Sofic-Dyck shifts

Marie-Pierre Béal, Michel Blockelet
and Cătălin Dima

Université Paris-Est
Laboratoire d'informatique Gaspard-Monge UMR 8049
Laboratoire d'algorithmique, complexité et logique

EQINOCS Meeting, January 2014

Overview

- Sofic-Dyck shifts : definition and characterization
- Zeta function
- Finite-type Dyck shifts, edge-Dyck shifts
- Decomposition theorem of edge-Dyck shifts
- Future work

Shifts of sequences of symbols over a finite alphabet

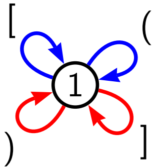
Definition

A subshift of sequences over A is the set of bi-infinite sequences X_F of symbols in A avoiding a given set F of finite blocks.

$A = \{a, b\}$, $F = \{aa\}$

$$\cdots ababbbabab \cdot abababbbbabbaba \cdots$$

Examples



The Dyck shift

Matched edges

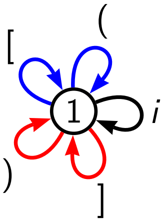
$1 \xrightarrow{(} 1$ is matched with $1 \xrightarrow{)} 1$

$1 \xrightarrow{[} 1$ is matched with $1 \xrightarrow{]} 1$

Allowed sequence : $\cdots [\,(\,(\,)\,)\,]\,[\,[\,(\cdots$

Forbidden blocks : $(\,]$, $[\,)$, $(\,(\,)\,]$, $\cdots$

Examples



The Motzkin shift

Matched edges

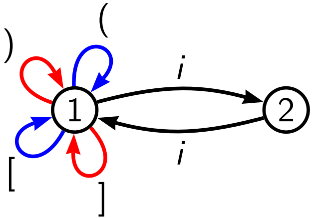
$1 \xrightarrow{(} 1$ is matched with $1 \xrightarrow{)} 1$

$1 \xrightarrow{[} 1$ is matched with $1 \xrightarrow{]} 1$

Allowed sequence : $\cdots [\, i\; i (\,(\,)\,)\,]\; i\, [\, i\; i\; i\,]\,[\,(\cdots$

Forbidden blocks : $(\,]$, $(\; i\; i\,]$, $\cdots$

Examples



The even-Motzkin shift

Matched edges

$1 \xrightarrow{(} 1$ is matched with $1 \xrightarrow{)} 1$

$1 \xrightarrow{[} 1$ is matched with $1 \xrightarrow{]} 1$

Allowed sequence :$\cdots i\,(\,[\,i\;i\,]\,i\;i)\,(\,(\cdots$

Forbidden sequence : $\cdots i\,(\,[\,i\,]\,i\;i\;)\;\cdots$

Sofic-Dyck shifts

Shifts of sequences over a *pushdown alphabet* A which is the disjoint union of (A_c, A_r, A_i) :
A_c is the set of call alphabet
A_r is the set of return alphabet
A_i is the set of internal alphabet

A *Dyck automaton* $(\mathcal{A}, M)$ over A is a directed labelled graph $\mathcal{A} = (Q, E, A)$ where $E \subset Q \times A \times Q$
M is the set of *matched edges* : a set of pairs $((p, a, q), (r, b, s))$ of edges of $\mathcal{A}$ with $a \in A_c$ and $b \in A_r$

equipped with a *graph semigroup* S generated by the set $E \cup \{x_{pq} \mid p, q \in Q\} \cup \{0\}$ with

