Row-constrained effective sets of colourings are sofic in the hyperbolic plane.

Nathalie Aubrun$^1$ and Mathieu Sablik$^{1,2}$

$^1$ LIP, ENS de Lyon – CNRS – INRIA – UCBL, Université de Lyon
$^2$ I2M, Aix-Marseille Université

nathalie.aubrun@ens-lyon.fr, sablik@latp.univ-mrs.fr

Abstract. In this article we prove that, restricted to the row-constrained case, effective sets of colourings in the hyperbolic plane are sofic.

Introduction

Multidimensional subshifts of finite type (SFT) and sofic subshifts, that are closed and shift-invariant subsets of colourings of $\mathbb{Z}^d$ for $d \geq 2$ given by local rules, have strong computational properties. For instance, it is not possible to decided whether such a subshift is empty or not [Ber66]. A clever result by Hochman [Hoc09], then improved independently in [AS13] and [DRS10], states that up to an increase of the dimension, effective subshifts are very close to sofic subshifts. In more details, any effective subshift of dimension $d$ can be found as the projective subaction of a sofic subshift of dimension $d + 1$. Symbolic dynamics can be defined on structures more general than $\mathbb{Z}^d$, for instance finitely presented groups, and a natural question is to determine whether results similar to Hochman’s can be proved in this case.

Our intuition is that we can obtain an even stronger result than Hochman’s on the hyperbolic plane. Two facts strengthen our intuition. First it is possible to encode Turing machine computations with local rules on the hyperbolic plane [Rob78]. Second the counting argument (see [Van12, p. 14]) used to prove the non soficness of the mirror subshift in $\mathbb{Z}^2$ – one example of effective subshift that can be proven not to be sofic – cannot be applied on the hyperbolic plane, for non-amenability reasons. This leads us to formulate the following conjecture, that basically means that the dimension increase is no longer needed to get a result similar to Hochman’s in the hyperbolic plane.

Conjecture 1. Effective subshifts are sofic on the hyperbolic plane.
Unfortunately we are for now unable to prove this conjecture, but we present here a preliminary result that will hopefully be a first step in proving Conjecture 1. The idea is to simplify the problem by enforcing the border of the configurations. In the Poincaré disk model, this corresponds to choosing a small disk tangent to the border of the whole Poincaré disk, and forcing colours inside that small disk. In the Poincaré upper half-plane model, this means that one horizontal line and everything above it is fixed. Note that the sets of configurations we now consider are no longer subshifts, since they do not satisfy any shift-invariance property. Nevertheless this approach makes sense for at least two reasons. First is corresponds to the intuitive vision of the way somebody would try to tile a surface with a set of tiles: the person would start his tiling on the border of the surface. The second reason is historical: before the proof of the undecidability of the domino problem on $\mathbb{Z}^2$ by Berger [Ber66] (deciding whether an SFT is empty or not) Wang first proved that the row-constrained problem, where a single tile is forced to appear, is undecidable [Wan61]. Also in the case of the hyperbolic plane, before Kari’s [Kar07] and Margenstern’s [Mar08] proofs of the undecidability of the unconstrained domino problem, Robinson remarked that the origin fixed domino problem was also undecidable [Rob78]. Note that here we consider something a little bit more general than fixing a single tile, since we fix an entire line.

The paper is organized as follows. In Section 1 we present the Poincaré upper half-plane model, then define hyperbolic tilings and explain how to encode Turing machine computations inside such objects. Section 2 is devoted to the dyadic encoding, a basic transformation on tilings that exploits the hyperbolic structure of our model. Finally in Section 3 we prove the main result.

1 Sets of colourings of the hyperbolic plane $\mathbb{H}^2$

In this section we present a formalism to define tilings on the hyperbolic plane. We consider generalized tilings, called sets of colourings, where local rules that define allowed configurations are not necessarily finite in number.

