Sofic-Dyck shifts

Marie-Pierre Béal, Michel Blockelet, Catalin Dima

Sofic-Dyck shifts☆

Marie-Pierre Béal

Université Paris-Est, Laboratoire d'informatique Gaspard-Monge, UMR 8049 CNRS

Michel Blockelet

Université Paris-Est, Laboratoire d'Algorithmique, Complexité et Logique

Cătălin Dima

Université Paris-Est, Laboratoire d'Algorithmique, Complexité et Logique

Abstract

We define the class of sofic-Dyck shifts which extends the class of Markov-Dyck shifts introduced by Inoue, Krieger and Matsumoto. Sofic-Dyck shifts are shifts of sequences whose finite factors form unambiguous context-free languages. We show that they correspond exactly to the class of shifts of sequences whose sets of factors are visibly pushdown languages. We give an expression of the zeta function of a sofic-Dyck shift.

Keywords: Dyck shift, Markov-Dyck shift, sofic-Dyck shift, sofic shift, symbolic dynamics, visibly pushdown automaton, visibly pushdown language, zeta function

1. Introduction

Shifts of sequences are defined as sets of bi-infinite sequences of symbols over a finite alphabet avoiding a given set of finite factors called forbidden factors. Well-known classes of shifts of sequences are the shifts of finite type which avoid a finite set of forbidden factors and the sofic shifts which avoid a regular set of forbidden factors. Sofic shifts may also be defined as labels of bi-infinite paths of a labeled directed graph.

Dyck shifts are shifts of sequences whose finite factors are factors of well-parenthesized words. They were introduced by Krieger in [28]. In [21], [31], [22], Inoue, Krieger, and Matsumoto investigated generalizations of Dyck shifts called

☆This work is supported by the French National Agency (ANR) through "Programme d'Investissements d'Avenir" (Project ACRONYME n°ANR-10-LABX-58) and through the ANR EQINOCS.

Email addresses: `beal@univ-mlv.fr` (Marie-Pierre Béal), `michel.blockelet@u-pec.fr` (Michel Blockelet), `catalin.dima@u-pec.fr` (Cătălin Dima)

Markov-Dyck shifts. Their languages of factors are unambiguous context-free languages. Such shifts are presented by a finite-state directed graph equipped with a graph inverse semigroup. The graph can be considered as an automaton which operates on words over an alphabet which is partitioned into two disjoint sets, one for the left parentheses, the other one for the right parentheses. In [22], Inoue and Krieger introduced an extension of Markov-Dyck shifts by constructing shifts from sofic systems and Dyck shifts. Examples of shifts of this type are the Motzkin shifts. Dyck shifts and their extensions are in general not synchronized but Krieger and Matsumoto introduced weaker notions of synchronization suitable for Markov-Dyck or Motzkin shifts (see [29], [36], [32]). Flow invariants for these shifts are obtained in [36] and [15]. In [30] (see also [20] and [19]), Krieger considers subshift presentations, called $\mathcal{R}$-graphs, with word-labeled edges partitioned into two disjoint sets of positive and negative edges equipped with a relation $\mathcal{R}$ between positive and negative edges going backwards.

In this paper, we introduce a larger class of shifts. We consider shifts of sequences presented by a finite-state automaton (a labeled graph) equipped with a set of pairs of edges called matched edges. The matched edges may not be consecutive edges of the graph. We call such structures Dyck automata. They may be equipped with a graph semigroup which is no more an inverse semigroup. The automaton operates on words over an alphabet which is partitioned into three disjoint sets of symbols, the call symbols, the return symbols, and the internal symbols (for which no matching constraints are required).

We call the shifts presented by Dyck automata sofic-Dyck shifts. We prove that this class is exactly the class of shifts of sequences whose set of factors is a visibly pushdown language of finite words. Equivalently, they can be defined as the sets of sequences which avoid some visibly pushdown language of factors. So these shifts could also be called visibly pushdown shifts.

Visibly pushdown languages were introduced by Mehlhorn [38] and Alur *et al.* [1, 2]. They form a natural and meaningful class inside the class of unambiguous context-free languages extending the parenthesis languages [37], [27], the bracketed languages [18], and the balanced languages [9], [10]. These languages share many interesting properties with regular languages like stability by intersection and complementation. Visibly pushdown languages are used as models for structured data files like XML files.

We define also a subclass of sofic-Dyck shifts called finite-type-Dyck shifts. We prove that sofic-Dyck shifts are images of finite-type-Dyck shifts under proper block maps, *i.e.* block maps mapping call (resp. return, internal) symbols to call (resp. return, internal) symbols. The classes of sofic-Dyck shifts and finite-type-Dyck shifts are invariant by proper conjugacies.

In a second part of the paper, we address the problem of the computation of the zeta function of sofic-Dyck shift presented by a Dyck automaton. The zeta function allows to count the number of periodic points of a subshift. It is a conjugacy invariant of a class of shifts. Two subshifts which are conjugate (or isomorphic) have the same zeta functions. The invariant is not complete and it is not known, even for shifts of finite type, whether the conjugacy is a decidable property [33].

The formula of the zeta function of a shift of finite type is due to Bowen and Lanford [14]. Formulas for the zeta function of a sofic shift were obtained by Manning [34] and Bowen [13]. Proofs of Bowen's formula can be found in [33] and [6, 5]. An $\mathbb{N}$-rational expression of the zeta function of a sofic shift has been obtained by Reutenauer in [39] (see also [12]). Formulas for zeta functions of flip systems of finite type are given in [24], and for sofic flip systems in [25]. The zeta functions of the Dyck shifts were determined by Keller in [23]. For the Motzkin shift where some unconstrained symbols are added to the alphabet of a Dyck shift, the zeta function was determined by Inoue in [21]. In [31], Krieger and Matsumoto obtained an expression for the zeta function of a Markov-Dyck shift by applying a formula of Keller and with a clever encoding of periodic points of the shift.

In Section 6, we give an expression of the zeta function of a sofic-Dyck shift. The proof combines techniques used for computing the zeta function of a (non Dyck) sofic shift and of a Markov-Dyck shift. We implicitly use the fact that the intersection of two visibly pushdown languages is a visibly pushdown language. We give an example of the computation of the zeta function of a sofic-Dyck shift.

A short version of this paper appeared in [7].

2. Shifts

We introduce below some basic notions of symbolic dynamics. We refer to [33, 26] for an introduction to this theory. Let A be a finite alphabet. The set of finite sequences or words over A is denoted by A^* and the set of nonempty finite sequences or words over A is denoted by A^+. The *shift transformation* σ on $A^{\mathbb{Z}}$ is defined by

$$\sigma((x_i)_{i\in\mathbb{Z}}) = (x_{i+1})_{i\in\mathbb{Z}},$$

for $(x_i)_{i\in\mathbb{Z}} \in A^{\mathbb{Z}}$. A *factor* of a bi-infinite sequence x is a finite word $x_i \cdots x_j$ for some i, j, the factor being the empty word if $j < i$.

A *subshift* (or *shift*) of $A^{\mathbb{Z}}$ is a closed shift-invariant subset of $A^{\mathbb{Z}}$ equipped with the product of the discrete topology. If X is a shift, a finite word is *allowed* for X (or is a *block of* X) if it appears as a factor of some bi-infinite sequence of X. We denote by $\mathcal{B}(X)$ the set of blocks of X and by $\mathcal{B}_n(X)$ the set blocks of length n of X. Let F be a set of finite words over the alphabet A. We denote by X_F the set of bi-infinite sequences of $A^{\mathbb{Z}}$ avoiding all words of F, *i.e.* where no factor belongs to F. The set X_F is a shift and any shift is the set of bi-infinite sequences avoiding all words of some set of finite words. When F can be chosen finite (resp. regular), the shift X_F is called a *shift of finite type* (resp. *sofic*).

Let L be a language of finite words over a finite alphabet A. The language is *extensible* if for any $u \in L$, there are letters $w, z \in A^+$ such that $wuz \in L$. It is *factorial* if any factor of a word of the language belongs to the language.

If X is a subshift, $\mathcal{B}(X)$ is a factorial extensible language. Conversely, if L is a factorial extensible language, then the set $\mathcal{B}^{-1}(L)$ of bi-infinite sequences x such that any finite factor of x belongs to L is a subshift [33].

Let $X \subseteq A^{\mathbb{Z}}$ be a shift and m, n be nonnegative integers. A map $\Phi : X \to B^{\mathbb{Z}}$ is called an (m,n)-*block map* with memory m and anticipation n if there exists a function $\phi : \mathcal{B}_{m+n+1}(X) \to B$ such that, for all $x \in X$ and any $i \in \mathbb{Z}$, $\Phi(x)_i = \phi(x_{i-m} \cdots x_{i-1} x_i x_{i+1} \cdots x_{i+n})$. A *block map* is a map which is an (m,n)-block map for some nonnegative integers m, n.

A *conjugacy* is a bijective block map from X to Y. A property of subshifts which is invariant by conjugacies is called a *conjugacy invariant*.

3. Sofic-Dyck shifts

In this section, we define the class of sofic-Dyck shifts which generalizes the class of Markov-Dyck shifts introduced in [28] and [35] (see also [31]).

We consider an alphabet A which is a disjoint union of three finite sets of letters, the set A_c of *call letters*, the set A_r of *return letters*, and the set A_i of *internal letters*. The set $A = A_c \sqcup A_r \sqcup A_i$ is called a *pushdown alphabet*.

The two sets of call and return symbols may not have the same size. We assume that any call symbol may match any return symbol. We denote by $\mathrm{MR}(A)$ the set of all finite words over A where every return symbol is matched with a call symbol, *i.e.* $u \in \mathrm{MR}(A)$ if for every prefix u' of u, the number of call symbols of u' is greater than or equal to the number of return symbols of u'. These words are called *matched-return*. Similarly, $\mathrm{MC}(A)$ denotes the set of all words where every call symbol is matched with a return symbol, *i.e.* $u \in \mathrm{MC}(A)$ if for every suffix u' of u, the number of return symbols of u' is greater than or equal to the number of call symbols of u'. These words are called *matched-call*. We say that a word is a *Dyck word* if it belongs to the intersection of $\mathrm{MC}(A)$ and $\mathrm{MR}(A)$. Dyck words are well-parenthesized or well-formed words. Note that the empty word or all words over A_i are Dyck words. The set of Dyck words over A is denoted by $\mathrm{Dyck}(A)$. For instance for $A_c = \{(, [\}$, $A_r = \{),]\}$, $A_i = \{i\}$, the word $(\,(\,[\,i\,)$ is matched-return, the word $(\,]\,i\,]$ is matched-call and $(\,[\,i\,]\,]\,(\,)$ is a Dyck word on A.

A *(finite) Dyck automaton* $\mathcal{A}$ over A is a pair $(\mathcal{G}, M)$ of an automaton (or a directed labeled graph) $\mathcal{G} = (Q, E, A)$ over A where Q is the finite set of states, $E \subseteq Q \times A \times Q$ is the set of edges, and with a set M of pairs of edges $((p,a,q),(r,b,s))$ such that $a \in A_c$ and $b \in A_r$. The edges labeled by call letters (resp. return, internal) letters are also called *call* (resp. *return*, *internal*) *edges* and are denoted by E_c (resp. E_r, E_i). The set M is called the set of *matched edges*. If e is an edge we denote by $s(e)$ its starting state and by $t(e)$ its target state.

A finite path π of $\mathcal{A}$ is said to be an *admissible path* if for any factor $(p,a,q) \cdot \pi_1 \cdot (r,b,s)$ of π with $a \in A_c$, $b \in A_r$ and the label of π_1 being a Dyck word on A, $((p,a,q),(r,b,s))$ is a matched pair. Hence any path of length zero is admissible and factors of finite admissible paths are admissible. A bi-infinite path is *admissible* if all its finite factors are admissible.

The *sofic-Dyck shift presented* by $\mathcal{A}$ is the set of labels of bi-infinite admissible paths of $\mathcal{A}$ and $\mathcal{A}$ is called a *presentation* of the shift.

An equivalent semantics of Dyck automata is given in [7] with a graph semigroup associated to $\mathcal{A}$. This graph semigroup is no more an inverse semigroup as for presentations associated to Markov-Dyck shifts [28].

Note that the label of a finite admissible path may not be a block of the presented shift since a finite admissible path may not be extensible to a bi-infinite admissible path.

Lemma 1. *The sofic-Dyck shift presented by a Dyck automaton is exactly the set of bi-infinite sequences x such that each finite factor of x is the label of a finite admissible path.*

Proof. Let X be the sofic-Dyck shift presented by a Dyck automaton $\mathcal{A}$. By definition, any finite factor of a bi-infinite sequence of X is the label of a finite admissible path.

