{H-1}$. Since all rational classes are singletons in $\sim_{H-1}$, the class c_{H-1} is an irrational class. This means that c_{H-1} is one of the $H-N$ irrational classes $c''_N, c''_N, \ldots, c''_{H-1}$. We prove that $c_{H-1} = c''_N$.

Since the class c_{H-1} is an irrational class, each of its states is mapped, by one letter, onto some rational state, and by the other letter onto an irrational state. Let a_H be the letter that maps all states of c'_H onto the state s_{H-1}, and all states of c''_H into the irrational class c''_{H-1}. It follows that the letter $\bar{a}_H$ maps all states of c'_H (those mapped by a_H onto the rational state) into some irrational class, and all states of c''_H (those mapped into irrational states by a_H) onto some rational state. See Figure 21.

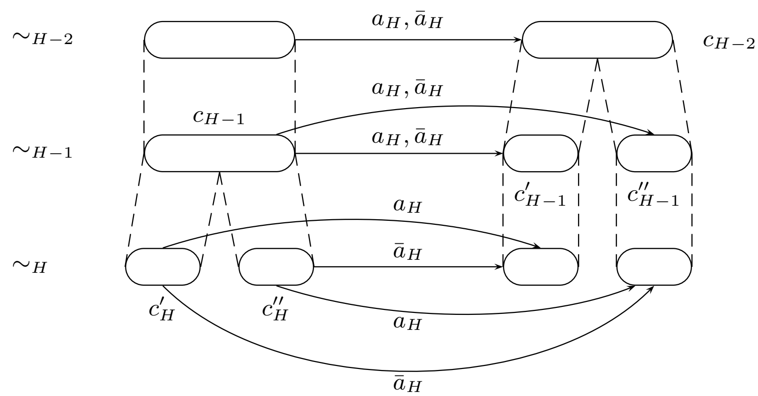

Fig. 21: The class c_{H-1} of $\sim_{H-1}$ is split into two classes c'_H and c''_H of $\sim_H$. The letter a_H maps states of c'_H into states of the class c'_{H-1} and states of the class c''_H into states of the class c''_{H-1} whereas the letter $\bar{a}_H$ has the opposite behavior: it maps c'_H into c''_{H-1} and c''_H into c'_{H-1}.

The class $c_{H-1} = c'_H \cup c''_H$ is a subset of an equivalence class of $\sim_{H-2}$. Consequently each of the letters maps all states of c_{H-1} onto states that are equivalent mod $\sim_{H-2}$, and so belong a class mod $\sim_{H-2}$ containing both a rational and irrational states. As we observed earlier, this class is $c_{H-2} = c'_{H-1} \cup c''_{H-1}$.

Thus the letter $\bar{a}_H$ maps states of c'_H into c''_{H-1} and states of c''_H onto s_{H-1}, that is, the letter $\bar{a}_H$ has the opposite behavior of letter a_H.

We have shown that each letter maps some of the states of the irrational class c_{H-1} to an irrational state, and at least one of its states to a rational one. Since each class c''_n, for $N < n \leq H-1$, is mapped, by the letter a_n, into an irrational class, the class c_{H-1} cannot be one of the classes c''_n for $N < n \leq H-1$, and so c_{H-1} is the only remaining irrational class, that is $c_{H-1} = c''_N$.

Recall that s_n denotes the rational state in class c'_n, for $n = N, \ldots, H-1$, and that there is a path

$$s_{H-1} \xrightarrow{a_{H-1}} s_{H-2} \xrightarrow{a_{H-2}} \cdots \xrightarrow{a_{N+3}} s_{N+2} \xrightarrow{a_{N+2}} s_{N+1} \xrightarrow{a_{N+1}} s_N$$

The state s_N is mapped by each letter to a rational state. Now the equivalence $\sim_{N-1}$ splits the class c_{N-2} of $\sim_{N-2}$ into two equivalence classes c'_{N-1} and c''_{N-1}, containing each one rational state, say $q' \in c'_{N-1}$ and $q'' \in c''_{N-1}$. Denote by a_N the letter that maps c'_N into c'_{N-1} and c''_N into c''_{N-1}. The states q' and q'' are the last rational states separated in Moore's algorithm.