Sofic-Dyck shifts

$E \cup \{x_{pq} \mid p, q \in Q\} \cup \{0\}$

$$
\begin{aligned}
0s = s0 &= 0 && \text{for } s \in S, \\
x_{pq}x_{qr} &= x_{pr} && \text{for } p, q, r \in Q, \\
x_{pq}x_{rs} &= 0 && \text{for } p, q, r, s \in Q, q \neq r, \\
(p, \ell, q) &= x_{pq} && \text{for } p, q, \in Q, \ell \in A_i, \\
(p, \color{blue}{a}\color{black}, q)x_{qr}(r, \color{red}{b}\color{black}, s) &= x_{ps} && \text{for } ((p, \color{blue}{a}\color{black}, q), (r, \color{red}{b}\color{black}, s)) \in M, \\
(p, \color{blue}{a}\color{black}, q)x_{qr}(r, \color{red}{b}\color{black}, s) &= 0 && \text{for } ((p, \color{blue}{a}\color{black}, q), (r, \color{red}{b}\color{black}, s)) \notin M, \\
(p, a, q)(r, b, s) &= 0, && \text{for } p, q, r, s \in Q, q \neq r, a, b \in A, \\
x_{pp}(p, a, q) = (p, a, q) &= (p, a, q)x_{qq} && \text{for } p, q \in Q, a \in A, \\
x_{pq}(r, a, s) = 0 &= (r, a, s)x_{tu} && \text{for } p, q \in Q, a \in A, \ q \neq r, \ s \neq t.
\end{aligned}
$$

Sofic-Dyck shifts

If π is a finite path, $f(\pi)$ is its image in the graph semigroup S
A finite path is *admissible* if $f(\pi) \neq 0$
A bi-infinite path is *admissible* if all its factors are admissible.
A word labeling an admissible path π such that $f(\pi) = x_{pq}$ is a *Dyck word* or a *well-matched word*.
A bi-infinite sequence is *accepted* by $(\mathcal{A}, M)$ if it is the label of a bi-infinite admissible path of $(\mathcal{A}, M)$.

A *sofic-Dyck shift* is a set of bi-infinite sequences accepted by a Dyck automaton.

Related classes of symbolic dynamical systems

Dyck shifts, *Krieger et al.*
Markov-Dyck shifts, *Krieger and Matsumoto*
Extensions of Markov-Dyck shifts, *Inoue and Krieger*
Shifts presented by $\mathcal{R}$-graphs, *Krieger*
Coded systems, *Blanchard and Hansel*

Visibly pushdown shifts

Proposition

The set of allowed blocks of a sofic-Dyck shift is a visibly pushdown language. Conversely, if L is a factorial extensible visibly pushdown language, then the shift of sequences whose factors belong to L is a sofic-Dyck shift.

It is not difficult to prove that the set of labels of finite admissible paths is a visibly pushdown language.

It is more complicate to prove that it holds also for the set of (allowed) blocks. Indeed, labels of finite admissible paths may not be blocks. Culik and Yu showed that the subset of bi-extensible words of a context-free language may not be context-free. It is true for factorial languages. We adapt the construction for the visibly pushdown case.

Visibly pushdown automaton

$M = (Q, I, \Gamma, \Delta, F)$

- Q is the finite state of states
- $A = (A_c, A_r, A_i)$ is the partitioned alphabet
- Γ is the stack alphabet
- $\Delta \subset \begin{cases} Q \times A_c \times Q \times (\Gamma \setminus \{\bot\}) \\ Q \times A_r \times (\Gamma \setminus \{\bot\}) \times Q \\ Q \times A_i \times Q \end{cases}$

$(p, \ell, q) \in \Delta \qquad p, \begin{array}{|c|} \\ \alpha \\ \vdots \\ \beta \\ \bot \\ \hline \end{array}$

Visibly pushdown automaton

$M = (Q, I, \Gamma, \Delta, F)$

- Q is the finite state of states
- $A = (A_c, A_r, A_i)$ is the partitioned alphabet
- Γ is the stack alphabet
- $\Delta \subset \begin{cases} Q \times A_c \times Q \times (\Gamma \setminus \{\perp\}) \\ Q \times A_r \times (\Gamma \setminus \{\perp\}) \times Q \\ Q \times A_i \times Q \end{cases}$

$(p, \ell, q) \in \Delta \qquad p, \begin{array}{|c|} \alpha \\ \vdots \\ \beta \\ \perp \\ \hline \end{array} \xrightarrow{\ell} q, \begin{array}{|c|} \alpha \\ \vdots \\ \beta \\ \perp \\ \hline \end{array}$

Visibly pushdown automaton

$M = (Q, I, \Gamma, \Delta, F)$

- Q is the finite state of states
- $A = (A_c, A_r, A_i)$ is the partitioned alphabet
- Γ is the stack alphabet
- $\Delta \subset \begin{cases} Q \times A_c \times Q \times (\Gamma \setminus \{\perp\}) \\ Q \times A_r \times (\Gamma \setminus \{\perp\}) \times Q \\ Q \times A_i \times Q \end{cases}$