1.1 The hyperbolic plane $\mathbb{H}^2$

In Figure 1, we show what is usually called the 2-fold horocyclic tesselation, depicted in the upper-half plane model of the hyperbolic plane. The tiles are arranged hierarchically, each sitting above two other tiles.
The are uncountably many tessellations of $\mathbb{H}^2$ with these tiles, but in the sequel we will work with only one of them. For more convenience, we will allow us to locate tiles by a finite word on the alphabet $\{\alpha, \alpha^{-1}, \beta, \beta^{-1}\}$. To do so, we arbitrarily choose a tile that will be represented by the empty word $\varepsilon$. Then if a tile is represented by the word $g$, its bottom left neighbour (resp. bottom right neighbour) will be represented by $g \cdot \alpha$ (resp. $g \cdot \alpha \cdot \beta$). This rule allows to represent every tile below the tile $\varepsilon$ by a word, the rest of the tessellation is obtained by using the trivial rules $\alpha \cdot \alpha^{-1} = \beta \cdot \beta^{-1} = \varepsilon$. In order to make these representations consistent we also need to add the relation $\alpha \cdot \beta^2 = \beta \cdot \alpha$. Thus the tessellation of $\mathbb{H}^2$ is embedded inside Baumslag-Solitar group $< a, b | ab^2 = ba >$ – more precisely one sheet of the Cayley graph of this group. Note that to one tile correspond infinitely many finite words, but not all finite words correspond to a tile in the tessellation.

![Fig. 1. One 2-fold horocyclic tessellation of $\mathbb{H}^2$.](image)

### 1.2 Sets of colourings on $\mathbb{H}^2$

Suppose the origin $\varepsilon$ is fixed, and let $g \in \{\alpha, \alpha^{-1}, \beta, \beta^{-1}\}^*$ be a position in $\mathbb{H}^2$. We denote by $U_n$ the *support of size $n$*, defined as

$$U_n = \{\alpha^p \cdot \beta^q : 0 \leq p \leq n - 1, 0 \leq q \leq 2^p - 1\},$$

and by $L_n$ *linear support of size $n$*, defined as

$$L_n = \{\alpha^{n+1} \cdot \beta^q : 0 \leq q \leq 2^{n+1} - 1\}.$$

Let $A$ be a finite alphabet. A *configuration* is a colouring of the 2-fold horocyclic tessellation of $\mathbb{H}^2$ with colours chosen in $A$. By abuse of
notation, we denote the set of configurations by $A^{\mathbb{H}^2}$. A pattern of size $n$ is a finite configuration $p \in A^{U_n}$, and $U_n$ is thus called the support of $p$. A linear pattern of size $n$ is a finite configuration $p \in A^{L_n}$.

Fig. 2. The supports $U_0$, $U_1$, $U_2$ and $L_1$.

We say that a pattern $p \in A^{U_n}$ appears in a configuration $x \in A^{\mathbb{H}^2}$ if there exists some position $g \in \{\alpha, \alpha^{-1}, \beta, \beta^{-1}\}^*$ such that $p = x|_g \cdot U_n$. Let $F$ be a set of patterns, it defines a set of colourings $\Sigma_F \subseteq A^{\mathbb{H}^2}$ as the set of configurations that avoid every pattern in $F$

$$\Sigma_F = \{x \in A^{\mathbb{H}^2} : \text{no pattern of } F \text{ appears in } x\}.$$

This notion of set of colourings is very closed to the classical notion of subshift in symbolic dynamics – at least from a combinatorial point of view – but in the case of the 2-fold horocyclic tessellation, we lack a real shift action to properly define subshifts as dynamical objects.

**Definition 1.** A set of colourings $\Sigma \subseteq A^{\mathbb{H}^2}$ is

1. of finite type (CFT) if there exists a finite set of forbidden patterns that defines it;
2. sofic if there exists a CFT $\Sigma' \subseteq B^{\mathbb{H}^2}$ and a letter-to-letter map $\Phi : B \to A$ such that $\Sigma = \Phi(\Sigma')$;
3. effective if there exists a recursively enumerable set of forbidden patterns that defines it.

