




Subshifts Defined by Nondeterministic and Alternating Plane-Walking Automata

Benjamin Hellouin de Menibus   

Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique, 91400, Orsay, France

Pacôme Perrotin   

Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique, 91400, Orsay, France

*Il voulut être sofic,
il ne fut que pompé.*

Abstract

Plane-walking automata were introduced by Salo & Törma to recognise languages of two-dimensional infinite words (subshifts), the counterpart of 4-way finite automata for two-dimensional finite words. We extend the model to allow for nondeterminism and alternation of quantifiers. We prove that the recognised subshifts form a strict subclass of sofic subshifts, and that the classes corresponding to existential and universal nondeterminism are incomparable and both larger than the deterministic class. We define a hierarchy of subshifts recognised by plane-walking automata with alternating quantifiers, which we conjecture to be strict.

2012 ACM Subject Classification Theory of computation → Automata over infinite objects; Theory of computation → Tree languages; Theory of computation → Regular languages

Keywords and phrases Formal languages, Finite automata, Subshifts, Symbolic dynamics, Tilings

Digital Object Identifier 10.4230/LIPIcs.STACS.2025.48

Funding The authors acknowledge financial support from the ANR-22-CE40-0011 project Inside Zero Entropy Systems.

Acknowledgements Many people have contributed to this article through discussions, suggestions and failed attempts; we wish to thank (in alphabetical order) Florent Becker, Amélia Durbec, Pierre Guillon and Guillaume Theyssier. We are grateful to Ville Salo for providing the proof of Theorem 8 and pointing us towards the Kari-Moore rectangles of Section 5.3, and to Denis Kuperberg and Thomas Colcombet for pointing us towards works on tree-walking automata cited in Section 5.2.

1 Introduction

One-dimensional finite automata are a classical model to recognise languages of finite words. They have been extended to recognise languages of finite patterns in two and more dimensions, often called *picture languages*, starting with the work of Blum and Hewitt in 1967 [3]. While the one-dimensional model is very robust to changes in definition, this is not the case in higher dimensions and many different models have been introduced with varying computational power; see [11] for a survey that focuses on the non-deterministic case.

Symbolic dynamics are concerned with subshifts, which are languages of infinite words or patterns. In dimension 1, sofic subshifts can be seen as the infinite counterparts to regular languages, and have three equivalent definitions: the set of infinite walks on a labelled graph (finite automaton without initial nor final states); the set of infinite words avoiding a regular set of forbidden subwords; the letter-to-letter projection of a subshift of finite type. The latter definition carries through to higher dimensions without difficulties to define higher-dimensional sofic subshifts.



© Benjamin Hellouin de Menibus and Pacôme Perrotin;
licensed under Creative Commons License CC-BY 4.0

42nd International Symposium on Theoretical Aspects of Computer Science (STACS 2025).

Editors: Olaf Beyersdorff, Michał Pilipczuk, Elaine Pimentel, and Nguyễn Kim Thăng;

Article No. 48; pp. 48:1–48:15



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



There are many ways to extend regular languages to higher dimensions. Some models such as *4-way automata* keep the notion of a linear *run* where the head reads the input pattern, letter by letter, by moving (“walking”) over the pattern; this contrasts with models such as *recognisable languages* where acceptance consists in checking local constraints instead of a run. Broadly speaking, the latter models are more powerful than the former, and the same phenomenon arises for languages on trees: tree automata vs. tree-walking automata. [7] provides a catalogue of the different kinds of models, while a more recent survey on the “walking” models can be found in [11].

When applying these models to subshifts, recognisable languages yields an alternative definition of sofic subshifts; this was first done in [8] to our knowledge, and we present this construction in Section 2.3. Recently, Salo and Törma introduced *plane-walking automata* (PWA) [14], a particular case of graph-walking automata, which are a counterpart of 4-way automata for infinite patterns. In particular, they proved that deterministic plane-walking automata define a class of subshifts that is strictly between subshifts of finite type and sofic subshifts, extending to infinite patterns a result from [9] on 4-way automata. It is also proved in [9] that, for finite patterns, nondeterministic 4-way automata are strictly more powerful than the deterministic version and that languages of alternating 4-way automata are incomparable with recognisable languages, a sharp contrast with the one-dimensional case.