$(p, a, q, \alpha) \in \Delta \qquad p, \begin{array}{|c|} \alpha \\ \vdots \\ \beta \\ \perp \\ \hline \end{array}$

Visibly pushdown automaton

$M = (Q, I, \Gamma, \Delta, F)$

- Q is the finite state of states
- $A = (A_c, A_r, A_i)$ is the partitioned alphabet
- Γ is the stack alphabet
- $\Delta \subset \begin{cases} Q \times A_c \times Q \times (\Gamma \setminus \{\perp\}) \\ Q \times A_r \times (\Gamma \setminus \{\perp\}) \times Q \\ Q \times A_i \times Q \end{cases}$

$(p, a, q, \alpha) \in \Delta$

$$p, \left|\begin{array}{c} \\ \alpha \\ \vdots \\ \beta \\ \perp \end{array}\right| \xrightarrow{a} q, \left|\begin{array}{c} \alpha \\ \alpha \\ \vdots \\ \beta \\ \perp \end{array}\right|$$

Visibly pushdown automaton

$M = (Q, I, \Gamma, \Delta, F)$

- Q is the finite state of states
- $A = (A_c, A_r, A_i)$ is the partitioned alphabet
- Γ is the stack alphabet
- $\Delta \subset \begin{cases} Q \times A_c \times Q \times (\Gamma \setminus \{\perp\}) \\ Q \times A_r \times (\Gamma \setminus \{\perp\}) \times Q \\ Q \times A_i \times Q \end{cases}$

$(p, b, \alpha, q) \in \Delta$ $\qquad p, \begin{array}{|c|} \alpha \\ \vdots \\ \beta \\ \perp \\ \hline \end{array}$

Visibly pushdown automaton

$M = (Q, I, \Gamma, \Delta, F)$

- Q is the finite state of states
- $A = (A_c, A_r, A_i)$ is the partitioned alphabet
- Γ is the stack alphabet
- $\Delta \subset \begin{cases} Q \times A_c \times Q \times (\Gamma \setminus \{\perp\}) \\ Q \times A_r \times (\Gamma \setminus \{\perp\}) \times Q \\ Q \times A_i \times Q \end{cases}$

$(p, b, \alpha, q) \in \Delta$

$$p, \begin{array}{|c|} \\ \alpha \\ \vdots \\ \beta \\ \perp \\ \hline \end{array} \xrightarrow{b} q, \begin{array}{|c|} \\ \\ \vdots \\ \beta \\ \perp \\ \hline \end{array}$$

Visibly pushdown grammar

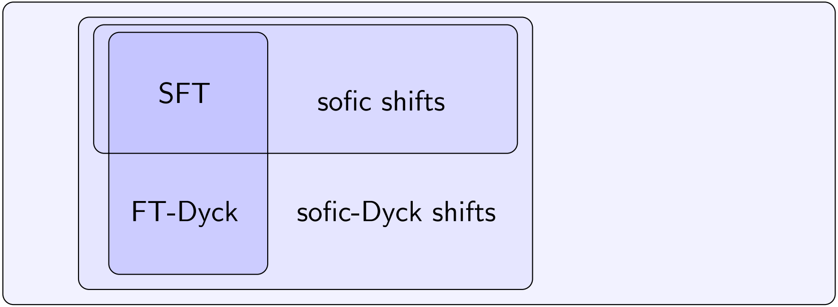
Visibly pushdown languages are generated by *visibly pushdown grammars* $G = (V, S, P)$ over A.
The set V of variables is partitioned into two disjoint sets V^0 and V^1.
V^0 derive only well-matched words
V^0 derive not well-matched words

- $X \to \varepsilon$;
- $X \to aY$, 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 $Z \in V^0$.

Finite-type-Dyck shifts

Finite-type-Dyck shift are accepted by *local* (or *definite*) Dyck automata.

We says that $(\mathcal{A}, M)$ is (m, a)-*local* if whenever two paths (or two admissible paths) $(p_i, a_i, p_{i+1})_{-m \le i \le a}$, $(q_i, a_i, q_{i+1})_{-m \le i \le a}$, of $\mathcal{A}$ of length $m + a$ have the same label, then $p_0 = q_0$.