1.3 Computation of Turing machine inside a CFT on $\mathbb{H}^2$

The idea is to put Turing machine computations the same way it is usually done in $\mathbb{Z}^2$. In this aim, we encode by local rules the lattice $\mathbb{Z} \times \mathbb{N}$ inside $\mathbb{H}^2$ (see the tiles marked by a $\bullet$ symbol in Figure 3). We denote this set of colourings by $\Sigma_{\mathbb{Z} \times \mathbb{N}} \subset \{\bullet, \emptyset\}^{\mathbb{H}^2}$. Obviously, $\Sigma_{\mathbb{Z} \times \mathbb{N}}$ is CFT, but contains the uniform configuration $\emptyset^{\mathbb{H}^2}$ if we do not impose more constraints.

Once one has this lattice, it is possible to encode the behaviour of any given Turing machine by local rules (see again Figure 3). Remember
that a Turing machine is a model of calculation composed by a finite automaton – the head of calculation – that can be in different states and moves on an infinite tape divided into boxes, each box containing a letter that can be modified by the head.

**Definition 2.** Let \( M = (Q, \Gamma, \#, q_0, \delta, Q_F) \) be a Turing machine, where:

- \( Q \) is a finite set of states of the head of calculation; \( q_0 \in Q \) is the initial state;
- \( \Gamma \) is a finite alphabet;
- \( \# \notin \Gamma \) is the blank symbol, with which the tape is initially filled;
- \( \delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{\leftarrow, \rightarrow\} \) is the transition function. Given the state of the head of calculation and the letter it can read on the tape — which thus depends on the position of the head of calculation on the tape — the letter on the tape is replaced or not by another one, the head of calculation moves to an adjacent box and changes or not of state;
- \( F \subset Q_F \) is the set of final states.

We do not give in details the alphabet \( A_M \) used to encode computation of a given Turing machine \( M \), but all letters in \( A_M \) and local rules appear in Figure 3. The idea is basically to use the encoding \( \Sigma_{\mathbb{Z} \times \mathbb{N}} \) of the lattice \( \mathbb{Z} \times \mathbb{N} \) and to adapt the classical construction by Wang [Wan61] to make sure that we get the space-time diagram of the Turing machine \( M \) starting on the empty word. When we fix a ●-line in \( \mathbb{H}^2 \), where the initial state \( q_0 \) appears once on a blank tape, one can easily get a computation zone which is infinite both in space and time (see Figure 3).

Without a lot of efforts, one can deduce from this construction the undecidability of the tiling problem with row-constrained in the hyperbolic plane (see [Rob78] for the original proof). The same result without the row-constrained assumption was proven only thirty years later by two different techniques (see [Kar07] for a proof on the 2-fold horocyclic tessellation of the hyperbolic plane and [Mar08] for a proof on (7, 3)-tessellation of Poincaré disc with heptagons).
Fig. 3. We consider the Turing machine $M_{ex}$ that enumerates on its tape the words $ab, aabb, aaabbb, \ldots$ and never halts. This machine uses the three letters alphabet $\{a, b, \parallel\}$ and five states $Q = \{q_0, q_a, q_b, q_{b+}, q_{\parallel}\}$. A separation symbol $\parallel$ is written at the end of each $a^n b^n$. On the top, an example of computation encoded inside a $4 \times 4$-grid in $\mathbb{Z}^2$. On the bottom, the same computation encoded inside a $4 \times 4$-grid in the hyperbolic plane: the grid is marked by $\bullet$ symbols.
2 Dyadic encoding in the hyperbolic half-plane

In this section we present a transformation on subshifts, that preserve both soficness and the property of being of finite type, that is based on the observation that every row of the hyperbolic half-plane contains twice as much cells as the row directly above it.