In this paper, we introduce nondeterminism and alternations to plane-walking automata and consider the classes of higher-dimensional subshifts obtained this way. We prove that subshifts accepted by Σ_1 PWA (existential nondeterminism) and Π_1 PWA (universal nondeterminism) form incomparable classes (Theorem 14), both strictly larger than the deterministic case, and that subshifts accepted by arbitrary alternating PWA still form a strict subclass of sofic subshifts (Theorem 8). This yields two new classes of higher-dimensional subshifts that are intermediate between finite type and sofic subshifts, and natural in that they generalise one-dimensional sofic subshifts. We introduce an alternating hierarchy of nondeterministic power from deterministic up to unbounded alternating plane-walking automata, and conjecture that this hierarchy of subshifts does not collapse; to our knowledge, this is not known even for finite words. Our proofs involve equivalents of the pumping lemma for two-dimensional infinite patterns.

Definitions and results are written with the two-dimensional case (\mathbb{Z}^2) in mind, although they extend easily to any higher dimension, and our definitions also extend to any finitely generated groups (see [15]).

2 Configurations and Subshifts

2.1 Symbolic Dynamics

We call *positions* the elements of \mathbb{Z}^2 , which is endowed with the Manhattan distance d . We use $\rightarrow = (1, 0)$ and $\uparrow = (0, 1)$, $\bullet = (0, 0)$ and so on; we often write 0 as a position instead of $(0, 0)$.

Let Σ be a finite set called *alphabet*. A *configuration* x is an element of $\Sigma^{\mathbb{Z}^2}$, while a *pattern* p is an element of Σ^S for some finite $S \subset \mathbb{Z}^2$, called the support of p and denoted $\text{supp}(p) = S$. For $i \in \mathbb{Z}^2$, $\sigma^i(x)$ is a new configuration shifted by i , i.e. $(\sigma^i(x))_j = x_{i+j}$. In dimension one, we call a pattern a *word*.

Given $\pi : \Sigma' \rightarrow \Sigma$, we extend π to $\Sigma'^{\mathbb{Z}^2}$ by putting $\pi(x)_p = \pi(x_p)$ for all $p \in \mathbb{Z}^2$.

A *subshift* X is a set of configurations that is closed in the product topology and invariant by all shifts; more concretely, it is defined as the set of configurations where no pattern from a set of forbidden patterns \mathcal{F} appears.

A subshift defined by a finite set of forbidden patterns is called *of finite type* (SFT for short). By a standard technique of higher block coding, replacing the alphabet Σ by $\Sigma^{[0,n] \times [0,n]}$ for n large enough, we can assume without loss of generality that forbidden patterns all have support $\{\bullet, \rightarrow\}$ or $\{\bullet, \uparrow\}$, and we do throughout the paper.

A subshift that can be written as $\pi(X)$, where X is an SFT on alphabet Σ' and $\pi : \Sigma' \rightarrow \Sigma$, is called *sofic*. In dimension one, sofic subshifts have alternative definitions as the set of bi-infinite walks on some labelled graph or as the set of bi-infinite words avoiding some regular set of forbidden words.

We denote the classes of SFT and sofic subshifts SFT and Sofic, respectively.

2.2 Two-dimensional Automata

We define an abstract model of two-dimensional automata. We use different notions of acceptance in the paper, and add additional restrictions to the model as necessary.

► **Definition 1.** *A two-dimensional automaton on \mathbb{Z}^2 is a labelled directed graph $A = (V, E, \Sigma, D, I)$, where*

- *V and E are finite sets of vertices and edges, respectively, where $E \subset V^2 \times \mathbb{Z}^2$ (we call the second component the direction);*
- *Σ is a finite alphabet and $D : V \rightarrow \Sigma$ associates a symbol to each vertex.*
- *$I \in V$ are the initial states and D is a bijection from I to Σ . For any $a \in \Sigma$, we denote i_a the only state in I such that $D(i_a) = a$.*

If the automaton is alternating, then there is additionally a map $Q : V \rightarrow \{\forall, \exists\}$ associating a quantifier to each vertex.

Notice that since E is finite, the set of possible directions is a finite subset of \mathbb{Z}^2 .

2.3 Recognisable Picture Languages

Recognisable languages, as introduced in [6], are a possible extension of regular languages to higher-dimensional words (or *picture languages*) as projections of local languages, which can be expressed using two-dimensional automata.

The same model applied on subshifts yields an alternative definition of sofic subshifts using two-dimensional automata.