Zeta function for sofic-Dyck shifts

Let X be a sofic-Dyck shift presented by a deterministic (or unambiguous) Dyck automaton.
Denoting by p_n the number of points of X of period n, the zeta function of X is defined as

$$\zeta_X(z) = \exp \sum_{n>0} \frac{p_n}{n} z^n.$$

Zeta function for
shifts of finite type, *Bowen, Lanford*
sofic shifts, *Manning, Bowen*
with $\mathbb{N}$-rationality, *Berstel and Reutenauer*
Dyck shifts, *Keller*
Motzkin shifts, *Inoue*
Markov-Dyck shifts, *Krieger and Matsumoto*
...

The computation combines technique from Bowen, Keller, Krieger and Matsumoto

Zeta function for sofic-Dyck shifts

Let $(\mathcal{A}, M)$ be a Dyck automaton.
$C = (C_{pq})$, where C_{pq} is the set of prime well-matched blocks labeling a path from p to q.
$M_c = (M_{c,pq})$, (resp. M_r) where $M_{c,pq}$ is the sum of call (resp. return) letters a labeling an edge from p to q (shifts Z_c and Z_r)
C_c (resp. C_r) is the matrix CM_c^* (resp. the matrix $M_r{}^* C$).

$\mathcal{A}_{\otimes\ell}$ is the labelled graph with states $Q_{\otimes\ell}$, the set of all subsets of Q having ℓ elements.
$P = (p_1, \ldots, p_\ell) \xrightarrow{a} P' = (p'_1, \ldots, p'_\ell)$,
if and only if there are edges labelled by a from p_i to q_i and $(q_1, \ldots, q_\ell)$ is an even permutation of P'.

Zeta function for sofic-Dyck shifts

Let $(\mathcal{A}, M)$ be a Dyck automaton.
$C = (C_{pq})$, where C_{pq} is the set of prime well-matched blocks labeling a path from p to q.
$M_c = (M_{c,pq})$, (resp. M_r) where $M_{c,pq}$ is the sum of call (resp. return) letters a labeling an edge from p to q (shifts Z_c and Z_r)
C_c (resp. C_r) is the matrix CM_c^* (resp. the matrix $M_r{}^*C$).

$\mathcal{A}_{\otimes\ell}$ is the labelled graph with states $Q_{\otimes\ell}$, the set of all subsets of Q having ℓ elements.
$P = (p_1, \ldots, p_\ell) \xrightarrow{-a} P' = (p'_1, \ldots, p'_\ell)$,
if and only if there are edges labelled by a from p_i to q_i and $(q_1, \ldots, q_\ell)$ is an odd permutation of P'.

Zeta function for sofic-Dyck shifts

Proposition

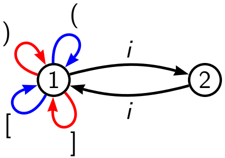
The zeta function of a sofic-Dyck shift accepted by a Dyck automaton $(\mathcal{A}, M)$ with matrices C, C_c, C_r, M_c, M_r is given by the following formula.

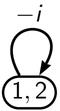
$$
\begin{aligned}
\zeta_X(z) &= \frac{\zeta_{X_{C_c}}(z)\zeta_{X_{C_r}}(z)\zeta_{Z_c}(z)\zeta_{Z_r}(z)}{\zeta_{X_C}(z)}, \\
&= \prod_{\ell=1}^{|Q|} \det(I - C_{c,\otimes\ell}(z))^{(-1)^\ell} \det(I - C_{r,\otimes\ell}(z))^{(-1)^\ell} \\
&\quad \det(I - M_{c,\otimes\ell}(z))^{(-1)^\ell} \det(I - M_{r,\otimes\ell}(z))^{(-1)^\ell} \det(I - C_{\otimes\ell}(z))^{(-1)^\ell+1}
\end{aligned}
$$

Example

Let X accepted by $(\mathcal{A}, M)$ over $A = (\{(,[\},\{),]\},\{i\})$

Matched edges : $(1 \xrightarrow{(} 1,\ 1 \xrightarrow{)} 1)$,$(1 \xrightarrow{[} 1,\ 1 \xrightarrow{]} 1)$.