2.1 Encoding on a single row all rows above it

We define a global function on configurations $\Phi$ that doubles the alphabet.

$$\Phi : A^{\mathbb{H}^2} \to (A \times A)^{\mathbb{H}^2}$$

Given a configuration $x \in A^{\mathbb{H}^2}$, we define $\Phi(x)$ as the configuration of $(A \times A)^{\mathbb{H}^2}$ such that its restriction to the first letter gives $x$, and the restriction to the second letter is given by the local rule pictured on Figure 4. Formally, for any position $g$ in $\mathbb{H}^2$, one has

$$\pi_1(\Phi(x)) = x$$
$$\pi_2(\Phi(x))_{ga} = \pi_2(\Phi(x))_g$$
$$\pi_2(\Phi(x))_{ga\beta} = \pi_1(\Phi(x))_g$$

for every $x \in A^{\mathbb{H}^2}$ where $\pi_i$ denotes the projection on the $i$th letter for $i \in \{1, 2\}$. Consequently for all $n \geq 1$ one has

$$\pi_2(\Phi(x))_{ga\beta n} = \pi_2(\Phi(x))_{ga\beta n-1} = \pi_2(\Phi(x))_{ga} = \pi_1(\Phi(x))_g.$$
For a given subshift $\Sigma \subseteq A^\mathbb{Z}$, the subshift $\Phi(\Sigma) \subseteq (A \times A)^\mathbb{Z}$ is called the dyadic encoding of $\Sigma$. Obviously if $\Sigma$ is SFT (resp. sofic, effective), then so is $\Phi(\Sigma)$.

2.2 Detecting patterns

In this section we describe more precisely how the transformation $\Phi$ acts on allowed and forbidden patterns of a set of colourings $\Sigma$.

Let $p \in A^{U_n}$, define $\tilde{p} = \{ \pi_2(\Phi(x))_{L_n} : x \in A^\mathbb{Z} \text{ such that } x_{U_n} = p \}$ the set of patterns which appear in the bottom of the pattern $p$ after application of $\Phi$. Some letters of an element in $\tilde{p}$ code letters in $p$, and others code letters that appears in $x$ outside $U_n$. If there is no ambiguity, denote $\tilde{p}_i$ as a letter.

**Proposition 1.** The pattern $p \in A^{U_n}$ appears in a configuration $x \in A^\mathbb{Z}$ in position $g$ (i.e. $p = x_{gU_n}$) if and only if an element of $\tilde{p}$ appears in $\Phi(x)$ in position $g \cdot L_n$ (i.e. $\pi_2(\Phi(x))_{g\cdot L_n} \in \tilde{p}$).

**Proof.** Let $p \in A^{U_n}$ and consider $\alpha^n \beta^q \in U_n$ with $0 \leq p \leq n$ and $0 \leq q \leq 2^n - 1$. By construction of $\Phi$, if $x \in A^\mathbb{Z}$ such that $p = x_{U_n}$ one has

$$\pi_2(\Phi(x))_{\alpha^{n+1} \beta (2^n + 1) 2^n - p} = \pi_2(\Phi(x))_{\alpha^n \beta \alpha^{n+1} - \beta 2^n - p} = \pi_1(\Phi(x))_{\alpha^n \beta^q}.$$ 

The proposition follows.

Proposition 1 means that the whole information about a pattern with support $U_n$ is entirely contained in a linear pattern with support $L_n$ of its dyadic encoding. Thus looking for occurrences of a pattern $p$ in a configuration $x$ is the same as looking for occurrences of $\tilde{p}$ in the configuration $\Phi(x)$.