► **Definition 2.** *Let A be a non-alternating two-dimensional automaton whose set of directions is $\{\uparrow, \rightarrow\}$. Given a pattern or a configuration x , an recognising run of A on x is a function $r : \text{supp}(x) \rightarrow V$ such that:*

- *for all $i \in \text{supp}(x)$, $D(r(i)) = x_i$;*
- *for all $i \in \text{supp}(x)$ such that $i + \rightarrow \in \text{supp}(x)$, there is an edge $(r(i), r(i + \rightarrow), \rightarrow) \in E$, and similarly for \uparrow .*

Then:

- *the recognised language $R_f(A)$ is the set of all patterns that admit a recognising run.*
- *the recognised subshift $R_\infty(A)$ is the set of all configurations that admit a recognising run.*

Notice that this definition is intrinsically nondeterministic in the choice of the run and makes no use of initial states, so we omit I in this section. The first equivalence of the following result appeared in [8, Proposition 7]; we provide a short proof in our framework.

► **Proposition 3.** *For a subshift X , the following are equivalent:*

1. X is sofic;
2. $X = R_\infty(A)$ for some automaton A ;
3. X is defined by a set of forbidden patterns $R_f(A)^c$ for some automaton A , where A is assumed to satisfy the hypotheses of Definition 2.

Proof. (1 \Rightarrow 2) Let Y be a SFT on alphabet Σ' given by a finite set \mathcal{F} of forbidden patterns and $\pi : \Sigma' \rightarrow \Sigma$ be such that $\pi(Y) = X$; remember that we assume that patterns in \mathcal{F} have support $\{\bullet, \uparrow\}$ or $\{\bullet, \rightarrow\}$. We define $A = (V, E, \Sigma', D)$ by setting $V = \Sigma'$ and $D = \pi$, and E is defined as follows: for all $p \in \Sigma'^{\{\bullet, \uparrow\}}$, $(p_\bullet, p_\uparrow, \uparrow) \in E$ if and only if $p \notin \mathcal{F}$, and similarly for \rightarrow . By construction, $y \in \Sigma'^{\mathbb{Z}^2}$ is a recognising run of A on x if and only if $y \in Y$ and $\pi(y) = x$, so $X = R_\infty(A)$.

(2 \Rightarrow 1). Let $A = (V, E, \Sigma', D)$ be an automaton and define the SFT $Y \subset V^{\mathbb{Z}^2}$ by the set of forbidden patterns:

$$\mathcal{F} = \{p \in V^{\{\bullet, \rightarrow\}} : (p_\bullet, p_\rightarrow, \rightarrow) \notin E\} \cup \{p \in V^{\{\bullet, \uparrow\}} : (p_\bullet, p_\uparrow, \uparrow) \notin E\}.$$

Again by construction, y is a recognising run of A on x if and only if $y \in Y$ and $D(y) = x$, so $X = D(Y)$ is sofic.

(2 \Leftrightarrow 3) The two statements are equivalent for any automaton A . If $x \in R_\infty(A)$ has a recognising run, then the restriction of this run to any finite pattern in x is recognising, so all patterns in x belong to $R_f(A)$. Conversely, if all patterns in x admit a recognising run, then x does as well by a standard compactness argument. ◀

Notice that the third condition involves complementation, in contrast with the one-dimensional case; recognisable languages are not closed by complement in dimension 2 [1].

3 Plane-walking Automata and Associated Subshifts

Plane-walking automata generalise the definition of one-dimensional sofic subshifts seen as the set of infinite walks on a labelled graph.

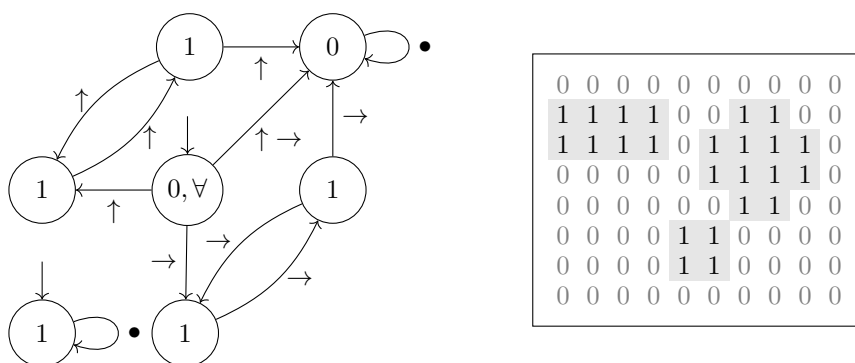
3.1 Definitions

The automata we use in this section correspond to the plane-walking automata (PWA) from Salo and Törnä [14] with one head and added nondeterminism. They are alternating two-dimensional automata with a specific acceptance condition, that we call alternating plane-walking automata to be consistent with the literature.