$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}, = \begin{bmatrix} (\ D_{11}\ ) + [\ D_{11}\ ], & i \\ i & 0 \end{bmatrix}, C_{\otimes 2} = \begin{bmatrix} C_{(1,2),(1,2)} \end{bmatrix} = \begin{bmatrix} -i \end{bmatrix}$$

where $D_{11} = (\ D_{11}\ )\ D_{11} + [\ D_{11}\ ]\ D_{11} + i\ i\ D_{11} + \varepsilon$.

Example

$$C_c = C{M_c}^* = \begin{bmatrix} C_{11} & i \\ i & 0 \end{bmatrix} \begin{bmatrix} \{(,[\}^* & 0 \\ 0 & \varepsilon \end{bmatrix} = \begin{bmatrix} C_{11}\{(,[\}^* & i \\ i\{(,[\}^* & 0 \end{bmatrix},$$

$$C_r = {M_r}^* C = \begin{bmatrix} \{),]\}^* & 0 \\ 0 & \varepsilon \end{bmatrix} \begin{bmatrix} C_{11} & i \\ i & 0 \end{bmatrix} = \begin{bmatrix} \{),]\}^* C_{11} & \{),]\}^* i \\ i & 0 \end{bmatrix}.$$

$$\prod_{\ell=1}^{2} \det(I - M_{c,\otimes \ell}(z))^{(-1)^\ell} = \prod_{\ell=1}^{2} \det(I - M_{r,\otimes \ell}(z))^{(-1)^\ell} = \frac{1}{1-2z}.$$

Example

We finally get

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

The entropy of the shift is

$$h(X) = \log\frac{1}{\rho} = \log\frac{2}{\sqrt{13}-3} \sim \log 3.3027.$$

$\mathbb{N}$-algebracity of zeta functions

Proposition

The zeta function of a finite-type-Dyck shift is a computable $\mathbb{N}$-algebraic series, i.e. is the generating series of some unambiguous context-free language

We conjecture that the result holds for all sofic-Dyck shifts.

Proper block map

A map $\Phi : X \to Y$ is called an (m, a)-*local map* (or an (m, a)-*block map*) if there exists a function $\phi : \mathcal{B}_{m+a+1}(X) \to B$ such that $\Phi(x)_i = \phi(x_{i-m} \cdots x_{i-1} x_i x_{i+1} \cdots x_{i+a})$.

A block map $\Phi : \mathrm{X}_A \to \mathrm{X}_{A'}$, where $A = (A_c, A_r, A_i)$ and $A' = (A'_c, A'_r, A'_i)$, is *proper* if $\Phi(x)_j \in A'_c$ (resp. A'_r, A'_i) whenever $x_j \in A_c$ (resp. A_r, A_i) for any j.

Proper conjugacy : conjugacy which is a proper block map.

Finite-type-Dyck shifts

Proposition

A subshift is a sofic-Dyck shift if and only it is the proper factor of a finite-type-Dyck shift.

Corollary

A proper factor of a sofic-Dyck shift is a sofic-Dyck shift.

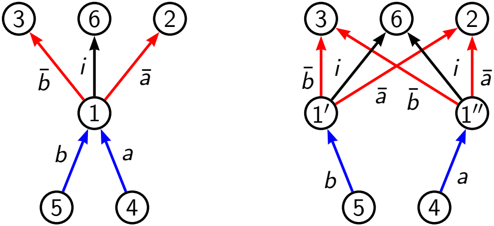
In-split of a Dyck automaton

$(\mathcal{A} = (Q, E, A), M)$ over $A = (A_c, A_r, A_i)$
Let $p \in Q$ and $\mathcal{P}$ a partition $(\mathcal{P}_1, \ldots, \mathcal{P}_k)$ of the edges coming in p.
$(\mathcal{A}' = (Q', E', A), M')$ is defined by

- $Q' = Q \setminus \{p\} \cup \{p_1, \ldots, p_k\}$,
- $(q, a, r) \in E'$ if $q, r \neq p$ and $(p, a, r) \in E$,
- $(q, a, p_i) \in E'$ for each i such that $(q, a, p) \in \mathcal{P}_i$,
- $(p_i, a, r) \in E'$ for each i such that $(p, a, r) \in E$.
- M' is the set of pairs of edges $(q, a, r), (s, b, t)$ where $a \in A_r, b \in A_c$ such that $(\pi(q), a, \pi(r)), (\pi(s), b, \pi(t)) \in M$ where $\pi(q) = q$ for $q \neq p$ and $\pi(p_i) = p$.