The class c_{H-1} is mapped, by both letters, into the class c_{H-2} (see Figure 21). The only rational state in c_{H-2} is s_{H-1}. The class $c_{H-1} = c''_N$ is mapped by the letter a_N into c''_{N-1}. This implies that $c_{H-2} \subseteq c''_{N-1}$ and consequently $s_{H-1} = q''$. Since the class c'_N is mapped by a_N into c'_{N-1}, the state s_N is mapped by a_N to q', and by $\bar{a}_N$ to one of the two rational states in c_{N-2}, thus to q' or to q''.

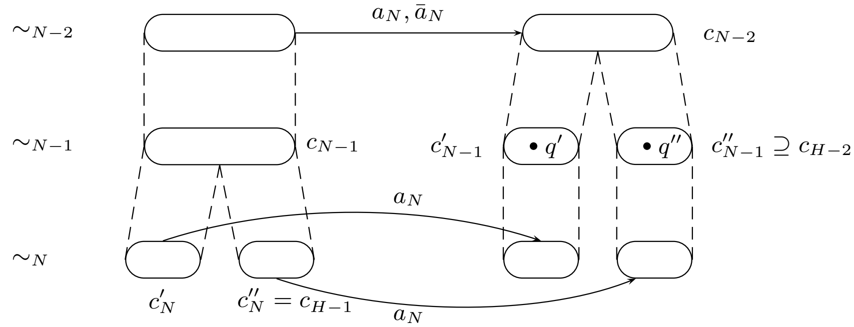

Fig. 22: The class c'_{N-1} of $\sim_{N-1}$ contains a rational state q', the class c''_{N-1} contains a rational state q''. The class c_{H-2} is a subset of c''_{N-1}, and $c''_N = c_{H-1}$.

In order to fit with the notations of the lazy path, it suffices now to set $h = H - N$, and $q_i = s_{H-1-i}$ and $b_i = a_{H-1-i}$, and $q_h = q''$. Then the path becomes

$$q_0 \xrightarrow{b_0} q_1 \xrightarrow{b_1} \cdots q_{h-2} \xrightarrow{b_{h-1}} q_{h-1}$$

Moreover, there is a transition $q_{h-1} \cdot b_h = q_h$, and for the other letter $\bar{b}_h$, either $q_{h-1} \cdot \bar{b}_h = q_0$ or $q_{h-1} \cdot \bar{b}_h = q_h$, and indeed q_0, q_h are the rational states that are equivalent in $\sim_{N-2}$. □

According to the (proof of the) preceding lemma, the irrational classes of $\sim_{H-1}$ and of $\sim_H$ behave as depicted in Figures 23 and 24.

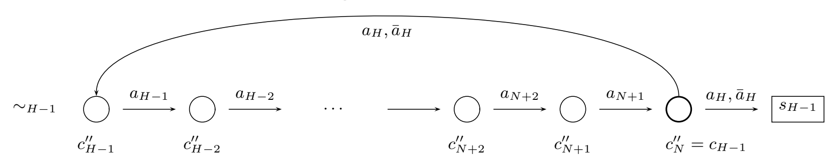

Fig. 23: The irrational classes of $\sim_{H-1}$.