► **Definition 4** (Subshifts defined by plane-walking automata). *Let $A = (V, E, \Sigma, D, I, Q)$ be an alternating PWA. Given $x \in \Sigma^{\mathbb{Z}^2}$, $p \in \mathbb{Z}^2$ and $v \in V$, there is an accepting run on x starting from (p, v) if $D(v) = x_p$ and:*

- $Q(v) = \exists$ and there is an edge $(v, v', d) \in E$ and an accepting run starting from $(p + d, v')$, or
- $Q(v) = \forall$ and all edges $(v, v', d) \in E$ with $D(v') = x_{p+d}$ have an accepting run starting from $(p + d, v')$; furthermore, there must be one such edge.

A configuration x is accepted by A if, for every $p \in \mathbb{Z}^2$, there is an accepting run of A on x starting from (p, i_{x_p}) . The set of configurations accepted by A is a subshift, denoted $L_\infty(A)$.



■ **Figure 1** On the left, an example of an alternating plane-walking automaton. On the right, a finite pattern of a configuration accepted by the automaton. The associated subshift is the subshift where all vertical and horizontal runs of 1s are either of even size or infinite.

The lack of a base case in the previous definition means that an accepting run must be infinite; in other words, a run accepts if it never reaches a position and state with no possible move. The model of Salo and Törma used rejecting states, which we opted not to do by symmetry w.r.t the lack of accepting states; this is a stylistic choice as a rejecting state can be simulated by a state with no outgoing edge, and an accepting state by a state with a loop with direction \bullet .

An alternating plane-walking automaton and its associated subshift are illustrated in Figure 1. Without loss of generality, by adding additional states, the set of directions can be restricted to $\{\uparrow, \downarrow, \leftarrow, \rightarrow, \bullet\}$ which we assume in the rest of this article.

We denote Alt the class of subshifts X such that $X = L_\infty(A)$ for some alternating PWA A . It is not difficult, as in Proposition 3, to extend the notion of acceptance to finite patterns (considering a run as accepting when it leaves the support of the pattern), where it coincides with alternating 4-way picture-walking automata. Subshifts in Alt can equivalently be defined by a set of forbidden patterns that are the complement of the language of some alternating PWA. As for recognisable languages, these languages are not closed by complement [11, Theorem 7].

Plane-walking automata as considered by Salo & Törma [14] are deterministic, in the sense that there is only one outgoing edge from each state on which the run does not fail immediately; therefore the quantifiers in the definition are unused. We denote by Δ_1 the class of subshifts defined by such deterministic plane-walking automata, and we call deterministic every state where the quantifier is not relevant, so we omit it in the pictures for clarity.

► **Definition 5** (Branch, Footprint). *A run on x can be represented by a tree where each vertex corresponds to a current position and state.*

A branch of a run is a branch in that tree, in the usual sense.

The footprint of a subtree or a branch is the set of all visited positions.

3.2 Comparison with SFT and Sofic Subshifts

We show that subshifts defined by alternating plane-walking automata are an intermediate class between the two classical classes of SFT and sofic subshifts. Notice that, in dimension one, the classical powerset construction tells us that added nondeterminism does not impact the power of the finite automata, so every class defined in this article coincide with sofic subshifts.

The next result is proved in [14] (Inclusion is stated without proof, Strictness is Lemma 1; Inclusion also follows from Lemma 15 below).

► **Proposition 6.** $SFT \subsetneq \Delta_1 \subset Alt$.

The following result appeared in [9] (Theorem 1) for finite patterns. We translate the proof in our framework since our statements differ due to differing models.

► **Proposition 7.** $Alt \subset Sofic$.

Proof. Let $A = (V, E, \Sigma, D, I, Q)$ be an alternating PWA. Let $\Sigma' = \Sigma \times \mathcal{P}(V)$, where a symbol from Σ' encodes the set of states starting from which there is an accepting run in the current position. Let π_1 and π_2 be the projections on the first and second component, respectively.