The converse part is due to the following classical compactness argument. Let x be a bi-infinite sequence such that each finite factor of x is the label of a finite admissible path. Thus for any positive integer i, there is a path

$$p_{i,-i-1} \xrightarrow{x_{-i}} p_{i,-i} \xrightarrow{x_{i-1}} \cdots p_{i,-1} \xrightarrow{x_0} p_{i,0} \xrightarrow{x_1} p_{i,1} \cdots \xrightarrow{x_i} p_{i,i},$$

which is admissible for $\mathcal{A}$. For each nonnegative integer m, there is an infinite number of such paths sharing the states p_k at all indices k for $-m \le k \le m$. Then $\pi = ((p_{k-1}, x_k, p_k))_{k \in \mathbb{Z}}$ is a bi-infinite path whose finite factors are admissible paths of $\mathcal{A}$. Thus the label x of π belongs to X. □

Proposition 1. *A sofic-Dyck shift is a subshift.*

Proof. Let X be a sofic-Dyck shift defined by an automaton $\mathcal{A}$. Let F be the set of finite words which are not the label of any finite admissible path of $\mathcal{A}$. Then $X = \mathsf{X}_F$ by Lemma 1 and thus X is a subshift. □

We denote respectively by $\mathrm{MR}(X)$, $\mathrm{MC}(X)$ and $\mathrm{Dyck}(X)$, the intersections of $\mathrm{MR}(A)$, $\mathrm{MC}(A)$ and $\mathrm{Dyck}(A)$ with the set of blocks of X.

Example 1. Let $A = A_c \sqcup A_r \sqcup A_i$ with $A_c = \{a_1, .., a_k\}$, $A_r = \{b_1, .., b_k\}$ and A_i is the empty set. The *Dyck shift* of order k over the alphabet A is the set of all sequences accepted by the one-state Dyck automaton $\mathcal{A} = (\mathcal{G}, M)$ containing all loops (p, a, p) for $a \in A$, and where the edge (p, a_i, p) is matched with the edge (p, b_i, p) for $1 \le i \le k$.

A Motzkin shift is the set of bi-infinite sequences presented by the automaton $A = A_c \sqcup A_r \sqcup A_i$ with $A_c = \{a_1, .., a_k\}$, $A_r = \{b_1, .., b_k\}$, the set A_i being no more the empty set. A Motzkin shift is represented in the left part of Figure 1. It is shown in [21] that the entropy of the Motzkin shift on this alphabet is $\log 4$. Another example is the sofic-Dyck shift X is presented by the Dyck automaton in the right part of Figure 1. For instance, the bi-infinite sequences $\cdots(([ii][])\cdots$ and $\cdots)))))\cdots$ belong to X while the sequences $\cdots([i][])\cdots$ or $\cdots(]\cdots$ do not. We have $(ii)() \in \mathrm{Dyck}(X)$ and $()([i \in \mathrm{MR}(X)$.

Note that a call symbol may match several return symbols and conversely although it is not the case in the above examples.

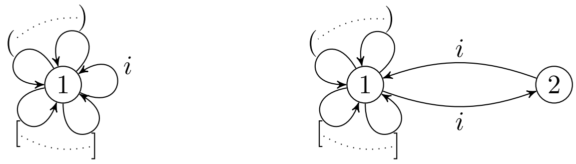

Figure 1: A Motzkin shift (on the left) over $A = A_c \sqcup A_r \sqcup A_i$ with $A_c = \{(, [\}$, $A_r = \{),]\}$ and $A_i = \{i\}$. A sofic-Dyck shift (on the right) over the same tri-partitioned alphabet. Matched edges are linked with a dotted line.

4. Finite-type-Dyck shifts

In this section we give a definition of a subclass of sofic-Dyck shifts called finite-type-Dyck shifts. We show that sofic-Dyck shifts are the images of finite-type-Dyck shifts by proper block maps.

Let A and B be two tri-partitioned alphabets. We say that a block-map $\Phi : A^{\mathbb{Z}} \to B^{\mathbb{Z}}$ is *proper* if and only if $\Phi(x)_i \in A_c$ (resp. A_r, A_i) whenever $x_i \in A_c$ (resp. A_r, A_i).

Let A be a tri-partitioned alphabet. If (u, v) and (u', v') are two pairs of words over A, we note $(u, v) \leq (u', v')$ if u is a suffix of u' and v is a prefix of v'.

Let $F \subseteq A^*$ and $U \subseteq (A^* \times A_c \times A^*) \times (A^* \times A_r \times A^*)$. We say that a finite or bi-infinite sequence x *avoids* F if, for each finite factor u of x, one has $u \notin F$. We say that a finite or bi-infinite sequence x *avoids* U if for each finite factor $u = vawbz$ of x with $a \in A_c, b \in A_r$, $w \in \text{Dyck}(A)$, there is no pair $((u_1, a, u_2), (v_1, b, v_2))$ in U such that $(u_1, u_2) \leq (v, wbz)$ and $(v_1, v_2) \leq (vaw, z)$.

A *finite-type-Dyck shift* over A is a set of bi-infinite sequences X for which there are two *finite* sets $F \subseteq A^*$, $U \subseteq (A^* \times A_c \times A^*) \times (A^* \times A_r \times A^*)$, such that X is the set of sequences *avoiding* F and U.

Proposition 2. *A finite-type-Dyck shift is a sofic-Dyck shift.*

Proof. Let X be a finite-type-Dyck shift of bi-infinite sequences over A avoiding two finite sets F and U. Without loss of generality we may assume that there are positive integers m, n such that $F \subseteq A^{m+n+1}$ and $U \subseteq (A^m \times A_c \times A^n) \times (A^m \times A_r \times A^n)$.

We define the Dyck automaton $\mathcal{A} = (\mathcal{G}, M)$ over A as follows. Let us denote $\mathcal{G} = (Q, E)$. We set

- $Q = \{(u, v) \mid u \in A^m, v \in A^n\}$,
- $E = \{((bu, av), a, (ua, vc)) \mid a, b, c \in A, u \in A^{m-1}, v \in A^{n-1}, buavc \notin F\}$,
- M is the set of pairs of edges $((du, av), a, (ua, vc))$, $((d'u', bv'), b, (u'b, v'c'))$, where $a \in A_c$, $b \in A_r$, $c, c', d, d' \in A$, $u, u' \in A^{m-1}, v, v' \in A^{n-1}$ and such that $((du, a, vc), (d'u', b, v'c')) \notin U$.

The sofic-Dyck shift presented by $\mathcal{A}$ is X. □

Proposition 3. *Sofic-Dyck shifts are the images of finite-type-Dyck shifts by proper block maps.*

Proof. We first show that any sofic-Dyck shift is the image of a finite-type-Dyck shift by a proper block map.

Let $\mathcal{A} = (\mathcal{G}, M)$ be a Dyck automaton accepting a sofic-Dyck shift X over A with $\mathcal{G} = (Q, E)$. Let $E = E_c \sqcup E_r \sqcup E_i$ be the tri-partitioned alphabet of edges of $\mathcal{A}$ where E_c (resp. E_r, E_i) is the set of call (resp. return, internal) edges of $\mathcal{A}$.

We define a Dyck automaton $\mathcal{B}$ over E as follows. The set of states of $\mathcal{B}$ is the set of states Q of $\mathcal{A}$. There is an edge $(p, e, q) \in \mathcal{B}$ if and only if e is an edge of $\mathcal{A}$ starting at p and ending in q. A pair of edges $((p, e, q), (r, f, s))$ of $\mathcal{B}$ is matched if (e, f) is a matched pair of $\mathcal{A}$.

Let Y be the sofic-Dyck shift presented by $\mathcal{B}$. It is the set of sequences avoiding

- $F = \{ef \in E^2 \mid t(e) \neq s(f)\}$,
- $U = \{((p, e, q), (r, f, s)) \in E_c \times E_r \mid (e, f) \notin M\}$,

Since F and U are finite, the shift Y is a finite-type-Dyck shift.

Let $\Phi : E^{\mathbb{Z}} \to A^{Z}$ be the $(0, 0)$-block map defined by $\phi : \mathcal{B}_1(Y) \to A$ as follows. We set $\phi(e) = a$ where a is the label of the edge e of $\mathcal{A}$. The map Φ is clearly a proper block map sending each bi-infinite admissible path of $\mathcal{A}$ to its label. As a consequence $X = \Phi(Y)$.

We now prove that the image of a finite-type-Dyck shift by a proper block map is a sofic-Dyck shift.

Let $\Phi : A^{\mathbb{Z}} \to B^{\mathbb{Z}}$ be a proper block map and X be a finite-type-Dyck shift of sequences over A. Without loss of generality we may assume that there are positive integers m, n such that Φ is a proper (m, n)-block map and X is the set of sequences avoiding two finite sets F and U with $F \subseteq A^{m+n+1}$ and $U \subseteq (A^m \times A_c \times A^n) \times (A^m \times A_r \times A^n)$.

Let $\phi : A^{m+n+1} \to B$ be the function defining Φ. We define the Dyck automaton $\mathcal{A}(\phi, F, U) = (\mathcal{G}, M)$ over $A \times B$ as follows. Let us denote $\mathcal{G} = (Q, E)$. We set

- $Q = \{(u, v) \mid u \in A^m, v \in A^n\}$,
- $E = \{((bu, av), (a, \phi(buavc)), (ua, vc)) \mid a, b, c \in A, u \in A^{m-1}, v \in A^{n-1}$ and $buavc \notin F\}$,
- M is the set of pairs of edges (e, f) with $e = ((bu, av), (a, \phi(buavc)), (ua, vc))$, $f = ((b'u', a'v'), (a', \phi(b'u'a'v'c')), (u'a', v'c'))$, where $a \in A_c$, $a' \in A_r$, $b, b', c, c' \in A$, $u, u' \in A^{m-1}, v, v' \in A^{n-1}$ and the pair (bu, a, vc), $(b'u', a', v'c') \notin U$.

Let $\mathcal{A}_1$ (resp. $\mathcal{A}_2$) be the Dyck automaton obtained by removing the second (resp. first) components of the labels of the edges. The Dyck automaton $\mathcal{A}_1$ is a presentation of X. Further, if $x \in X$, there is a unique admissible path of

$\mathcal{A}_1$ labeled by x. Indeed, each factor of a bi-infinite path of $\mathcal{A}_1$ labeled by uv with $u \in A^m$, $v \in A^n$, goes through the state (u,v) after reading u. A pair of bi-infinite sequences (x,y) is the label of a bi-infinite admissible path of $\mathcal{A}(\phi, F, U)$ if and only if $x \in X$ and $\Phi(x) = y$. Hence $\mathcal{A}_2$ is a presentation of $\Phi(X)$ which is thus sofic-Dyck. □

Proposition 4. *The image of a sofic-Dyck shift by a proper block map is a sofic-Dyck shift.*

Proof. Let Φ a proper block map from a sofic-Dyck shift X onto Y. By Proposition 3, X is the image of a finite-type-Dyck shift S by a proper block map Ψ. The map $\Phi \circ \Psi : S \to Y$ is a proper block map and thus its image Y is a sofic-Dyck shift by Proposition 3. □

The following corollary is a direct consequence of Proposition 4.

Corollary 1. *The class of sofic-Dyck shifts is invariant by proper conjugacy.*

We prove below that the same result holds for finite-type-Dyck shifts.

Proposition 5. *The class of finite-type-Dyck shifts is invariant by proper conjugacy.*

Proof. Let X be a finite-type-Dyck shift over A which is properly conjugate to a shift Y over B. Let Φ be a proper block map from $A^{\mathbb{Z}}$ to B^{Z} that induces a conjugacy from X to Y. Without loss of generality we may assume that there are positive integers m, n such that Φ is a proper (m,n)-block map and X is the set of sequences avoiding two finite sets F and U with $F \subseteq A^{m+n+1}$ and $U \subseteq (A^m \times A_c \times A^n) \times (A^m \times A_r \times A^n)$.

Let $\Psi = \Phi^{-1} : Y \to X$ be the proper (m', n')-block map inverse of Φ and ψ the block function of Ψ. It induces a map (still denoted by ψ) from $B^{m'+m+1+n+n'}$ to A^{m+1+n}. We set $d = m' + m$, $k = n' + n$, $r = d + 1 + k$. Let $F' = B^{5r} \setminus \mathcal{B}_{5r}(Y)$ and let $U' \subseteq (B^d \times B_c \times B^k) \times (B^d \times B_r \times B^k)$ be the set of pairs (u', v') such that $(\psi(u'), \psi(v')) \in U$. Let us show that Y is the set Z of sequences avoiding F' and U'.

By construction, $Y \subseteq Z$. Let now $z \in Z$. We prove by induction that each factor of length jr belong to $\mathcal{B}(Y)$ for $j \geq 5$. We first have by definition of Z that each factor of length $5r$ belongs to $\mathcal{B}(Y)$. Assume now that each factor of z of length jr belongs to $\mathcal{B}(Y)$ for some $j \geq 5$. Let z' be a factor of z of length $2(j-1)r$ decomposed as $z' = u'_1 u'_2 u'_3 w' v'_3 v'_2 v'_1$ with $|u'_1| = |u'_2| = |v'_1| = |v'_2| = r$, $|w'| = 2r$ and $|u'_3| = |v_3| = (j-4)r$. The factors $u' = u'_1 u'_2 u'_3 w'$ and $v' = w' v'_3 v'_2 v'_1$ of z' are of length jr and are assumed to be blocks of Y. We set $u = u_1 u_2 u_3 w_1 = \psi(u')$, where $|u_1| = |u'_1| - m'$ and $|w_1| = |w'| - n'$ and $v = w_2 v_3 v_2 v_1 = \psi(v')$, where $|w_2| = |w'| - m'$ and $|v_1| = |v_1| - n'$. Note that $w_1[m', |w_1| - 1] = w_2[0, |w_2| - n']$. Hence w_1 and w_2 overlap on a part w of length at least $|w'| - m' - n' = 2r - m' - n' \geq m + n$. Since $u', v' \in \mathcal{B}(Y)$, we have $u, v \in \mathcal{B}(X)$.