It remains to exhibit a Sturmian word x such that $\hat{\mathcal{A}}$ is an extension of the claimed form. As already mentioned, each of the subtrees of the infinite irrational

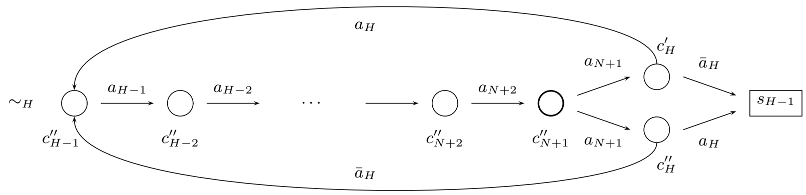

Fig. 24: The irrational classes of $\sim_H$ and s_{H-1}.

branch contributes a state to $\hat{\mathcal{A}}$. The states of the infinite branch are denoted by $\mathbb{N}$. They are numbered such that the next-state function of $\hat{\mathcal{A}}$ maps a state n in $\mathbb{N}$ to the state $n+1$ in $\mathbb{N}$ by one of the two letters, and to a rational state by the other.

The irrational path of the tree is labeled by an infinite word

$$y = y_0 y_1 y_2 \cdots \in \{0,1\}^\omega$$

and corresponds in $\hat{\mathcal{A}}$ to the infinite path

$$0 \xrightarrow{y_0} 1 \xrightarrow{y_1} 2 \to \cdots \to n \xrightarrow{y_{n+1}} n+1 \to \cdots$$

For each state n, the letter $\bar{y}_n$ maps n into a rational state. It follows from the previous properties of $\sim_{H-1}$ that

$$n \sim_{H-1} n' \iff n \equiv n' \bmod h\,.$$

We suppose for simplicity that $0 \in c''_{H-1}$, and we define the infinite word $x = x_0 x_1 \cdots$ by $x_0 = y_{h-1}$, $x_1 = y_{2h-1}$, and in general $x_i = y_{(i+1)h-1}$. We prove that x is a Sturmian word, by showing that it has $n+1$ factors of length n for each $n \geq 0$.

The key lemma is the following

Lemma 11. *Let $k, \ell \geq 0$ be integers with $k + \ell = h - 1$. Let $n \equiv n' \mod h$ and set $i = \lfloor n/h \rfloor$ and $i' = \lfloor n'/h \rfloor$. For all integer $j \geq 0$,*

$$n \sim_{H+jh+k} n' \iff \begin{cases} x_i \cdots x_{i+j-1} = x_{i'} \cdots x_{i'+j-1} & \textit{if } (n \bmod h) < \ell \\ x_i \cdots x_{i+j} = x_{i'} \cdots x_{i'+j} & \textit{otherwise.} \end{cases}$$

Observe that this is Lemma 5, but the assumptions are different. Indeed, in Lemma 5, the word x defines an automaton and a property of the Moore equivalences is looked for. Here we know the Moore equivalences and we want to describe a property of a word x that will allow us to show that it is Sturmian. However, the arguments of the proof are quite similar, so we sketch only one basic, but typical step.

The proof is by induction on $H + jh + k$. For $j = k = 0$, there is nothing to prove except when $n \equiv h - 1 \mod h$. In this case, the claim becomes that $n \equiv_H n'$ if and only if $x_i = x_{i'}$, with $n = ih + h - 1$.

By definition, one has $n \xrightarrow{x_i} n+1$ and $n' \xrightarrow{x_{i'}} n'+1$. Now, the equivalence relation $\equiv_H$ splits the irrational class c''_{H-1} of $\sim_{H-1}$ containing both n, n' into two irrational classes c'_H and c''_H of $\sim_H$ (see Figures 23 and 24). The states n, n' are in the same of these classes iff they are mapped by the same letter x_i into an irrational state. Thus $x_i = x'_i$.

It remains to count the number of factors of the word x. Set $k = h-1, \ell = 0$ in the previous lemma. There are $H+(j+1)h$ classes in the equivalence relation $\sim_{H+jh+h-1}$. Among them, $N = H - h$ are rational classes, so there are $(j+2)h$ irrational classes. Any two classes containing integers n, n' which are congruent modulo h define the same factor of length $j+1$ of x, so there are exactly $j+2$ distinct factors of length $j+1$ in the infinite word x. This concludes the proof of Theorem 1.