**Proposition 2.** Let $x \in A^\mathbb{Z}$, $g \in 2^\mathbb{Z}$ and $n \in \mathbb{N}$. Consider the pattern $p = x_{gU_n}$ of support $U_n$, then

$$\pi_2(\Phi(x))_{\alpha^n \beta \alpha^{n+1} \beta^k 2^k - 1} = \tilde{p}_i \text{ for all } k \geq 1 \text{ and } i \in \{0, \ldots |\tilde{p}| - 1\}.$$ 

**Proof.** From Proposition 1 we deduce that $\pi_2(\Phi(x))_{\alpha^n \beta \alpha^{n+1} \beta^k} = \tilde{p}_i$ for all $i \in \{0, \ldots |\tilde{p}| - 1\}$. The result follows from the fact that for all $k \geq 1$ one has $\pi_2(\Phi(x))_{\alpha^n \beta^k} = \pi_2(\Phi(x))_{g \cdot \beta^k} = \alpha^k \beta^{2^k - 1}$. 
3 Effective sets of colourings are sofic on the hyperbolic half-plane

We say that a set of colourings $T \subset A^{\mathbb{H}^2}$ is row-constrained if there exists a special symbol $\approx \in A$ that appears in every configuration $x \in T$, and such that its presence forces all letters on rows above it (including letters on the same row) to be also $\approx$ – we say that the letter $\approx$ has the half-plane property. In this section we prove the following result.

**Theorem 1.** Any row-constrained effective set of colourings on $\mathbb{H}^2$ is sofic. In other words, if $T$ is an effective set of colorings on $A$ and if one letter $\approx \in A$ has the half-plane property, then the row-constrained set of colorings $T \cap \{ x \in A^{\mathbb{H}^2}, \approx \text{ appears in } x \}$ is sofic.

For more readibility, if $x$ is a configuration of a row-constrained set of colorings, we do not picture the half-plane filled with $\approx$ but letters below this half plane with a double line on the top of the pentagon (see Figure 5 for instance).

3.1 Sketch of the proof

We will encode Turing machine computations inside a row-constrained CFT on $\mathbb{H}^2$. Thanks to the dyadic encoding presented in Section 2 it is enough to check the occurrences of forbidden patterns produced by this machine on infinitely many rows in the dyadic encoding of the original configuration.

3.2 A four layers construction

Let $T$ be a row-constrained effective set of colourings on some alphabet $A$. Let $\mathcal{M}$ be a Turing machine that enumerates a set of forbidden patterns for $T$ – we assume that the machine runs on a one-sided tape. We construct a four layers row-constrained CFT $\Sigma_\mathcal{M} = \Sigma_1 \times \Sigma_2 \times \Sigma_3 \times \Sigma_4$. Remark that in this construction, only the third layer $\Sigma_3$ will be row-constrained, and so will be $\Sigma_\mathcal{M}$.

**First layer: configurations in $A^{\mathbb{H}^2}$.** The first layer $\Sigma_1$ only contains configurations in $A^{\mathbb{H}^2}$, with no constraint on them, except the ones that will be given by interaction with other layers.
**Second layer: computation zones.** The second layer $\Sigma_2$ contains computation zones. First define the row-constrained CFT $\Sigma'_2$ on alphabet $\{a, b, a^*, b^*\}$ as the one defined by the set of allowed patterns appearing in Figure 5. Then $\Sigma_2$ is the row-constrained CFT on the product alphabet $\{a, b, a^*, b^*\} \times \{\bullet, \emptyset\}$, seen as a subset of $\Sigma'_2 \times \Sigma_{\mathbb{Z} \times \mathbb{N}}$ with the additional rule that a letter of the product alphabet with either $a^*$ or $b^*$ on its first coordinate is always associated with $\bullet$ on its second coordinate.