Let $\mathcal{A}(\phi, F, U)$ be the Dyck automaton defined in the proof of Proposition 3. A pair of bi-infinite sequences (x,y) is the label of a bi-infinite admissible path of $\mathcal{A}(\phi, F, U)$ if and only if $x \in X$ and $\Phi(x) = y$. Further, all finite paths of the input Dyck automaton $\mathcal{A}_1$ of $\mathcal{A}(\phi, F, U)$ which are labeled by a given block $x_1 x_2 \in \mathcal{B}(X)$ with $|x_1| = m$ and $|x_2| = n$ go through the same state after reading x_1. As a consequence, since $|w| \geq m + n$, there is a path in $\mathcal{A}_1$ labeled by $x' = u_1 u_2 u_3 w_1 t v_3 v_2 v_1 = \psi(z')$, where $w_2 = wt$. Since z' avoid U', we have x' avoids U. Hence $x' \in \mathcal{B}(X)$, implying $\phi(x') = u'_0 u'_2 u'_3 w' v'_3 v'_2 v'_0$, where u'_0 is the suffix of u'_1 of length $|u'_1| - m' - m$ and v'_0 is the prefix of v'_1 of length $|v'_1| - n' - n$. We obtain that $u'_2 u'_3 w' v'_3 v'_2 \in B(Y)$. Hence each factor of length $2(j-2)r$ of z belongs to $\mathcal{B}(Y)$ and $2(j-2)r \geq (j+1)r$ for $j \geq 5$. This proves that each factor of z belongs to $\mathcal{B}(Y)$. We get $Z = Y$ and Y is a finite-type-Dyck shift. □

5. Presentations of sofic-Dyck shifts

In this section we define several particular presentations of sofic-Dyck shifts which will be useful for the computation of zeta function.

A Dyck automaton is *deterministic*[1] if there is at most one edge starting in a given state and with a given label. Sofic shifts (see [33]) always have a deterministic presentation. Although visibly pushdown languages are accepted by deterministic visibly pushdown automata [2], sofic-Dyck shifts may not be presented by any deterministic Dyck automaton as is shown in Example 3. Indeed, the two notions of determinism do not match. The notion of determinism for visibly pushdown languages includes the stack symbol as input for return transitions of visibly pushdown automata.

Let $\mathcal{A}$ be a Dyck automaton. We define the *left reduction* of $\mathcal{A}$ as the Dyck automaton obtained through some determinization process. The process is an adaptation to Dyck automata of the determinization of visibly pushdown automata [1]. It is sketched in [8] and we detail it here.

Let $\mathcal{A} = (\mathcal{G}, M)$ with $\mathcal{G} = (Q, E)$ be a Dyck automaton over A. We define a Dyck automaton $\mathcal{D} = (\mathcal{H}, N)$ over A, where $\mathcal{H} = (Q', E')$ with $Q' = \mathfrak{P}(Q \times Q) \times \mathfrak{P}(Q)$ and $\mathfrak{P}(Q)$ is the set of subsets of Q. States are pairs (S, R) where S is called the *summary*[2] of the state and R is a nonempty subset of Q. The state $I = (\varnothing, Q)$ is called the initial state. For each state (S, R), the set S is empty if and only the admissible paths going from I to (S, R) are labeled by a matched-call word. It is nonempty if all admissible paths going from I to (S, R) are of the form

$$I \xrightarrow{u} (S", R") \xrightarrow{a} (T, U) \xrightarrow{w} (S, R),$$

where $a \in A_c$ and w is a Dyck word. If there is such a path, the summary S of the state (S, R) is the set of pairs (p, q) in $U \times R$ such that there is an admissible path of $\mathcal{A}$ labeled by the Dyck word w from p to q. In both cases, if there is a path labeled by v in $\mathcal{D}$ from I to (S, R), then R is the set of states q such that there is an admissible path in $\mathcal{A}$ labeled by v ending in q.

[1]Deterministic presentations are also called *right-resolving* in [33].
[2]The definition of summaries differs slightly from the one given in [1].

For a subset R of Q, we denote by $\mathrm{Diag}(R)$ the set of all pairs (p,p) for $p \in R$. The edges of $\mathcal{D}$ are defined as follows.

- For every $\ell \in A_i$, $((S,R),\ell,(S',R')) \in E'$ if $S' = \{(p,q) \mid \exists r \in Q, (p,r) \in S, (r,\ell,q) \in E\}$ and $R' = \{q \mid \exists p \in R, (p,\ell,q) \in E\}$ is nonempty.
- For every $a \in A_c$, $((S,R),a,(\mathrm{Diag}(R'),R')) \in E'$ if $R' = \{q \mid \exists p \in R, (p,a,q) \in E\}$ is nonempty.
- For every $b \in A_r$, the edges stating from (S,R) with $S \neq \varnothing$ labeled by b are defined as follows. For any edge $((S'',R''),a,(T,U))$ with $a \in A_c$ we define
  - Update $= \{(p,p') \mid \exists p_1,p_2 \colon (p,a,p_1) \in E, (p_1,p_2) \in S, (p_2,b,p') \in E, ((p,a,p_1),(p_2,b,p')) \in M\}$,
  - $S' = \{(p,q) \mid \exists p', (p,p') \in S", (p',q) \in \text{Update}\}$,
  - $R' = \{q \mid \exists p \in R", (p,q) \in \text{Update}\}$.

  If R' is not empty, we define an edge $((S,R),b,(S',R')) \in E'$ and set this edge matched with $((S",R"),a,(T,U))$.
- For every $b \in A_r$, we define an edge $((\varnothing,R),b,(\varnothing,V)) \in E'$ where $V = \{q \mid \exists p \in R, (p,b,q) \in E\}$ is nonempty. This return edge is not matched with any call edge.

We only keep in $\mathcal{D}$ the states reachable from I.

Proposition 6. *The left reduction of a Dyck automaton $\mathcal{A}$ presents the same sofic-Dyck shift as $\mathcal{A}$.*

Proof. Let X be the sofic-Dyck presented by $\mathcal{A}$ and $\mathcal{D}$ be the left reduction of $\mathcal{A}$. Let v be the label of an admissible path of $\mathcal{A}$ going from p to q. By construction there is an admissible path of $\mathcal{D}$ labeled by v going from I to some state (S,R) with $q \in R$. Thus labels of finite admissible paths of $\mathcal{A}$ are labels of finite admissible paths of $\mathcal{D}$.

Conversely, let v be the label of some finite admissible path π of $\mathcal{D}$. We claim that v is the label of an admissible path of $\mathcal{A}$.

We prove the claim by recurrence on the length of v. It is true if v is the empty word. Let $v = uc$ where $c \in A$ and $\pi = (S_1,R_1) \xrightarrow{u} (S,R) \xrightarrow{c} (S',R')$. By induction hypothesis, we assume that for any state $r \in R$ there is an admissible path labeled by u from some state $q \in R_1$ to r. If the edge $((S,R),c,(S',R'))$ is a call or internal edge or is a return edge not matched with a call edge of π, the result holds by construction for uc. Let us assume that

$$\pi = (S_1,R_1) \xrightarrow{u} (S",R") \xrightarrow{a} (T,U) \xrightarrow{w} (S,R) \xrightarrow{b} (S',R'),$$

where w is a Dyck word over A, $a \in A_c$, and $b \in A_r$. By induction hypothesis, we assume that for any state $p \in R"$ there is an admissible path $q \xrightarrow{u} p$ in $\mathcal{A}$ for some $q \in R_1$.

For any $r \in R'$ there are $p \in R''$ and $(p_1, p_2) \in S$ such that (p, a, p_1) and (p_2, b, r) are matched in $\mathcal{A}$. Further, S is the set of pairs $(s, s') \in U \times R$ such that there is an admissible path in $\mathcal{A}$ labeled by w from s to s'. It follows that there is in $\mathcal{A}$ an admissible path labeled by w from p_1 to p_2 and thus an admissible path $q \xrightarrow{u} p \xrightarrow{a} p_1 \xrightarrow{w} p_2 \xrightarrow{b} r$ in $\mathcal{A}$ which concludes the proof of the claim. □

Note that since the label of an admissible path of $\mathcal{D}$ is the label of an admissible path of $\mathcal{A}$, each label of an admissible path of $\mathcal{D}$ is the label of an admissible path of $\mathcal{D}$ starting at I.

We similarly define the *right reduction* of $\mathcal{A}$ with a co-determinization of $\mathcal{A}$ and an exchange of roles played by call and return edges. Note the left reduction of $\mathcal{A}$ may have more states than $\mathcal{A}$.

Let L be a language of finite words. A Dyck automaton is *L-deterministic* if there is at most one admissible path starting in a given state and with a given label in L.

By construction the left reduction of a Dyck automaton is A_c-deterministic and A_i-deterministic.

A Dyck automaton is *weak-deterministic* if there is a state I such that for any word u there is at most one admissible path labeled by u starting at I.

Proposition 7. *The left reduction of a Dyck automaton is weak-deterministic.*

Proof. Let $\mathcal{D}$ be the left reduction of a Dyck automaton $\mathcal{A}$ and let I be the initial state of $\mathcal{D}$. Let us suppose that the property is false. We consider two minimal-length distinct admissible paths starting at I and sharing the same label.

$$I \xrightarrow{u} (S, R) \xrightarrow{b} (T, U),$$
$$I \xrightarrow{u} (S', R') \xrightarrow{b} (T', U'),$$

with $b \in A$ and $(T, U) \neq (T', U')$. We may assume $(S, R) = (S', R')$ since these paths are of minimal length. Since $\mathcal{D}$ is A_c-deterministic and A_i-deterministic, we may assume that $b \in A_r$. By definition, we have $U = U'$. If ub is matched-call, T and T' are empty, hence $(T, U) = (T', U')$. If $ub = u'awb$, where w is a Dyck word and $a \in A_c$, the two above paths are

$$I \xrightarrow{u'} (S_1, R_1) \xrightarrow{a} (S_2, R_2) \xrightarrow{w} (S, R) \xrightarrow{b} (T, U),$$
$$I \xrightarrow{u'} (S_1, R_1) \xrightarrow{a} (S_2, R_2) \xrightarrow{w} (S, R) \xrightarrow{b} (T', U).$$

Since the paths are admissible, $((S, R), b, (T, U))$ is matched with $((S_1, R_1), a, (S_2, R_2))$ and $((S, R), b, (T', U))$ is matched with $((S_1, R_1), a, (S_2, R_2))$. By definition of the summary we get $T = T'$, a contradiction. □

Corollary 2. *The left reduction of a Dyck automaton over A is Dyck(A)-deterministic.*

Proof. Let (S, R) be a state of the left reduction of a Dyck automaton over A and w be a Dyck word over A. Let us assume that there are two admissible paths π_1 and π_2 labeled by w starting at (S, R). Since there is an admissible path π from I to (S, R), the paths $\pi\pi_1$ and $\pi\pi_2$ are two admissible paths starting at I. They are then equal by Proposition 7. □

Example 2. The Dyck automaton $\mathcal{A}$ on the left of Figure 2 has as left reduction the Dyck automaton on the right of the picture. The initial state is the state $I = (\varnothing, \{1, 2, 3\})$.

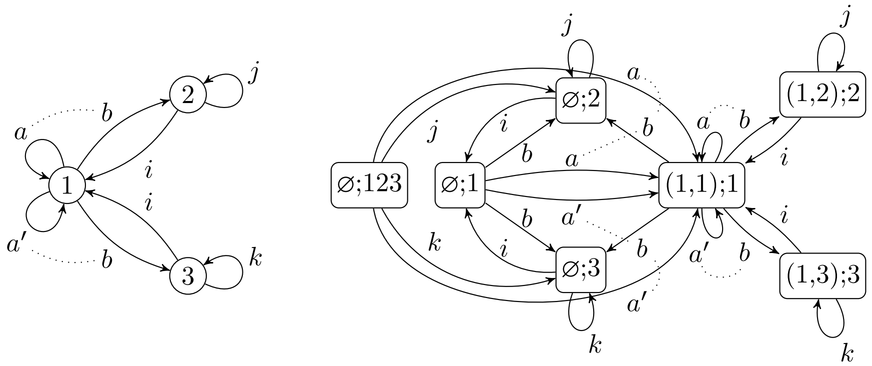

Figure 2: A Dyck automaton $\mathcal{A}$ (on the left) over $A = A_c \sqcup A_r \sqcup A_i$ with $A_c = \{a, a'\}$, $A_r = \{b\}$ and $A_i = \{i, j, k\}$. The left reduction of $\mathcal{A}$ (on the right) over the same tri-partitioned alphabet. Matched edges are linked with a dotted line and each state is represented by its summary set S of pairs of edges and the set R.