7 Trees with infinite rank

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

Example 14. 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 $\{(n,b) \in Q\}$ with even n. The set E of transitions is defined as follows. Let $n = 2^k m$ where $m \geq 1$ is odd. The integer 2^k is the greatest power of 2 which divides n.

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

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

For making the description more readable, we write $\textcircled{n}$ for $(n,0)$ and $\boxed{n}$ for $(n,1)$. In Figure 25 we give a picture of this automaton.

The next table gives the explicit values of the next-state function for values of n up to 16. The states are arranged in order to make the state splitting more easy to follow. Recall that even states are final.

	$\textcircled{4}$	$\textcircled{6}$	$\textcircled{10}$	$\boxed{10}$	$\textcircled{14}$	$\boxed{14}$	$\boxed{6}$	$\textcircled{8}$	$\boxed{4}$	$\textcircled{12}$	$\boxed{12}$	$\textcircled{16}$	$\boxed{8}$	$\boxed{16}$
0	$\textcircled{3}$	$\textcircled{7}$	$\textcircled{11}$	$\boxed{11}$	$\textcircled{15}$	$\boxed{15}$	$\boxed{7}$	$\textcircled{5}$	$\boxed{5}$	$\textcircled{13}$	$\boxed{13}$	$\textcircled{9}$	$\boxed{9}$	$\boxed{17}$
1	$\textcircled{4}$	$\textcircled{4}$	$\textcircled{4}$	$\textcircled{4}$	$\textcircled{4}$	$\textcircled{4}$	$\textcircled{4}$	$\textcircled{3}$	$\textcircled{3}$	$\textcircled{3}$	$\textcircled{3}$	$\textcircled{5}$	$\textcircled{5}$	$\textcircled{9}$

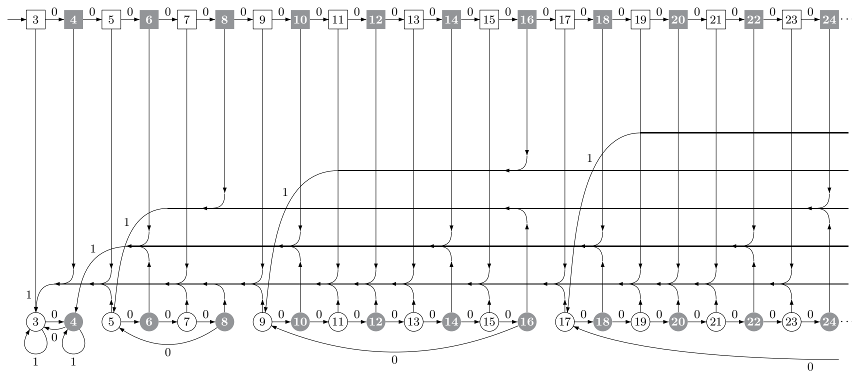

Fig. 25: Final states are dark. Observe the fractal-like structure, with a doubling of the size of each block.

	③	⑤	⑨	$\boxed{9}$	⑬	$\boxed{13}$	$\boxed{5}$	⑦	⑪	$\boxed{11}$	$\boxed{3}$	⑮	$\boxed{15}$	$\boxed{7}$
0	④	⑥	⑩	$\boxed{10}$	⑭	$\boxed{14}$	$\boxed{6}$	⑧	⑫	$\boxed{12}$	$\boxed{4}$	⑯	$\boxed{16}$	$\boxed{8}$
1	③	③	③	③	③	③	③	③	③	③	③	③	③	③

In order to show that the automaton is slow, we describe how the Moore algorithm behaves. This will show that the automaton is minimal. We start by considering the very first splittings. We use the shorthand $a + b\mathbb{N}$ for the arithmetic progression $\{a + bm \mid m \geq 0\}$.