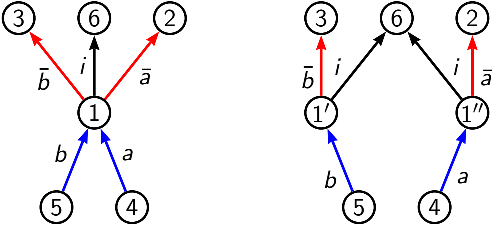
Example

A Dyck state-splitting of the state 1 into $1'$ and 1".



Trim in-split of a Dyck automaton

A trim Dyck state-splitting of the state 1 into $1'$ and 1".
Edges (p_i, a, r) which are not essential in $(\mathcal{A}', M')$ are removed from E'.
Matched pairs $(q, a, r), (p_i, b, t)$ or $(p_i, b, t), (q, a, r)$ which are not essential are removed from M'.



Edge-Dyck shifts

A *Dyck graph* $(\mathcal{G} = (Q, E \subset Q \times Q), M)$ is composed of a graph $\mathcal{G}$, where the edges $E = (E_c, E_r, E_i)$ are partitioned into three categories (*call edges*, *return edges*, and *internal edges*).

An *edge-Dyck shift* $\mathsf{X}_{(\mathcal{G},M)}$ is the set of admissible bi-infinite paths of a Dyck graph.

Proposition

Each finite-type-Dyck shift is properly conjugate to a finite-type edge-Dyck shift.

A decomposition theorem for edge-Dyck shifts

Theorem

Let (G, M), $(\mathcal{H}, N)$ be two Dyck graphs such that $X_{(\mathcal{G},M)}$ and $X_{(\mathcal{H},N)}$ are properly conjugate. Then there are finite sequences of Dyck graphs $(\mathcal{G}_i, M_i)$, $(\mathcal{H}_j, N_j)$ and Dyck (or trim Dyck) in-splittings $\Psi_i : (\mathcal{G}_i, M_i) \to (\mathcal{G}_{i+1}, M_{i+1})$, $\Delta_j : (\mathcal{H}_j, N_j) \to (\mathcal{H}_{j+1}, N_{j+1})$, such that $(\mathcal{G}_1, M_1) = (\mathcal{G}, M)$, $(\mathcal{H}_1, N_1) = (\mathcal{H}, N)$, and $(\mathcal{G}_k, M_k) = (\mathcal{H}_{k'}, N_{k'})$, up to renaming of the states.

$$(\mathcal{G}, M) \xrightarrow{\Psi_1} \dots \xrightarrow{\Psi_k} (\mathcal{G}_k, M_k) = (\mathcal{H}_{k'}, N_{k'}) \xleftarrow{\Delta_{k'}} \dots \xleftarrow{\Delta_1} (\mathcal{H}, N)$$

In-splittings commute but (unfortunately) not trim in-splittings.

Future work

- Decidability properties
- Characterization of deterministic sofic-Dyck shifts
- Minimal presentations
- Synchronization properties : sofic-Dyck shifts are not synchronized. Krieger and Matsumoto introduced the notion of λ-synchronization which is weaker and suitable for Markov-Dyck or Motzkin shifts. Which sofic-Dyck shifts are λ-synchronized ? For instance sofic-Dyck shift accepted by a Dyck-automaton which is strongly connected by well-matched words are λ-synchronized.
- λ-graphs for λ-synchronized sofic-Dyck shifts.
- Characterization of flow equivalence : proper conjugacy + internal-symbol expansion.

Preprints

M.-P. Béal, M. Blockelet, and C. Dima, "Sofic-Dyck shifts," *CoRR*, vol. http ://arxiv.org/1305.7413, 2013.

——, "Finite-type-Dyck shifts," *CoRR*, vol. http ://arxiv.org/1311.4223, 2013.

M.-P. Béal, M. Blockelet, and C. Dima, "Zeta functions of finite-type-Dyck shifts are $\mathbb{N}$-algebraic", ITA 2014.