Example 3. The sofic-Dyck shift X presented by the Dyck automaton $\mathcal{A}$ of Figure 3 has no deterministic presentation. Let us briefly give a sketch of the proof

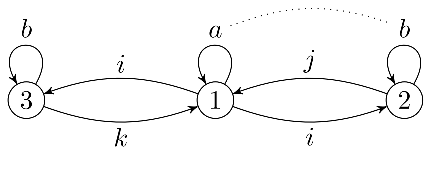

Figure 3: A Dyck automaton $\mathcal{A}$ over A, with $A_c = \{a\}$, $A_r = \{b\}$ and $A_i = \{i, j, k\}$, presenting a sofic-Dyck shift which has no deterministic presentation. Matched edges are linked with a dotted line.

of this fact. Let $\mathcal{B}$ be a deterministic Dyck automaton over $\mathcal{A}$ accepting the same shift X. For any positive integers n, m, r, the words $(a^{n+m} i b^n j)^r (i b^m j)^r (i b^m k)^r$ are blocks of X and are thus factors of labels of admissible paths in $\mathcal{B}$, the edges labeled by b of the path labeling $(i b^m j)^r$ being matched with the edges labeled by a of the path labeling $(a^{n+m} i b^n j)$. If $\mathcal{B}$ is finite and deterministic, this implies that there are $m, r, s > 0$ and $n' > n > 0$ such that $(a^{n'+n+m} i b^n j)^r (i b^m j)^r (i b^m k)^r$ is a block of X, a contradiction.

5.1. Visibly pushdown shifts

In this section we show that the class of sofic-Dyck shifts is the class of *visibly pushdown shifts*, *i.e.* the subshifts whose set of blocks are factorial extensible visibly pushdown languages.

The class of visibly pushdown languages of finite words can be described either by pushdown automata or by context-free grammars.

A *visibly pushdown automaton* on finite words over $A = A_c \sqcup A_r \sqcup A_i$ is a tuple $M = (Q, I, \Gamma, \Delta, F)$ where Q is a finite set of states, $I \subseteq Q$ is a set of initial states, Γ is a finite stack alphabet that contains a special bottom-of-stack symbol $\perp$, $\Delta \subseteq (Q \times A_c \times Q \times (\Gamma \setminus \{\perp\})) \cup (Q \times A_r \times \Gamma \times Q) \cup (Q \times A_i \times Q)$, and $F \subseteq Q$ is a set of final states.

A transition (p, a, q, γ), where $a \in A_c$ and $\gamma \neq \perp$, is a push-transition. On reading a, the stack symbol γ is pushed onto the stack and the control changes from state p to q. A transition (p, a, γ, q) is a pop-transition. The symbol γ is read from the top of the stack and popped. If $\gamma = \perp$, the symbol is read but not popped. A transition (p, a, q) is a local action.

A stack is a nonempty finite sequence over Γ starting with $\perp$. A *run* of M labeled by $w = a_1 \,.\,. a_k$ is a sequence $(p_0, \sigma_0) \cdots (p_k, \sigma_k)$ where $p_i \in Q$, $\sigma_0 = \perp$, $\sigma_i \in (\Gamma \setminus \{\perp\})$ for $1 \leq i \leq k$, and such that:

- If a_i is call symbol, then there are $\gamma_i \in \Gamma$ and $(p_{i-1}, a_i, p_i, \gamma_i) \in \Delta$ with $\sigma_i = \sigma_{i-1} \cdot \gamma_i$.
- If a_i is a return symbol, then there are $\gamma_i \in \Gamma$ and $(p_{i-1}, a_i, \gamma_i, p_i) \in \Delta$ with either $\gamma_i \neq \perp$ and $\sigma_i \cdot \gamma_i = \sigma_{i-1}$ or $\gamma_i = \perp$ and $\sigma_i = \sigma_{i-1} = \perp$.
- If a_i is an internal symbol, then $(p_{i-1}, a_i, p_i) \in \Delta$ and $\sigma_i = \sigma_{i-1}$.

A run is *accepting* if $p_0 \in I$, $\sigma_0 = \perp$, and the last state is final, *i.e.* $p_k \in F$. A word over A is *accepted* if it is the label of an accepting run. The language of words accepted by M is denoted by $L(M)$. The language accepted by a visibly pushdown automaton is called a *visibly pushdown language*.

We will use also the following grammar-based characterization (see [2]).

A context-free grammar over an alphabet A is a tuple $G = (V, S, P)$, where V is a finite set of variables, $S \in V$ is a start variable and P is a finite set of rules of the form $X \to \alpha$ such that $X \in V$ and $\alpha \in (V \cup A)^*$. The semantics of the grammar G is defined by the derivation relation $\to$ over $(V \cup A)^*$. If $X \to \alpha$ is a rule and β, β' are words of $(V \cup A)^*$, then $\beta X \beta' \to \beta \alpha \beta'$ holds. The language *accepted by the grammar* G, denoted $L(G)$ is the set of words u in B^* such that $S \overset{*}{\Longrightarrow} u$, where $\overset{*}{\Longrightarrow}$ is the transitive closure of the relation $\to$.

Let A be a tri-partitioned alphabet. A context-free grammar $G = (V, S, P)$ over A is a *visibly pushdown grammar* with respect to A if the set V of variables is partitioned into two disjoint sets V^0 and V^1, such that all rules in P are of one of the following forms

- $X \to \varepsilon$;
- $X \to aY$, such that if $X \in V^0$, then $a \in A_i$ and $Y \in V^0$;
- $X \to aYbZ$, such that $a \in A_c$, $b \in A_r$, $Y \in V^0$, and if $X \in V^0$, then $Z \in V^0$.

The variables in V^0 derive only Dyck words. The variables in V^1 derive words that may contain unmatched call letters as well as unmatched return letters. In the rule $X \to aY$, if a is a call it is unmatched and the variable X must be in V^1 if a is a call or return. In the rule $X \to aYbZ$, the symbols a and b are the matching call and return. The words generated by Y belong to V^0 and thus are Dyck words. Furthermore, if X is required to generate Dyck words, then Z also.

It is shown in [2] that a language is visibly pushdown language if and only if it is accepted by a visibly pushdown grammar.

Proposition 8. *The set of labels of finite admissible paths of a Dyck automaton is a visibly pushdown language.*

Proof. Let $\mathcal{A} = (\mathcal{G}, M)$ be a Dyck automaton over A, where $\mathcal{G} = (Q, E)$.

We define a visibly pushdown automaton $V = (Q, I, \Gamma, \Delta, F)$ over A, where $I = F = Q$ and Γ is the set of edges of $\mathcal{A}$. The set of transitions Δ is obtained as follows.

- If $(p, a, q) \in E$ with $a \in A_c$, then $(p, a, q, (p, a, q)) \in \Delta$.
- If $(p, b, q) \in E$ with $b \in A_r$, then $(p, b, \gamma, q) \in \Delta$ for each call edge γ which is matched with the return edge (p, b, q).
- If $(p, \ell, q) \in E$ with $\ell \in A_i$, then $(p, \ell, q) \in \Delta$.

Let w be a finite word over A. There is run $(p_0, \sigma_0)\cdots(p_k, \sigma_k)$ in V labeled by w such that $\sigma_0 = \bot$, $p_0 = p$ and $p_k = q$ if and only if w be the label of an admissible path π of $\mathcal{A}$ going from p to q. Thus w is the label of an admissible path of $\mathcal{A}$ if and only if it is the label of an accepting run of V, which proves the proposition. □

In order to prove that the set of blocks of sofic-Dyck shift is a visibly pushdown language, we have to prove that the subset of words labeling a finite admissible path which are extensible to labels of bi-infinite admissible paths is also a visibly pushdown language.

Let L be a language of finite words over A. We denote by $\mathcal{E}(L)$ the set words $w \in L$ such that, for any integer n, there are words u, v of length greater than n such that $uwv \in L$. Note that $\mathcal{E}(L)$ is a factorial language when L is factorial. This set is called in [16] the *bi-extensible* subset of L.

We show below that the bi-extensible subset of a factorial visibly pushdown language is a visibly pushdown language. It is shown in [16] that it is not true that the bi-extensible subset of a context-free language is a context-free language but the result holds for factorial context-free languages. We prove a similar result for factorial visibly pushdown languages.

We first recall the following pumping lemma (see [16, Lemma 5.6]).

Lemma 2. *Let $G = (V, S, P)$ be a context-free grammar and $L = L(G)$. Then for any integer $t > 0$, there exists an integer $p(t)$ such that for each $z \in L$ and any set K of distinguished positions in z, if $|K| \geq p(t)$, then there is a decomposition $z = ux_1 \cdots x_t w y_1 \cdots y_t v$ such that*

- *There exists a variable $X \in V$ such that*

$$S \xrightarrow{*} uXv \xrightarrow{*} ux_1Xy_1v \xrightarrow{*} \cdots \xrightarrow{*} ux_1 \cdots x_t X y_t \cdots y_1 v \xrightarrow{*} ux_1 \cdots x_t w y_t \cdots y_1 v.$$

- *For any $i_1, .., i_t$, we have $ux_1^{i_1} \cdots x_t^{i_t} w y_t^{i_t} \cdots y_1^{i_1} v \in L$.*
- *If $K(x)$ denotes the distinguished positions of K in a word x, then either $K(u), K(x_1), .. K(x_t), K(w) \neq \emptyset$, or $K(w), K(y_t), .. K(y_1)$, $K(v) \neq \emptyset$. We also have $|K(x_1) \cup .. K(x_t) \cup K(w) \cup K(y_t) \cup .. K(y_1)| \leq p(t)$.*

Proposition 9. *If L is a factorial visibly pushdown language, then $\mathcal{E}(L)$ is a factorial visibly pushdown language.*

Proof. Let $G = (V, S, P)$ be a visibly pushdown grammar over A accepting L. We define a grammar $G' = (V \cup \{X_i\}, S, P')$ over $A' = (A_c \cup \{\$_1\}, A_r \cup \{\overline{\$_1}\}, A_i \cup \{\$_0\})$ obtained by adding the following rules to G:

- $X \to \$_1 X \overline{\$_1} X_1$ and $\mathsf{X}_1 \to \varepsilon$, for each $X \in V^0$ such that $X \xrightarrow[G]{*} uXv$ with $u, v \in A^+$,
- $X \to \$_0 X$, for each $X \in V$ such that $X \xrightarrow[G]{*} uX$ with $u \in A^+$.

Note that it is not possible to have a rule $X \in V$ such that $X \xrightarrow{*} Xu$ with $u \in A^+$. The grammar G' is a visibly pushdown grammar over A'.

Let $L_1 = \{w \in A^* \mid \exists w_i \in A^*,\ w_1 \$ w_2 w w_3 \$' w_4 \in L(G'), \$ = \$_0$ or $\overline{\$_1}, \$' = \$_0$ or $\$_1\}$, $L_2 = \{w \in A^* \mid \exists w_i \in A^*, w_1 \$_1 w_2 w w_3 \overline{\$_1} w_4, \in L(G')\}$ and $L_3 = L_1 \cup L_2$.

Let us prove that $L_3 \subseteq \mathcal{E}(L)$. We first consider a word $w \in L_2$ such that $w_1 \$_1 w_2 w w_3 \overline{\$_1} w_4 \in L(G')$. Then $w_2 w w_3$ is generated in G by some variable $X \in V^0$ such that $X \xrightarrow{*} uXv$, for $u, v \in A^+$. Thus, for any integer n, we have $u^n w_2 w w_3 v^n \in L$. Thus $w \in \mathcal{E}(L)$.

Let us consider a word $w \in L_1$ such that $w_1 \overline{\$_1} w_2 w w_3 \$_1 w_4 \in L(G')$. Thus there are words u_1, u_2, u_3, u_4 such that $u_1 \$_1 u_2 \overline{\$_1} w_2 w w_3 \$_1 u_3 \overline{\$_1} u_4 \in L(G')$. It follows that there are words $x, y, z, t \in A^+$ such that $u_1 x^n u_2 y^n w_2 w w_3\ z^n u_3 t^n v_4 \in L$, for any positive integer n. Thus $w \in \mathcal{E}(L)$.

We now consider the case where $w \in L_1$ with $w_1 \$_0 w_2 w w_3 \$_0 w_4 \in L(G')$. Then there are variables X, Y such that $X \xrightarrow{*} uX$ for some $u \in A^+$, $Y \xrightarrow{*} vY$ with $v \in A^+$, such that $S \xrightarrow{*} \alpha X \beta Y \gamma$ and $X\beta \xrightarrow{*} w_2 w w_3$. It follows that, for any positive integer n, we have $u^n w_2 w w_3 v^n \in L$. Hence $w \in \mathcal{E}(L)$. The remaining cases are proved similarly.

We now prove that $\mathcal{E}(L) \subseteq L_3$. Let $z \in \mathcal{E}(L)$ of length t. We choose $z_1, z_2 \in A^+$ of length greater than $p(t)$, where $p(t)$ is defined in Lemma 2, such that

$z' = z_1 z z_2 \in L$. For technical reasons that will appear below, we also choose $|z_2| > 4|K|(|z_1| + |z|)$.