The minimization algorithm starts with an equivalence relation composed of the two classes of even (final) and odd (nonfinal) states.

In the first step of partition refinement, only the letter 1 induces a splitting: it splits the even states into two classes, one composed of ④ and the states with numbers in $6+4\mathbb{N}$, the other containing $\boxed{4}$ and the states with numbers in $8+4\mathbb{N}$. Observe that after this splitting, the siblings $\boxed{4}$ and ④ are in distinct equivalence classes.

At the next step, again only one new class is created. The letter 0 splits the odd states into two classes, the first composed of ③ and of the states with numbers in $5+4\mathbb{N}$, the second composed of $\boxed{3}$ and the states in $7+4\mathbb{N}$. Observe that at this stage, there are four equivalence classes, and the siblings $\boxed{4}$ and ④, and the siblings $\boxed{3}$ and ③ are in distinct equivalence classes.

In the next step, the letter 0 isolates state ④, that is ④ is a singleton class. Indeed, it is mapped on ③ whereas all states with numbers in $6 + 4\mathbb{N}$ are mapped on states in $7 + 4\mathbb{N}$.

As a consequence, in the next step ③ is itself isolated, since it is mapped by 0 on ④ whereas the states in $5 + 4\mathbb{N}$ are mapped on states in $6 + 4\mathbb{N}$. At this stage, there are six equivalence classes, namely the states ④ and ③ which are singletons and the classes composed of states with numbers in $5 + 4\mathbb{N}$ and

$6+4\mathbb{N}$, the class containing $\boxed{4}$ and states with numbers in $8+4\mathbb{N}$, and finally the class containing $\boxed{3}$ and states with numbers in $7+4\mathbb{N}$.

Consider one more step. The letter 1 splits the class containing $\boxed{4}$ and states with numbers in $8+4\mathbb{N}$ since $\textcircled{8}$ and states with numbers in $12+8\mathbb{N}$ are mapped on $\textcircled{3}$ whereas $\boxed{8}$ and states with numbers in $16+8\mathbb{N}$ are mapped on $\textcircled{5}$. Thus in particular, this step puts $\boxed{8}$ and $\textcircled{8}$ in distinct equivalence classes.

We are now able to give an overview of the minimization process. It operates on the automaton by levels, each level taking care of the states up to the next power of two (both rational and irrational). Thus the first level operates on the four states numbered 3 and 4, the second level operates on the eight states numbered from 5 to 8, and so on.

Each level is composed of a separation phase, followed by an isolation phase. Each step in the separation phase puts the states $\boxed{n}$ and $\textcircled{n}$ in distinct equivalence classes. This separation phase is by decreasing numbers, starting with a power of two. The isolation phase creates singleton classes. Each step creates a singleton class $\textcircled{n}$, again by decreasing numbers.

Thus the four steps of the two phases in level 1 are summarized by

1	$\textcircled{4}$	$\boxed{4}$	separated
01	$\textcircled{3}$	$\boxed{3}$	separated
001	$\textcircled{4}$		isolated
0001	$\textcircled{3}$		isolated

Thus the eight steps of the two phases in level 1 are summarized by

$10^{3}1$	$\textcircled{8}$	$\boxed{8}$	separated
$010^{3}1$	$\textcircled{7}$	$\boxed{7}$	separated
$0010^{3}1$	$\textcircled{6}$	$\boxed{6}$	separated
$00010^{3}1$	$\textcircled{5}$	$\boxed{7}$	separated
$0^{4}10^{3}1$	$\textcircled{8}$		isolated
$0^{5}10^{3}1$	$\textcircled{7}$		isolated
$0^{6}10^{3}1$	$\textcircled{6}$		isolated
$0^{7}10^{3}1$	$\textcircled{5}$		isolated

In the first column we have given a word that separates or isolates the corresponding states.

This description shows why the automaton is minimal: states of the form $\textcircled{n}$ eventually become singleton classes, states of the form $\boxed{n}$ never become singleton classes, but two such states will eventually be separated.