![Fig. 5. An example of row-constrained configuration on $\{a, b, a^*, b^*\}^{\mathbb{Z}_2}$ in $\Sigma'_2$. For more readability, letters $a$ and $a^*$ are represented by grey cells and letters $b$ and $b^*$ by white cells.](image)

Let $x \in \Sigma_2$. We call type a computation zone (resp. type b computation zone) a pattern uniformly filled with product letters with $a$’s and $a^*$’s (resp. $b$’s and $b^*$’s) on its first coordinate, with support $\bigcup_{i=1}^{k} \beta^i \cdot L_n$ where $n$ – the height – and $k$ – the width – are maximal, that appears in $x$. This second layer is made such that every configuration in $\Sigma_2$ is made of computation zones of type $a$ and $b$ that alternate, and type $a$ zones merge with their type $b$ right neighbour to form a larger computation zone. With no more constraints, these patterns may define infinitely high computation zones of bounded width at some point, as suggested in Figure 5, or arbitrarily wide computation zones of bounded height. This would be problematic, since we will need arbitrarily large – in both directions – computation zones. We will see how this problem is fixed by interacting with the fourth layer.

**Third layer: dyadic encoding.** The third layer $\Sigma_3$ contains the dyadic encoding of the first layer $\Phi(\Sigma_1)$. This can be done using local rules between first and third layers.
Fourth layer: Turing machine calculations. The fourth layer $\Sigma_4$ contains calculations of a Turing machine $\widetilde{M}$ that has the following behaviour

1. the machine $\widetilde{M}$ has two tapes, the computation tape that is initially filled with blank symbols $\#$, of which width coincides with the width of the computation zone, and the detecting tape of which width coincides with the width of the computation zone and its left and right neighbours (hence three times wider than the computation tape) on which the first row of $\Sigma_1$ is copied out. The widths and overlappings of detecting tapes ensure that any linear pattern will eventually be contained in a single detecting tape.

2. the machine $\widetilde{M}$ simulates $M$ to produce a forbidden pattern $p$ for $T$, and transforms it into the set of patterns $\tilde{p}$

3. it checks whether every element of $\tilde{p}$ appears or not on the detecting tape

4. if a pattern of $\tilde{p}$ is detected, then the machine $\widetilde{M}$ instantaneously reaches a special state $q_f$, that will be forbidden in the final set of colourings

5. once the head of calculation tries to go to the right of the rightmost cell in the current computation zone, the computation zone is closed and merges with its right or left neighbour to get a twice bigger computation zone. This can be done using local rules between second and fourth layers.

3.3 Main result

Theorem 1 Any row-constrained effective set of colourings on $\mathbb{H}^2$ is sofic.

Proof. Let $T$ be a row-constrained effective set of colourings. We denote by $\Sigma = \pi_1(\Sigma_M)$ where $\Sigma_M$ is the sofic set of colourings described in Section 3.2. It is straightforward that the construction described in Section 3.2 provides a sofic set of colourings $\Sigma$ such that $T \subseteq \Sigma$. It remains to prove that the local rules defining $\Sigma$ forces all configurations $x \in \Sigma$ to also belong to $T$.

Let $x$ be a configuration in $\Sigma$, and suppose that a pattern $p = \pi_1(x_{x\cdot U_n})$ is forbidden in $T$. Then the Turing machine that enumerates forbidden patterns in $T$ will enumerate $p$ at some point. By construction of the computation zones on the second layer, there exists a computation zone large enough to check whether one element of $\tilde{p}$ appears on its associated detecting tape. By Proposition 2, we know that one element of
\( \bar{p} \) appears on every row in \( \mathbb{H}^2 \) containing the element \( g \cdot \alpha^{n+k} \) for every \( k \geq 1 \) – every row which is located below pattern \( p \). Thus any forbidden pattern will eventually be detected, which ensures that \( x \) is in \( T \).

4 Conclusion

The ideas presented in this article constitute only a first step in proving Conjecture 1. The natural idea would consist in using Goodman-Strauss hierarchical aperiodic tiling of the hyperbolic plane [GS10] to define arbitrarily large computation zones without any constraints. But a few points still remain to be lightened. Note also that Conjecture 1 is optimal in a certain sense: the statement is no longer true on Baumslag-Solitar group \( \langle a, b \mid ab^2 = ba \rangle \), since the counting argument presented in [Van12, p. 14] can be adapted to prove that the mirror subshift is not sofic on this structure.

References