We consider a set of distinguished positions in z_1. By Lemma 2, there is a variable X in V such that

$$S \xrightarrow{*} uXv \xrightarrow{*} ux_1Xy_1v \xrightarrow{*} \cdots ux_1\cdots x_tXy_t\cdots y_1v \xrightarrow{*} ux_1\cdots x_twy_t\cdots y_1v = z'.$$

Let T be the induced derivation tree and T' be the subtree of T (labeled by X) generating w. Let π be the path going from the root of T to the parent of the root of T'. The length ℓ of π is at most $|ux_1\cdots x_t|$ since all rules of G produce either the empty word or a non empty word over $V \cup A$ with a terminal symbol on the left.

At least one of two following cases holds.

- $K(u), K(x_1), .., K(x_t), K(w) \neq \varnothing$.
  - If z is a factor of $wy_t\cdots y_1$ and $X \to \$_1 X\overline{\$_1}X_1$ is a rule of G', then $u\$_1x_1\cdots x_t\ wy_t\cdots y_1\overline{\$_1}v \in L(G')$ and thus $z \in L_3$.
  - If z is a factor of $wy_t\cdots y_1$ and $X \to \$_1X\overline{\$_1}X_1$ is not a rule of G'. Then $y_t\cdots y_1 = \varepsilon$ and $X \to \$_0X$ is a rule of G'. Thus the word $\hat{z}$ obtained after inserting $\$_0$ between u and x_1 is still in $L(G')$.

    Furthermore, z is a factor of w. We set $w = w_1zw_2$. If $|w_2| < |K|$, $|y_t\cdots y_1v| = |z_2| - |w_2| > 4|K| \times |z_1z| - |K| \geq 3|K| \times |ux_1\cdots x_t| \geq 3\ell|K|$. We denote by R the set of nodes in T which are children of nodes of π on the right of π and thus generate $y_t\cdots y_1v$. The size of the set R is at most 3ℓ since all rules of G have an arity at most 4.

    At most 3ℓ variables generating a sequence of length greater than $3\ell|K|$, there is a variable Y in R such that Y generates a factor of length at least $|K|$ of $y_t\cdots y_1v$ which is a factor of z_2. We do a second pumping for words generated by Y using distinguished positions on $y_t\cdots y_1v$ and get that the word obtained from $\hat{z}$ after inserting either $\$_0$ or $\$_1$ in $y_t\cdots y_1v$ is still in $L(G')$.

    If $|w_2| \geq |K|$, we do a second pumping for words generated by X using distinguished positions on w_2 and get that the word obtained from $\hat{z}$ after inserting either $\$_0$ or $\$_1$ in w_2 is still in $L(G')$.
  - If z is a factor of $y_t\cdots y_1v$, then w can be replaced either by $\overline{\$_1}w$ or by $\$_0w$ and gives a word $\hat{z}$. A similar argument as above for a second pumping still holds. We have this time $|z_2| > 4|K| \times |z_1z| \geq 4|K| \times |ux_1\cdots x_t| \geq 3\ell|K|$. Hence there is a variable Y in R such that Y generates a factor of length at least $|K|$ of z_2. We do a second pumping for words generated by Y using distinguished positions on z_2 and get that the word obtained from $\hat{z}$ after inserting either $\$_0$ or $\$_1$ in z_2 is still in $L(G')$.
  - Otherwise z is a factor of $wy_t\cdots y_1v$ and z crosses w, $y_t\cdots y_1$ and v. Then $|y_t\cdots y_1| < |z| = t$. Thus there is $1 \leq i \leq t$ such that $y_i = \varepsilon$. So we

can replace x_i by $\$_0 x_i$ obtaining $\hat{z}$. Again here there is a variable Y in R such that Y generates a factor of length at least $|K|$ of z_2. We do a second pumping for words generated by Y using distinguished positions on z_2 and get that the word obtained from $\hat{z}$ after inserting either $\$_0$ or $\$_1$ in z_2 is still in $L(G')$.

- $K(w), K(y_t), .., K(y_1), K(v) \neq \varnothing$. Then z is a factor of v since the distinguished positions are on z_1. Then w can be replaced either by $\overline{\$_1} w$ or by $\$_0 w$. The second pumping is done as the in the last item of the previous case.

We obtain that for any $z \in \mathcal{E}(L)$, there is in $L(G')$ either a word of the form $w_1 \$ w_2 z w_3 \$' w_4$ with $\$ = \$_0$ or $\overline{\$_1}$, $\$' = \$_0$ or $\$_1$ or a word of the form $w_1 \$_1 w_2 z w_3 \overline{\$_1} w_4$. Thus $z \in L_3$. Hence $L_3 = \mathcal{E}(L)$.

We now show that L_3 is a visibly pushdown language. Indeed, let us show that $L_{\$\$'} = \{w \in A^* \mid w_1 \$ w_2 w w_3 \$' w_4 \in L(G')\}$ is visibly pushdown.

Let $L' = \mathrm{Fact}(L(G')) \cap \$ A^* \$'$, where $\mathrm{Fact}(L(G'))$ denotes the set of factors of $L(G')$, and let $L" = \mathrm{Fact}(L') \cap A^*$. Since the class of visibly pushdown languages is closed by prefix and suffix, it is closed by factor. Hence the languages L' and $L"$ are visibly pushdown. We have $L_{\$\$'} = L"$.

As a consequence, $L_{\$\$'}$ is visibly pushdown. The class of visibly pushdown languages being closed by union, we get that L_3 is visibly pushdown.

Note that that G' can be constructed in an effective way since it is decidable whether $X \xrightarrow{*} uXv$ or $X \xrightarrow{*} uX$ for some words $u, v \in A^+$. □

Theorem 1. *Let X be a sofic-Dyck shift. Then $\mathcal{B}(X)$ is a visibly pushdown language. Conversely, if L is a factorial extensible visibly pushdown language, then $\mathcal{B}^{-1}(L)$ is a sofic-Dyck shift.*

Proof. Let X be the sofic-Dyck shift presented by a Dyck automaton $\mathcal{A}$. By Propositions 8, the set L of labels of finite admissible paths of $\mathcal{A}$ is a visibly pushdown language. By 9, the language $\mathcal{E}(L)$ also. By Lemma 1, we have $\mathcal{B}(X) = \mathcal{E}(L)$ and thus $\mathcal{B}(X)$ is a visibly pushdown language.

Conversely, let L be a factorial extensible visibly pushdown language. Let $G = (V, S, P)$ be a visibly pushdown grammar over A accepting L. We may assume that variables that do not generate any word are discarded. We define a Dyck automaton $\mathcal{A} = (\mathcal{G}, M)$ with $\mathcal{G} = (V \cup (V \times (\{\$\} \cup (A \times V)), E)$ as follows. We denote below by $(X, \circ)$ any state which is either X or $(X, \$)$, or $(X, (a, Y))$.

- If $X \to \ell Y \in P$ with $\ell \in A_i$, then $((X, \circ), \ell, Y) \in E$.
- If $X \to aY \in P$ with $a \in A_c$, then $((X, \circ), a, (Y, \$)) \in E$.
- If $X \to aYbZ \in P$, then $((X, \circ), a, (Y, (b, Z))) \in E$.
- If $X \to bY \in P$ with $b \in A_r$, then $((X, \circ), b, Y) \in E$.
- If $b \in A_r$, $Z \to \varepsilon$ and $Z \in V^0$, then $((Z, \circ), b, T) \in E$ for any $T \in V$. Each of these edges is also matched with each edge of the form $((X, \circ)), a, (Y, (b, T)))$.

Note that all states $(X, \circ)$ have the same outgoing edges. A state $(X, \circ)$ is *nullable* if X generates the empty word.

We claim that if w is a word generated by X in G, there is an admissible path in $\mathcal{A}$ labeled by w from X to some nullable state T or $(T, \$)$.

The proof is by induction on the size of w. Let us first consider the case $w = \varepsilon$. If w is generated by Z, then $Z \to \varepsilon$ is a rule of G. Thus the claim is true.

If w is nonempty, since w is generated by X, then either $X \to aY \in P$, $w = aw_1$ with w_1 is generated by Y and a is not matched with symbols of w_1, or $X \to aYbZ \in P$, $w = aw_1bw_2$ and w_1, w_2 are generated by Y and Z respectively, with $Y \in V^0$.

In the first case, there is an edge $(X, a, (Y, \$))$. By induction, there is an admissible path in $\mathcal{A}$ from Y to some nullable state T or $(T, \$)$. Thus there is an admissible path labeled by w_1 from $(Y, \$)$ to some nullable state T or $(T, \$)$ and thus there is an admissible path labeled by w from X to some nullable state T or $(T, \$)$.

In the second case, there is an edge $(X, a, (Y, (b, Z)))$. By induction, there is an admissible path labeled by w_1 from $(Y, (b, Z))$ to some nullable state T or $(T, \$)$. There is also an admissible path labeled by w_2 from $(Z, \circ)$ to some nullable state U or $(U, \$)$ and an edge $((T, \circ), b, Z)$. Thus we obtain the path

$$X \xrightarrow{a} (Y, (b, Z)) \xrightarrow{w_1} T(\text{or } (T, \$)) \xrightarrow{b} Z \xrightarrow{w_2} U(\text{or } (U, \$)).$$

Since T is nullable and in V^0, any edge $((T, \circ), b, Z)$ is matched with $(X, a, (Y, (b, Z)))$, this path is an admissible path labeled by w going from X to either U or $(U, \$)$. Thus L is included in the set of labels of admissible paths of $\mathcal{A}$.

Conversely, let w be the label of an admissible path π in $\mathcal{A}$ starting at a state $(X, \circ)$. Then w is a prefix of a word generated by X in G. If w is moreover a Dyck word and X is nullable, then w is generated by X.

The proof is again by induction on the size of w. Note that it holds for the empty word. We first decompose π into one of the following paths:

1. $(X, \circ) \xrightarrow{a} Y \xrightarrow{w_1} (U, \circ)$, with $a \in A_i$,
2. $(X, \circ) \xrightarrow{a} (Y, \$) \xrightarrow{w_1} (U, \circ)$, $a \in A_c$ not matched with letters of w_1
3. $(X, \circ) \xrightarrow{a} Y \xrightarrow{w_1} (U, \circ)$, with $a \in A_r$,
4. $(X, \circ) \xrightarrow{a} (Y, (b, Z)) \xrightarrow{w_1} (U, \circ)$, a not matched with letters of w_1,
5. $(X, \circ) \xrightarrow{a} (Y, (b, Z)) \xrightarrow{w_1} (T, \circ) \xrightarrow{b} Z \xrightarrow{w_2} (U, \circ)$, with $a \in A_c, b \in A_r$, and w_1 is a Dyck word.

In Cases (1) to (3), by induction, w_1 is a prefix of a word generated by Y and there is a rule $X \to aY$ in G. Thus aw_1 is a prefix of a word generated by X. If w is a Dyck word and U is nullable, then $w = aw_1$, where w_1 is a Dyck word $a \in A_i$. By induction hypothesis, the word w_1 is generated by Y and thus w is generated by X.

In Case (4), by induction, w_1 is a prefix of a word generated by Y and there is a rule $X \to aYbZ$ in G. Thus aw_1 is a prefix of a word generated by X. The word w is never a Dyck word.

In Case (5), there is a rule $X \to aYbZ$ in G and the edges $((X, \circ), a, (Y, (b, Z))$ and $((T, \circ), b, Z)$ are matched Thus T is nullable and in V^0. By induction w_2 is a prefix of a word generated by Z. Since w_1 is a Dyck word, by induction again, w_1 is generated by Y. It follows that w is a prefix of a word generated by X. If w is a Dyck word, then w_2 is a Dyck word and thus w_2 is generated by Z. As a consequence, w is generated by X.

Thus labels of admissible paths of $\mathcal{A}$ are prefixes of words of L. Since L is factorial, they belong to L. As a consequence L is the set of labels of finite admissible paths of $\mathcal{A}$. By definition, $\mathcal{B}^{-1}(L)$ is the set of infinite sequences whose finite factors belong to L and thus $\mathcal{A}$ presents $\mathcal{B}^{-1}(L)$. □

This gives the following characterization of sofic-Dyck shifts.

Theorem 2. *Sofic-Dyck shifts over A are shifts X_F where F is a visibly pushdown language over A.*

Proof. If X is a sofic-Dyck shift over A, then Theorem 1 says that $\mathcal{B}(X)$ is a visibly pushdown language over A. Let $F = A^* \setminus \mathcal{B}(X)$. Since visibly pushdown languages are closed y complementation, F is visibly pushdown and $X = X_{\mathcal{F}}$.

Conversely, if $X = \mathsf{X}_F$ where F is a visibly pushdown language. Let $L = A^* \setminus F$ which is a factorial visibly pushdown language. The set $\mathcal{B}(X)$ is the set of extensible words of L and is thus visibly pushdown. Thus X is sofic-Dyck. □

Proposition 10. *It is decidable whether a sofic-Dyck shift is empty.*

Proof. Let X be a sofic-Dyck shift. By Theorem 1, the set of blocks of X is generated by a context-free grammar which is furthermore computable from some Dyck automaton accepting the sofic-Dyck shift. Since the emptiness is decidable for a language generated by a context-free grammar, the emptiness of X is decidable. Indeed, X is nonempty if and only if its set of blocks is nonempty. □

6. Zeta function of sofic-Dyck shifts

Zeta functions count the periodic orbits of subshifts and constitute stronger invariants by conjugacies than the entropy (see [33]).