Remark 2. The automaton of the previous example may be modified in two ways, giving still a slow automaton.

(a) The first modification consists in defining mapping each state (n, b) with n odd by both letters onto the state $(n+1, b)$. One observes that the Moore minimization algorithm behaves in the same way. Now, both the rank and the degree of the tree are infinite.

(b) The other modification is to map all states that where mapped on $\boxed{3}$ in the initial automaton on state $\textcircled{3}$. Again, the minimization process does not change. There are no rational states anymore, so the rank is 0 and the degree is infinite.

This second modification gives some additional information about the splitting of irrational states in a slow automaton. We have seen examples, as the Dyck automaton, where splitting is by isolating irrational states. There were also examples such as the automaton of the indicator trees, where irrational classes are never singletons. Here we get an example where some of the states are isolates (all states (n)) and others are never (the states [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 6.

Example 15. 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 $n = 2^k 3^\ell m$ 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.

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

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

In Figure 26 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 [n] and states of the form $(n,2)$ as lozenges ◇n◇ .

The Moore algorithm here is again slow. It is more complicated. The circled and square states are used in a very similar way to that of Example 14, but with powers of 3. The powers of 3 work for circles and lozenges. It remains to check that the two processes never collide (which would produce more than a single split in some step). We leave the formal verification to the reader.

Example 16. We define a tree by giving a (minimal) automaton accepting its language. The set of states of the automaton is $Q = \{(n,0) \in \mathbb{N} \mid n \geq 3\} \cup$

$\{(n, 1) \in \mathbb{N} \mid n \geq 0\} \cup \{(2^k, 2) \in \mathbb{N} \mid k \geq 1\}$. The set of final states is the set $\{(n, b) \in Q\}$ with even n. The set E of transitions is defined as follows. Let $n = 2^k m$ where $m \geq 1$ is odd. The integer 2^k is the greatest power of 2 which divides n. First $(n, 2) \cdot 0 = (n, 2) \cdot 1 = (n + 1, 0)$ for all n. Next $(0, 1) \cdot 0 = (0, 1) \cdot 1 = (1, 1)$. For $n \geq 1$ and $b = 0, 1$,

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

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

In Figure 27 we give a picture of this automaton; again states of the form $(n, 0)$ are drawn as circles ⓝ and states of the form $(n, 1)$ as squares $\boxed{n}$ and states of the form $(n, 2)$ as lozenges ◇n◇ .

The initial state $(0, 1)$ is the only irrational state for which both outgoing edges lead to nonfinal states. Since it has no incoming edge, this ensures that it cannot appear in an irrational path in an other place than the root. Thus the example shows that Proposition 4 does not hold in this case. However this tree is Sturmian. In particular, observe that rational states $(2^n, 2)$ corrrespond to rational nodes in the tree which share the same prefix tree of increasing finite height than the root $(0, 1)$.

8 Concluding Remarks

We have introduced the notion of Sturmian trees and we have considered two parameters, the degree and the rank. We have described the structure of Sturmian trees of finite rank and finite degree. This is the main contribution of this paper.

We have given several examples of Sturmian trees of finite rank and infinite degree. We have also given examples of Sturmian trees of infinite rank. These have some kind of fractal structure. We have built Sturmian trees of infinite rank and of degree two, and with both infinite degree and infinite rank. These examples show that the structure of general Sturmian trees is quite involved, and there seems not to be such a concise characterization as it exists in the case of Sturmian words.

Acknowledgements

We thank the referees for their careful reading of the manuscript and for their comments which greatly improved the exposition.