In this section, we give an expression of the zeta function of a sofic-Dyck shift which extends the formula obtained by Krieger and Matsumoto in [31] for Markov-Dyck shifts. The proof of Krieger and Matsumoto is based on Markov-Dyck codes which encode periodic sequences. We use a similar encoding to compute the zeta function of sofic-Dyck shifts.

As counting periodic points for sofic shifts is trickier than for shifts of finite type, counting periodic points of sofic-Dyck shifts is also trickier than for Markov-Dyck of finite-type-Dyck shifts.

6.1. Definition and general formula

The *zeta function* $\zeta_X(z)$ of the shift X is defined as the zeta function of its set of periodic patterns, *i.e.*

$$\zeta_X(z) = \exp \sum_{n\geq 1} p_n \frac{z^n}{n},$$

where p_n the number of sequences of X of period n, *i.e.* of sequences x such that $\sigma^n(x) = x$. Note that n may not be the smallest period of x.

Call *periodic pattern* of X a word u such that the bi-infinite concatenation of u belongs to X and denote $P(X)$ the set of periodic patterns of X. These definitions are extended to σ-invariant sets of bi-infinite sequences which may not be shifts (*i.e.* which may not be closed subsets of sequences).

Let $\mathcal{A}$ be a Dyck automaton over a tri-partitioned alphabet A.

We say that a Dyck word w over A is *prime* if it is nonempty and any Dyck word prefix of w is w or the empty word. We denote by $\text{Prime}(A)$ the set of prime Dyck words over A and by $\text{Prime}(X)$ the set of prime Dyck words which are blocks of a shift X.

We define the following matrices where Q is the set of states of $\mathcal{A}$.

- $C = (C_{pq})_{p,q\in Q}$ where C_{pq} is the set of prime Dyck words labeling an admissible path from p to q in $\mathcal{A}$.
- $M_c = (M_{c,pq})$, (resp. M_r) where $M_{c,pq}$ is the sum of call (resp. return) letters labeling an edge from p to q in $\mathcal{A}$.

Let H be one of the matrices C, $CM_c{}^*$, M_c, $M_r{}^*C$ or M_r. We call *H-path* a path $(p_i, c_i, p_{i+1})_{i\in I}$ in $\mathcal{A}$, where I is $\mathbb{Z}$ or an interval and $c_i \in H_{p_i p_{i+1}}$. Note that an H-path is admissible. We denote by X_H be the σ-invariant set containing all of sequences labeling a bi-infinite H-path of $\mathcal{A}$.

Proposition 11. *Let X be a the sofic-Dyck shift accepted by a Dyck automaton $\mathcal{A}$. We have $P(X) = P(\mathsf{X}_{M_c}) \sqcup P(\mathsf{X}_{M_r}) \sqcup ((P(\mathsf{X}_{CM_c{}^*}) \cup p(\mathsf{X}_{M_r{}^*C})))$, and $P(X_C) = P(\mathsf{X}_{CM_c{}^*}) \cap P(\mathsf{X}_{M_r{}^*C})$, where $\sqcup$ denotes a disjoint union.*

For a finite word u, we denote the *balance* of u by $\text{bal}(u)$. It is the difference between the number of letters of u in A_c and the number of letters of u in A_r. A word u is *positive* if $\text{bal}(u) > 0$ and $\text{bal}(v) \geq 0$ for any prefix v of u. We say that u and v are *conjugate* if they are words w, t such that $u = wt$ and $v = tw$.

Proof. Let us assume that a sequence x of X is equal to $u^\infty = \cdots uu \cdot uu \cdots$. Let $u = u_0 u_1 \cdots u_{n-1}$ where u_i are letters. We consider the following three cases.

- If $\text{bal}(u) = 0$, then u is conjugate to a word in $\text{Prime}(X)^*$ and thus x is a periodic point of X_C.
- If $\text{bal}(u) > 0$, then u is conjugate to a word v such that $\text{bal}(v_0 \ldots v_i) \geq 0$ for any $0 \leq i \leq n-1$. If $v \in A_c^+$, then x is a periodic point of X_{M_c}. If $v \notin A_c^+$, there are two indices $0 \leq m_1 < m_2 \leq n-1$ such that $\text{bal}(v_0 \ldots v_{m_1}) = \text{bal}(v_0 \ldots v_{m_2})$.

Let (m_1, m_2) two such indices with moreover $\mathrm{bal}(v_0 \mathinner{..} v_{m_1}) = \mathrm{bal}(v_0 \mathinner{..} v_{m_2})$ minimal. Let $w = v_{m_1} \cdots v_{n-1} v_0 \cdots v_{m_1 - 1}$. The word w is again a conjugate of v and u.

Let j_1 be the largest integer less than or equal to $n-1$ such that $w_0 \cdots w_{j_1}$ has a suffix in $\mathrm{Prime}(X)$ and i_1 be the smallest integer such that $w_{i_1} \cdots w_{j_1} \in \mathrm{Prime}(X)^*$. Then $w_{i_1} \cdots w_{n-1} \in \mathrm{Prime}(X)^+ A_c^*$ and $w_0 \cdots w_{i_1 - 1}$ is a positive word. We define indices i_2, j_2 similarly for the word $w_0 \cdots w_{i_1-1}$ and thus iteratively decompose w into a product of words in $\mathrm{Prime}(X) A_c^*$. It follows that x belongs to $\mathsf{X}_{CM_c^*}$.

- If $\mathrm{bal}(u) < 0$, we denote by $\tilde{u}$ the word $u_{n-1} \cdots u_0$. By exchanging the roles played by call and return symbols, we have $\mathrm{bal}(\tilde{u}) > 0$ and thus either $\tilde{u}$ is conjugate to a word in A_r^+ or $\tilde{u}$ is conjugate to a word in $(\tilde{P} A_r^*)^+$, where $\tilde{P} = \{\tilde{c} \mid c \in \mathrm{Prime}(X)\}$. We thus get that u is conjugate to a word in $(A_r^* \,\mathrm{Prime}(X))^+$ and x belongs to $\mathsf{X}_{M_r^* C}$.

□

As a consequence, we obtain the following expression of the zeta function of a sofic-Dyck shift.

Proposition 12. *Let X be a sofic-Dyck shift presented by a Dyck automaton $\mathcal{A}$ and C, M_r, M_c defined as above from $\mathcal{A}$. The zeta function of X is*

$$\zeta_X(z) = \frac{\zeta_{\mathsf{X}_{CM_c{}^*}}(z)\,\zeta_{\mathsf{X}_{M_r{}^* C}}(z)\,\zeta_{\mathsf{X}_{M_c}}(z)\,\zeta_{\mathsf{X}_{M_r}}(z)}{\zeta_{\mathsf{X}_C}(z)} \tag{1}$$

Proof. The formula is a direct consequence of Proposition 11 and of the definition of the zeta function. □

We recall below the notion of circular codes (see for instance [11]). We say that a subset S of nonempty words over A^* is *a circular code* if for all $n, m \geq 1$ and $x_1, x_2, \ldots, x_n \in S$, $y_1, y_2, \ldots, y_m \in S$ and $p \in A^*$ and $s \in A^+$, the equalities

$$s x_2 x_3 \cdots x_n p = y_1 y_2 \cdots y_m, \tag{2}$$
$$x_1 = ps \tag{3}$$

implies

$$n = m \qquad p = \varepsilon \qquad \text{and} \qquad x_i = y_i \qquad (1 \leq i \leq n).$$

Proposition 13. *Let A be a tri-partitioned alphabet. The sets* $\mathrm{Prime}(A)$ *and* $\mathrm{Prime}(A) A_c{}^*$ *are circular codes.*

Proof. We prove that $\mathrm{Prime}(A) A_c{}^*$ is circular. This implies that its subset $\mathrm{Prime}(A)$ is circular.

Let us suppose that Equations 2 and 3 imply $n = m$ and $x_i = y_i$ for $n+m < N$. Assume that Equations 2 and 3 hold for some n, m with $n + m = N$.

Let us assume that $s \neq x_1$. Since s is a prefix of some $y_1 y_2 \cdots y_j$ and a suffix of x_1, we have $s \in \mathrm{Prime}(A)$ or $s \in {A_c}^+$. As $ps \in \mathrm{Prime}(A){A_c}^*$, we get that $p \in \mathrm{Prime}(A)$ and $s \in {A_c}^+$. It implies that the balance of each nonempty prefix of $sx_2x_3 \cdots x_n p$ is positive, in contradiction with y_1 prefix of $sx_2x_3 \cdots x_n p$. Hence $s = x_1$ and $p = \varepsilon$. If $y_1 \neq x_1$, one of these two words is a prefix of the other. Let us assume that $x_1 = y_1 z$ with $z \in A_c^*$. Then $zx_2x_3 \cdots x_n$ is positive, a contradiction with that fact that it has y_2 as prefix. Thus $x_1 = y_1$. By iteration of this process, we get $n = m$ and $x_i = y_i$. □

The notion of circular matrix below extends the classical notion of circular codes. We say that the matrix $(H_{pq})_{p,q \in Q}$, where each H_{pq} is a set of nonempty words over A is *circular* if for all $n, m \geq 1$ and $x_1 \in H_{p_0,p_1}, x_2 \in H_{p_1,p_2}, \ldots, x_n \in H_{p_{n-1}p_0}$, $y_1 \in H_{q_0,q_1}, y_2 \in H_{q_1,q_2}, \ldots, y_m \in H_{q_{m-1}q_0}$ and $p \in A^*$ and $s \in A^+$, the equalities

$$sx_2x_3 \cdots x_n p = y_1 y_2 \cdots y_m, \tag{4}$$

$$x_1 = ps \tag{5}$$

implies

$$n = m \qquad p = \varepsilon \qquad \text{and} \qquad x_i = y_i \qquad (1 \leq i \leq n).$$

Proposition 14. *Let $\mathcal{A}$ be a Dyck automaton. The matrices C, M_c and $C{M_c}^*$ defined from $\mathcal{A}$ are circular matrices.*

Proof. It is a direct consequence of the fact that $\mathrm{Prime}(A)$, A and $\mathrm{Prime}(A)A_c^*$ are circular codes. □

We say that $\mathcal{A}$ is *left reduced* (resp. *right reduced*) it is the left (resp. right) reduction of some Dyck automaton.

We say that $\mathcal{A}$ is *H-deterministic* if and only if for any two (admissible) H-paths sharing the same start and label are equal.

Proposition 15. *If $\mathcal{A}$ is left reduced, it is H-deterministic when H is M_c, C or $C{M_c}^*$.*

Proof. The Dyck automaton $\mathcal{A}$ is M_c-deterministic by construction. It is C-deterministic by Proposition 2. □

One proves similarly that

Proposition 16. *If $\mathcal{A}$ is right reduced, it is H-codeterministic for H is M_r, C or ${M_r}^*C$.*

In order to count periodic sequences of sofic-Dyck shifts, we need some machinery similar to the one used to count the periodic sequences of sofic shifts (see for instance [33]).

Let $\mathcal{A}$ be a Dyck automaton over A where $\mathcal{A} = (\mathcal{G}, M)$ with $\mathcal{G} = (Q, E)$. Let ℓ be a positive integer. We fix an ordering on the states Q. We define the Dyck automaton $\mathcal{A}_{\otimes\ell} = (\mathcal{G}_{\otimes\ell}, M_{\otimes\ell})$ over a new alphabet A' where $\mathcal{G}_{\otimes\ell} = (Q_{\otimes\ell}, E_{\otimes\ell})$ as follows.

- We set $A' = (A'_c, A'_r, A'_i)$ with $A'_c = A_c \cup \{-a \mid a \in A_c\}$, $A'_r = A_r \cup \{-a \mid a \in A_r\}$, and $A'_i = A_i \cup \{-a \mid a \in A_i\}$.
- We denote by $Q_{\otimes\ell}$ the set of ordered ℓ-uples of distinct states of Q.
- Let $P = (p_1, \dots, p_\ell)$, $R = (r_1, \dots, r_\ell)$, be two elements of $Q_{\otimes\ell}$. Thus $p_1 < \cdots < p_\ell$ and $r_1 < \cdots < r_\ell$. There is an edge labeled by a from P to R in $\mathcal{A}_{\otimes\ell}$ if and only if there are edges labeled by a from p_i to p'_i for $1 \le i \le \ell$ and R is an even permutation of $(p'_1, \dots, p'_\ell)$. If the permutation is odd we assign the label $-a$. Otherwise, there is no edge with label a or $-a$ from P to R.
- We define $M_{\otimes\ell}$ as the set of pairs of edges $((p_1, \dots, p_\ell), a, (p'_1, \dots, p'_\ell))$, and $((r_1, \dots, r_\ell), \pm b, (r'_1, \dots, r'_\ell))$ of $\mathcal{A}_{\otimes\ell}$ such that each edge (p_i, a, p'_i) is matched with (r_i, b, r'_i) for $1 \le i \le \ell$.

We say that a path of $\mathcal{A}_{\otimes\ell}$ is *admissible* if it is admissible when the signs of the labels are omitted, the sign of the label of a path being the product of the signs of the labels of the edges of the path.

We denote by $C_{\otimes\ell,PP'}$ the set of signed prime Dyck words c labeling an admissible path in $\mathcal{A}_{\otimes\ell}$ from P to P'. We denote by $C_{\otimes\ell}$ the matrix $(C_{\otimes\ell,PP'})_{P,P' \in Q_{\otimes\ell}}$ whose coefficients are sums of signed words of A^+. More generally, if H denotes one of the matrices $C, CM_c{}^*, M_c, M_r{}^*C, M_r$ defined from $\mathcal{A}$, we denote by $H_{\otimes\ell}$ the matrix defined from $\mathcal{A}_{\otimes\ell}$ similarly.

6.2. Computation of the zeta function of X_H

Denote $\mathbb{Z}\langle\langle A\rangle\rangle$ the set of noncommutative formal power series over the alphabet A with coefficients in $\mathbb{Z}$. Let $\mathbb{Z}[[A]]$ be the usual commutative algebra of formal power series in the variables a in A and $\pi: \mathbb{Z}\langle\langle A\rangle\rangle \to \mathbb{Z}[[A]]$ be the natural homomorphism. Let S be a commutative or noncommutative series. One can write $S = \sum_{n\ge 0}[S]_n$ where each $[S]_n$ is the homogeneous part of S of degree n. We denote by $\theta: \mathbb{Z}[[A]] \to \mathbb{Z}[[z]]$ the homomorphism such that $\theta(a) = z$ for any letter $a \in A$. The homomorphism θ and π extends to matrices with coefficients in $\mathbb{Z}\langle\langle A\rangle\rangle$ and $\mathbb{Z}[[A]]$ respectively.

Proposition 17. *Let $\mathcal{A}$ be a left reduced Dyck automaton and H be one of the matrices $C, CM_c{}^*, M_c$ defined from $\mathcal{A}$. We have*

$$\pi P_n(\mathsf{X}_H) = \sum_{\ell=1}^{|Q|} (-1)^{\ell+1} \operatorname{trace} \sum_{1\le j\le n} j[\pi H_{\otimes\ell}]_j [(1-\pi H_{\otimes\ell})^{-1}]_{n-j}.$$

where $P_n(\mathsf{X}_H)$ is the set of periodic pattern of X_H of length n.

Proof. With a slight abuse of notations, we will say that a word u belongs to H if u is belongs to some H_{pq}.

Let x be a periodic sequence of X_H of period $n \ge 1$. We have $\sigma^n(x) = x$ if and only if x is a two-sided infinite concatenation of a word $w = vx_2 \cdots x_k u$

of length n with $x_i \in H$, $x_1 = uv \in H$, and $v \neq \varepsilon$. Let $j = |x_1|$. The sequences $x, \sigma(x), \dots, \sigma^{j-1}(x)$ are all distinct by circularity of the matrix H. Since $\pi(w) = uvx_2 \dots x_k = x_1 \dots x_k$, we get that

$$\pi P_n(\mathsf{X}_H) = \sum_{1 \le j \le n} \sum_{k \ge 1} j E_{n,j,k} = \sum_{1 \le j \le n} j E_{n,j},$$

where $E_{n,j,k}$ is the H-path labels of length n which are concatenation of k words $x_1 \dots x_k$ of H with $|x_1| = j$, and $E_{n,j}$ is the union of the $E_{n,j,k}$. The sets $E_{n,j,k}$ and $E_{n,j',k'}$ are disjoint for $k \neq k'$ or for $j \neq j'$.

Let us denote by $D_{n,j,k}$ and $D_{n,j}$ the matrices

$$D_{n,j,k} = [H]_j [H^{k-1}]_{n-j},$$
$$D_{n,j} = \sum_{k \ge 1} D_{n,j,k}.$$

Then $E_{n,j,k}$ (resp. $E_{n,j}$) is the set of labels of $D_{n,j,k}$-paths (resp. $D_{n,k}$-paths).

Let j be a fixed integer between 1 and n. Let us show that

$$\sum_{w \in E_{n,j}} w = \sum_{\ell=1}^{|Q|} (-1)^{\ell+1} \operatorname{trace}((D_{n,j})_{\otimes \ell}).$$

Note that w appears in $\operatorname{trace}((D_{n,j})_{\otimes \ell})$ for some integer ℓ with $1 \le \ell \le |Q|$ only if $w \in E_{n,j}$.

Thus we can write

$$\sum_{\ell=1}^{|Q|} (-1)^{\ell+1} \operatorname{trace}((D_{n,j})_{\otimes \ell}) = \sum_{w \in E_{n,j}} c(w)\, w, \tag{6}$$

where $c(w) \in \mathbb{Z}$.

We will show that $c(w) = 1$ for every word w such that $w \in E_{n,j}$.

Let w be such a word. Since $\mathcal{A}$ is H-deterministic, the coefficient of each word in H^k_{pq} for $k \ge 1$ and fixed states p, q, is at most one since there is at most one H-path in $\mathcal{A}$ going from p to q and labeling a given word. Hence the coefficient of w in each $(D_{n,j})_{pq}$ is at most one for fixed states p, q.

If $w \in E_{n,j}$ there must be at least one nonempty subset R of Q (of cardinal m) on which w acts as a permutation μ_w of R induced by $D_{n,j}$, *i.e.* such that the coefficient of w in $(D_{n,j})_{p\mu_w(p)}$ is one for each $p \in R$. If two subsets have this property, then does the union. Hence there is a largest subset $P \subseteq Q$ on which w acts as a permutation. At this point we need a combinatorial lemma used in [33, Lemma 6.4.9]. We recall its proof for the sake of completeness.

In the following lemma, the notation $\mu|_R$ means the restriction of a permutation μ to a set of states R and $\varepsilon(\mu)$ is the signature of the permutation μ.

Lemma 3. *[33, Lemma 6.4.9] Let μ be a permutation of a finite set P and let $\mathcal{P} = \{R \subseteq P \mid R \neq \varnothing,\ \mu(R) = R\}$. Then*

$$\sum_{R \in \mathcal{P}} (-1)^{|R|+1} \varepsilon(\mu|_R) = 1.$$

Proof of Lemma 3. Recall that P decomposes under μ into disjoint cycles, say $P_1, \cdots, P_d$. Thus each $\mu|_{P_i}$ is a cyclic permutation and so

$$\varepsilon(\mu|_{P_i}) = (-1)^{1+|P_i|}.$$

The nonempty sets $R \subseteq P$ for which $\mu(R) = R$ are exactly the nonempty unions of sub-collections of $\{P_1, \cdots, P_d\}$. Thus

$$\begin{aligned}\sum_{R \in \mathcal{P}} (-1)^{|R|+1} \varepsilon(\mu|_R) &= \sum_{\varnothing \neq K \subseteq \{1,\dots,d\}} (-1)^{1+|\cup_{k\in K} P_k|} \varepsilon(\mu|_{\cup_{k\in K} P_k}), \\ &= \sum_{\varnothing \neq K \subseteq \{1,\dots,d\}} (-1)^{1+\sum_{k\in K}|P_k|} \prod_{k \in K} (-1)^{1+|P_k|}, \\ &= \sum_{\varnothing \neq K \subseteq \{1,\dots,d\}} (-1)^{|K|+1+2\sum_{k\in K}|P_k|}, \\ &= \sum_{i=1}^{d} (-1)^{i+1} \binom{d}{i} = 1 - (1-1)^d = 1.\end{aligned}$$

□

Returning to the computation of the coefficient $c(w)$ in Equation 6, let P be the largest subset of Q on which w acts as a permutation. The coefficient $c(w)$ is by definition of $(D_{n,j})_{\otimes \ell}$,

$$c(w) = \sum_{R \in \mathcal{P}} (-1)^{|R|+1} \varepsilon(\mu_w|_R) = 1.$$

Hence

$$\sum_{\ell=1}^{|Q|} (-1)^{\ell+1} \operatorname{trace}((D_{n,j})_{\otimes \ell}) = \sum_{w \in E_{n,j}} w,. \tag{7}$$

We get

$$\begin{aligned}\pi \mathcal{P}_n(\mathsf{X}_H) &= \sum_{j=1}^{n} j E_{n,j}, \\ &= \sum_{\ell=1}^{|Q|} (-1)^{\ell+1} \sum_{j=1}^{n} j \operatorname{trace}(\pi(D_{n,j})_{\otimes \ell}), \\ &= \sum_{\ell=1}^{|Q|} (-1)^{\ell+1} \operatorname{trace} \sum_{j=1}^{n} j \sum_{k \geq 0} \pi([H]_j [H^k]_{n-j})_{\otimes \ell}, \\ &= \sum_{\ell=1}^{|Q|} (-1)^{\ell+1} \operatorname{trace} \sum_{j=1}^{n} j \sum_{k \geq 0} ([\pi H_{\otimes \ell}]_j [\pi H_{\otimes \ell}^k]_{n-j}), \\ &= \sum_{\ell=1}^{|Q|} (-1)^{\ell+1} \operatorname{trace} \sum_{j=1}^{n} j [\pi H_{\otimes \ell}]_j [(1 - \pi H_{\otimes \ell})^{-1}]_{n-j}.\end{aligned}$$

□

Proposition 18. *Let $\mathcal{A}$ be a left reduced Dyck automaton. The zeta function of X_H, where H is one of the matrices $C, CM_c{}^*, M_c$ defined from $\mathcal{A}$, is*

$$\zeta_{\mathsf{X}_H}(z) = \prod_{\ell=1}^{|Q|} \det(I - H_{\otimes\ell}(z))^{(-1)^\ell},$$

where $H_{\otimes\ell}(z) = \theta\pi H_{\otimes\ell}$.

The same formula holds for X_H when H is equal to C, $M_r{}^*C$ or M_r when $\mathcal{A}$ be a right reduced.

Proof. We get from Proposition 17

$$\begin{aligned}
&\sum_{n\geq 1} \frac{\theta\pi P_n(\mathsf{X}_H)}{n} \\
&= \sum_{n\geq 1} \frac{1}{n} \sum_{\ell=1}^{|Q|} (-1)^{\ell+1} \operatorname{trace} \sum_{j=1}^{n} j[\theta\pi H_{\otimes\ell}]_j [(I - \theta\pi H_{\otimes\ell})^{-1}]_{n-j}, \\
&= \sum_{\ell=1}^{|Q|} (-1)^{\ell+1} \sum_{n\geq 1} \frac{1}{n} \operatorname{trace} \sum_{j=0}^{n-1} (j+1)[\theta\pi H_{\otimes\ell}]_{j+1} [(I - \theta\pi H_{\otimes\ell})^{-1}]_{n-j-1}, \\
&= \sum_{\ell=1}^{|Q|} (-1)^{\ell+1} \sum_{n\geq 1} \frac{1}{n} \operatorname{trace} \sum_{j=0}^{n-1} [\mathrm{d}\theta\pi H_{\otimes\ell}]_j [(I - \theta\pi H_{\otimes\ell})^{-1}]_{n-j-1}, \\
&= \sum_{\ell=1}^{|Q|} (-1)^{\ell+1} \sum_{n\geq 1} \frac{1}{n} \operatorname{trace}[(\mathrm{d}\theta\pi H_{\otimes\ell})(I - \theta\pi H_{\otimes\ell})^{-1}]_{n-1}, \\
&= \sum_{\ell=1}^{|Q|} (-1)^{\ell+1} \sum_{n\geq 1} \frac{1}{n} \operatorname{trace}[(\mathrm{d}\log(I - \theta\pi H_{\otimes\ell})]_{n-1}, \\
&= \sum_{\ell=1}^{|Q|} (-1)^{\ell+1} \sum_{n\geq 1} \operatorname{trace}[-\log(I - \theta\pi H_{\otimes\ell})]_n, \\
&= \sum_{\ell=1}^{|Q|} (-1)^{\ell+1} \operatorname{trace}(-\log(I - \theta\pi H_{\otimes\ell})),
\end{aligned}$$

where d denotes the derivative with respect to the variable z.

Thus, using Jacobi's formula, we obtain

$$\begin{aligned}
\zeta(\mathsf{X}_H)(z) &= \exp\operatorname{trace}(\sum_{\ell=1}^{|Q|} (-1)^{\ell+1}(-\log(I - \theta\pi H_{\otimes\ell}))), \\
&= \det\exp(\sum_{\ell=1}^{|Q|} (-1)^{\ell} \log(I - \theta\pi H_{\otimes\ell}))), \\
&= \prod_{\ell=1}^{|Q|} \det(I - \theta\pi H_{\otimes\ell})^{(-1)^\ell}.
\end{aligned}$$

□

6.3. Computation of the zeta function of X

The previous computations allow us to obtain directly the following general formula for the zeta function of a sofic-Dyck shift X.