References

1. P. Arnoux and G. Rauzy. Représentation géométrique de suites de complexité $2n+1$. *Bulletin Société Mathématique de France*, 119:199–215, 1991.
2. A. Carpi, A. de Luca, and S. Varricchio. Special factors and uniqueness conditions in rational trees. *Theory of Computing Systems*, 34:375–395, 2001.
3. B. Courcelle. Fundamental properties of infinite trees. *Theoretical Computer Science*, 25:95–165, 1983.
4. E. M. Coven and G. A. Hedlund. Sequences with minimal block growth. *Mathematical Systems Theory*, 7:138–153, 1973.
5. M. Lothaire. *Algebraic Combinatorics on Words*, volume 90 of *Encyclopedia of Mathematics*. Cambridge University Press, 2002.
6. N. Pytheas-Fogg. *Substitutions in Dynamics, Arithmetics and Combinatorics*, volume 1794 of *Lecture Notes in Mathematics*. Springer-Verlag, 2002. Edited by V. Berthé, S. Ferenczi, C. Mauduit and A. Siegel.
7. J. Sakarovitch. *Élements de théorie des automates*. Vuibert, 2003.

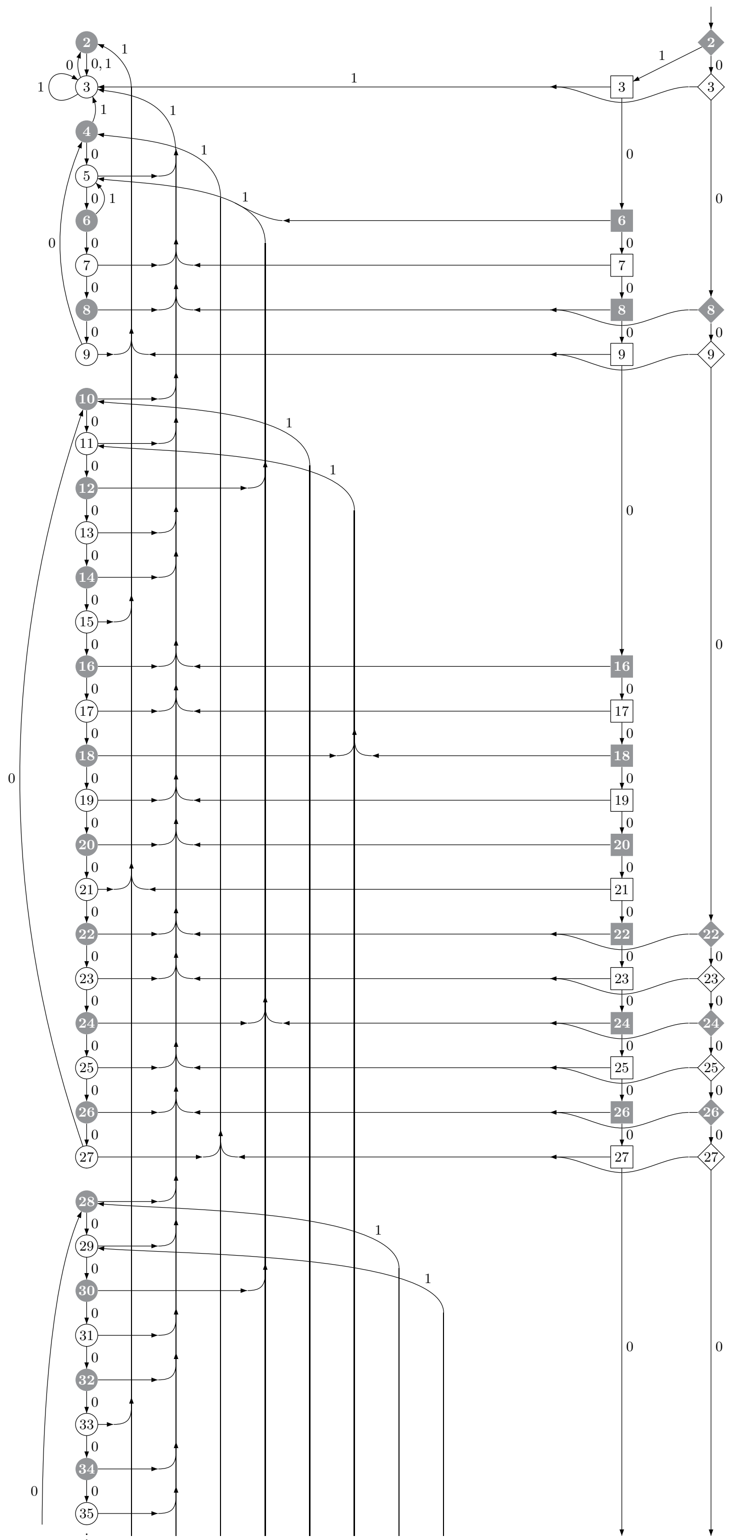

Fig. 26: Sturmian tree of degree 2

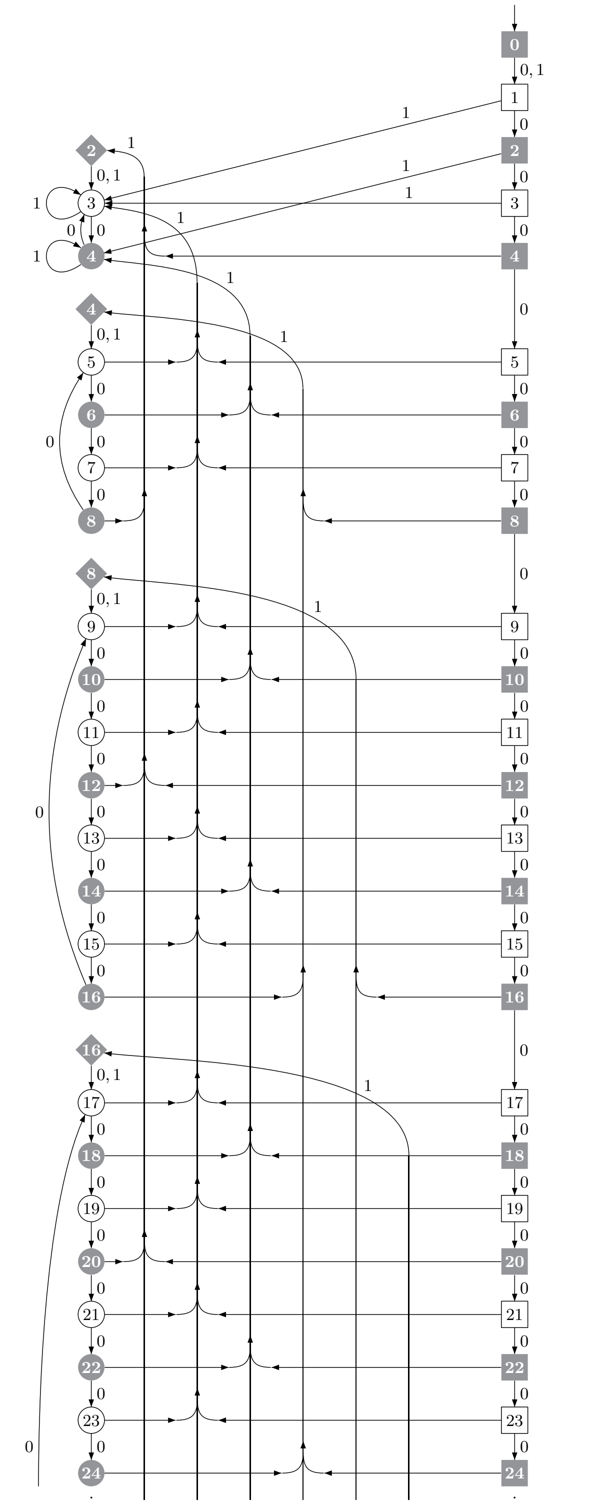

Fig. 27: A tree with infinite rank. Final states are dark.