Theorem 3. *The zeta function of a sofic-Dyck shift accepted by a left reduced Dyck (resp. right reduced) Dyck automaton $\mathcal{A}$ (resp. $\mathcal{B}$) is given by the following formula, where C, M_c and $CM_c{}^*$ are defined from $\mathcal{A}$ and M_r and $M_r{}^*C$ are defined from $\mathcal{B}$.*

$$\zeta_X(z) = \frac{\zeta_{\mathsf{X}_{CM_c{}^*}}(z)\zeta_{\mathsf{X}_{M_r{}^*C}}(z)\zeta_{\mathsf{X}_{M_c}}(z)\zeta_{\mathsf{X}_{M_r}}(z)}{\zeta_{\mathsf{X}_C}(z)}$$
$$= \prod_{\ell=1}^{|Q|} \det(I - (CM_c{}^*)_{\otimes\ell}(z))^{(-1)^\ell} \det(I - (M_r{}^*C)_{\otimes\ell}(z))^{(-1)^\ell}$$
$$\det(I - C_{\otimes\ell}(z))^{(-1)^\ell+1} \det(I - M_{r,\otimes\ell}(z))^{(-1)^\ell} \det(I - M_{c,\otimes\ell}(z))^{(-1)^\ell}.$$

Corollary 3. *The zeta function of a sofic-Dyck shift is $\mathbb{Z}$-algebraic.*

Example 4. Let $\mathcal{A}$ be the Dyck automaton over A pictured on the left part of Figure 4, where $A = (\{a, a'\}, \{b, b'\}, \{i\})$. The Dyck automaton $\mathcal{A}_{\otimes 1}$ is the same as $\mathcal{A}$. The Dyck automaton $\mathcal{A}_{\otimes 2}$ is pictured on the right part of Figure 4. Let

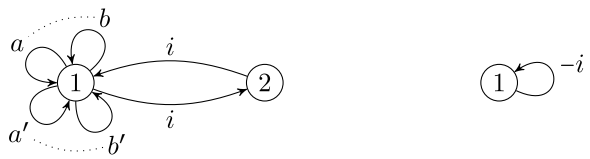

Figure 4: A Dyck automaton $\mathcal{A}$ (on the left) over $A = A_c \sqcup A_r \sqcup A_i$ with $A_c = \{a, a'\}$, $A_r = \{b, b'\}$ and $A_i = \{i\}$ and the Dyck automaton $\mathcal{A}_{\otimes 2}$ (on the right). Matched edges are linked with a dotted line.

us compute the zeta function of $\mathsf{X}_\mathcal{C}$ for this automaton. Let

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}, \; C_{\otimes 2} = \begin{bmatrix} C_{(1,2),(1,2)} \end{bmatrix}.$$

We have $C_{11} = aD_{11}b + a'D_{11}b'$, $C_{22} = 0$, $C_{12} = i$, $C_{21} = i$, with $D_{11} = aD_{11}bD_{11} + a'D_{11}b'D_{11} + iiD_{11} + \varepsilon$. Hence

$$2z^2 D_{11}^2(z) - (1 - z^2)D_{11}(z) + 1 = 0$$

Since the coefficient of z^0 in $D_{11}(z)$ is 1, we get

$$D_{11}(z) = \frac{1 - z^2 - \sqrt{1 - 10z^2 + z^4}}{4z^2}.$$

Hence

$$C_{11}(z) = 2z^2 D_{11}(z) = \frac{1 - z^2 - \sqrt{1 - 10z^2 + z^4}}{2}.$$

We have $C_{22}(z) = 0$, $C_{12}(z) = C_{21}(z) = z$. We also have $C_{(1,2),(1,2)} = -i$ and thus $C_{(1,2),(1,2)}(z) = -z$. Thus

$$\begin{aligned}
\zeta_{X_C}(z) &= \prod_{\ell=1}^{2} \det(I - C_{\otimes \ell}(z))^{(-1)^\ell} \\
&= (1+z) \begin{vmatrix} 1 - \frac{1-z^2-\sqrt{1-10z^2+z^4}}{2} & -z \\ -z & 1 \end{vmatrix}^{-1}, \\
&= \frac{1+z}{1 - z^2 - \frac{1-z^2-\sqrt{1-10z^2+z^4}}{2}}.
\end{aligned}$$

For $H = M_c, M_r$, we have

$$\prod_{\ell=1}^{2} \det(I - H_{\otimes \ell}(z))^{(-1)^\ell} = \frac{1}{1-2z}.$$

We also have

$$\begin{aligned}
C{M_c}^* &= \begin{bmatrix} C_{11} & i \\ i & 0 \end{bmatrix} \begin{bmatrix} \{a,a'\}^* & 0 \\ 0 & \varepsilon \end{bmatrix} = \begin{bmatrix} C_{11}\{a,a'\}^* & i \\ i\{a,a'\}^* & 0 \end{bmatrix}, \\
{M_r}^* C &= \begin{bmatrix} \{b,b'\}^* & 0 \\ 0 & \varepsilon \end{bmatrix} \begin{bmatrix} C_{11} & i \\ i & 0 \end{bmatrix} = \begin{bmatrix} \{b,b'\}^* C_{11} & \{b,b'\}^* i \\ i & 0 \end{bmatrix}.
\end{aligned}$$

$$\begin{aligned}
\prod_{\ell=1}^{2} \det(I - (C{M_c}^*)_{\otimes \ell}(z))^{(-1)^\ell} &= (1+z) \begin{vmatrix} 1 - \frac{C_{11}(z)}{(1-2z)} & -z \\ -\frac{z}{1-2z} & 1 \end{vmatrix}^{-1} \\
&= \frac{(1+z)(1-2z)}{1 - 2z - z^2 - C_{11}(z)}.
\end{aligned}$$

The same equality holds for ${M_r}^* C$. We finally get

$$\begin{aligned}
\zeta_X(z) &= \frac{(1+z)(1 - z^2 - C_{11}(z))}{(1 - 2z - z^2 - C_{11}(z))^2}, \\
&= \frac{(1+z)(1 - z^2 - \frac{1-z^2-\sqrt{1-10z^2+z^4}}{2})}{(1 - 2z - z^2 - \frac{1-z^2-\sqrt{1-10z^2+z^4}}{2})^2}.
\end{aligned}$$

The above formula shows that the zeta function of a sofic-Dyck shift is a $\mathbb{Z}$-algebraic series. It is proved in [8] that the zeta function of a finite-type-Dyck shifts is the generating series of an unambiguous context-free language, *i.e.* is an $\mathbb{N}$-algebraic function. We conjecture that the result also holds for sofic-Dyck shifts.

There is no known criterion for a $\mathbb{Z}$-algebraic series with coefficients in $\mathbb{N}$ to be $\mathbb{N}$-algebraic but there are some necessary conditions on the asymptotic behavior of the coefficients (see the Drmota-Lalley-Woods Theorem in [17, VII.6.1] and recent insights from Banderier and Drmota in [3, 4]).

Acknowledgements The authors would like to thank Arnaud Carayol for pointing us some mistakes in a preliminary version of this paper and Pavel Heller and Wolfgang Krieger for helpful comments.

7. References

[1] R. Alur and P. Madhusudan. Visibly pushdown languages. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 202–211 (electronic), New York, 2004. ACM.

[2] R. Alur and P. Madhusudan. Adding nesting structure to words. *J. ACM*, 56(3), 2009.

[3] C. Banderier and M. Drmota. Coefficients of algebraic functions: formulae and asymptotics. In *DMTCS Proc. AS, 25th International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC 2013)*, pages 1065–1076, 2013.

[4] C. Banderier and M. Drmota. Formulae and asymptotics for coefficients of algebraic functions. *Combin. Probab. Comput.*, 24(1):1–53, 2015.

[5] M.-P. Béal. *Codage symbolique.* Masson, 1993.

[6] M.-P. Béal. Puissance extérieure d'un automate déterministe, application au calcul de la fonction zêta d'un système sofique. *RAIRO Inform. Théor. Appl.*, 29, 1995.

[7] M.-P. Béal, M. Blockelet, and C. Dima. Sofic-Dyck shifts. In *39th International Symposium on Mathematical Foundations of Computer Science MFCS 2014, (MFCS 2014)*, Lecture Notes in Computer Science. Springer-Verlag, 2014.

[8] M.-P. Béal, M. Blockelet, and C. Dima. Zeta functions of finite-type-Dyck shifts are $\mathbb{N}$-algebraic. In *2014 Information Theory and Applications Workshop, ITA 2014*. IEEE, 2014.

[9] J. Berstel and L. Boasson. Balanced grammars and their languages. In *Formal and Natural Computing*, volume 2300 of *Lecture Notes in Computer Science*, pages 3–25. Springer, 2002.

[10] J. Berstel and L. Boasson. Formal properties of XML grammars and languages. *Acta Inf.*, 38(9):649–671, 2002.

[11] J. Berstel, D. Perrin, and C. Reutenauer. *Codes and Automata*, volume 129 of *Encyclopedia of Mathematics and its Applications.* Cambridge University Press, Cambridge, 2010.

[12] J. Berstel and C. Reutenauer. Zeta functions of formal languages. *Trans. Amer. Math. Soc.*, 321:533–546, 1990.

[13] R. Bowen. Symbolic dynamics. In *On axiom A diffeomorphism*, number 35 in CBMS Reg. Conf. American Mathematical Society, 1978.

[14] R. Bowen and O. Lanford. Zeta functions of restrictions of the shift transformation. In *Proc. Sympos. Pure Math.*, volume 14, pages 43–50. American Mathematical Society, 1970.

[15] A. Costa and B. Steinberg. A categorical invariant of flow equivalence of shifts. *CoRR*, 2013. abs/1304.3487.

[16] K. Culik, II and S. Yu. Cellular automata, $\omega\omega$-regular sets, and sofic systems. *Discrete Appl. Math.*, 32(2):85–101, 1991.

[17] P. Flajolet and R. Sedgewick. *Analytic Combinatorics*. Cambridge University Press, Cambridge, 2009.

[18] S. Ginsburg and M. A. Harrison. Bracketed context-free languages. *J. Comput. Syst. Sci.*, 1(1):1–23, 1967.

[19] T. Hamachi and W. Krieger. A construction of subshifts and a class of semigroups. *CoRR*, 2013. abs/1303.4158.

[20] T. Hamachi and W. Krieger. On certain subshifts and their associated monoids. *CoRR*, 2013. abs/1202.5207.

[21] K. Inoue. The zeta function, periodic points and entropies of the Motzkin shift. *CoRR*, 2006. math/0602100.

[22] K. Inoue and W. Krieger. Subshifts from sofic shifts and Dyck shifts, zeta functions and topological entropy. *CoRR*, 2010. abs/1001.1839.

[23] G. Keller. Circular codes, loop counting, and zeta-functions. *J. Combin. Theory Ser. A*, 56(1):75–83, 1991.

[24] Y.-O. Kim, J. Lee, and K. K. Park. A zeta function for flip systems. *Pacific J. Math.*, 209(2):289–301, 2003.

[25] Y.-O. KIM and S. RYU. On the number of fixed points of a sofic shift-flip system. *Ergodic Theory and Dynamical Systems*, 35:482–498, 4 2015.

[26] B. P. Kitchens. *Symbolic Dynamics*. Universitext. Springer-Verlag, Berlin, 1998. One-sided, two-sided and countable state Markov shifts.

[27] D. E. Knuth. A characterization of parenthesis languages. *Information and Control*, 11(3):269–289, 1967.

[28] W. Krieger. On the uniqueness of the equilibrium state. *Math. Systems Theory*, 8(2):97–104, 1974/75.

[29] W. Krieger. On g-functions for subshifts. In *Dynamics & stochastics*, volume 48 of *IMS Lecture Notes Monogr. Ser.*, pages 306–316. Inst. Math. Statist., Beachwood, OH, 2006.

[30] W. Krieger. On subshift presentations. *CoRR*, 2012. abs/1209.2578.

[31] W. Krieger and K. Matsumoto. Zeta functions and topological entropy of the Markov-Dyck shifts. *Münster J. Math.*, 4:171–183, 2011.

[32] W. Krieger and K. Matsumoto. A notion of synchronization of symbolic dynamics and a class of C^*-algebras. *Acta Appl. Math.*, 126:263–275, 2013.

[33] D. Lind and B. Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, Cambridge, 1995.

[34] A. Manning. Axiom A diffeomorphisms have rational zeta functions. *Bull. London Math. Soc.*, 3:215–220, 1971.

[35] K. Matsumoto. C^*-algebras arising from Dyck systems of topological Markov chains. *Math. Scand.*, 109(1):31–54, 2011.

[36] K. Matsumoto. A certain synchronizing property of subshifts and flow equivalence. *Israel J. Math.*, 196(1):235–272, 2013.

[37] R. McNaughton. Parenthesis grammars. *J. ACM*, 14(3):490–500, 1967.

[38] K. Mehlhorn. Pebbling moutain ranges and its application of dcfl-recognition. In *Automata, Languages and Programming, 7th Colloquium, Noordweijkerhout, The Netherland, July 14-18, 1980, Proceedings*, volume 85 of *Lecture Notes in Computer Science*, pages 422–435. Springer, 1980.

[39] C. Reutenauer. $\mathbb{N}$-rationality of zeta functions. *Adv. in Appl. Math.*, 18(1):1–17, 1997.