We define a SFT $Y \subset \Sigma'^{\mathbb{Z}^2}$ by the following set \mathcal{F} of forbidden patterns: for $p \in \Sigma'^{\{\leftarrow, \rightarrow, \uparrow, \downarrow, \bullet\}}$, denote $p_\bullet = (s, S)$ where $s \in \Sigma$ and $S \in \mathcal{P}(V)$. Then $p \in \mathcal{F}$ if and only if:

1. $\exists v \in S, D(v) \neq s$, or
2. $S \cap I = \emptyset$, or
3. there is $v \in S$ such that
 - $Q(v) = \exists$ and for all $(v, v', d) \in E$, we have $p_d = (s', S')$ with $v' \notin S'$.
 - $Q(v) = \forall$ and there is $(v, v', d) \in E$ such that $p_d = (s', S')$ with $v' \notin S'$.

We claim that $L_\infty(A) = \pi_1(Y)$.

Given a configuration $y \in Y$, we prove inductively the following statement: there is an accepting run starting from (i, v) on $\pi_1(y)$ for all $i \in \mathbb{Z}^2$ and $v \in \pi_2(y_i)$. If $Q(v) = \exists$, then Condition 3 ensures we find an edge (v, v', d) with $v' \in \pi_2(y_{i+d})$. Condition 1 and iterating this argument yields an accepting run starting from $(i + d, v')$. A similar argument works for \forall . Condition 2 ensures that there is an initial state in $\pi_2(y_i)$, so $\pi_1(y)$ is accepted by A .

Conversely, given a configuration $x \in L_\infty(A)$, we define $y_i = (x_i, S_i)$ where $S_i \subset V$ is the set of states v such that there is an accepting run on x from (i, v) . Since x admits an accepting run from any position for some initial state, it is easy to see that all three conditions above are satisfied and $y \in Y$. ◀

The proof of the following result is due to Ville Salo (personal communication).

► **Theorem 8.** $Alt \subsetneq Sofic$.

► **Definition 9.** Given a one-dimensional subshift $X \subset \Sigma^{\mathbb{Z}}$, its two-dimensional lift is defined as

$$X^\uparrow = \{y \in \Sigma^{\mathbb{Z}^2} : \exists x \in X, \forall i, j, y_{i,j} = x_i\}.$$

By the constructions of Aubrun-Sablik [2] or Durand-Romaschenko-Shen [5], the lift of any one-dimensional subshift given by a computable set of forbidden patterns is a sofic subshift.

► **Lemma 10.** If X^\uparrow is accepted by an alternating PWA, then X is sofic.

Proof. Let A be the automaton that accepts X^\uparrow in the sense that $X^\uparrow = L_\infty(A)$, and A' be the automaton obtained by replacing every direction \uparrow or \downarrow by \bullet in A .

Since every configuration in X^\uparrow is constant in the vertical direction, A' accepts every configuration in X^\uparrow (as well as additional configurations that are not constant vertically). Since A' only travels horizontally, it can be seen as a two-way one-dimensional automaton

that accepts exactly the one-dimensional configurations that lift to X^\uparrow ; in other words, A' accepts X . It follows that X is defined by a regular set of forbidden patterns, so it is sofic. ◀

To prove Theorem 8, see that the lift X^\uparrow of any non-sofic one-dimensional subshift X given by a computable set of forbidden patterns is sofic and not accepted by any alternating PWA. It is well-known that such subshifts exist; see e.g. [12, Example 3.1.7].

4 Alternating Hierarchy for Plane-walking Automata

In this paper, we are interested in comparing the power of plane-walking automata with varying access to nondeterminism. We introduce an alternating hierarchy of subshifts, similar to the classic alternating hierarchies of propositional logic formulæ. We are not able to prove that this forms a strict hierarchy of subshifts, but we show that the hierarchy does not collapse at the first level.

4.1 Definitions

Starting from Δ_1 (deterministic) PWA, we define Σ_1 , resp. Π_1 , PWA as the existential, resp. universal, PWA, that is, all states are labelled by \exists quantifiers, resp. \forall quantifiers. We call Σ_1 and Π_1 the corresponding classes of subshifts.

By definition, $\Delta_1 \subset \Sigma_1 \cap \Pi_1$. The rest of this paper is dedicated to show that Σ_1 and Π_1 are incomparable sets (and thus strictly larger than Δ_1). First, we define the whole hierarchy inductively by using automata decompositions.

► **Definition 11** (Decomposition). *A two-dimensional automaton $A = (V, E, \Sigma, D, I, Q)$ has a decomposition $(S, V \setminus S)$ for $S \subseteq V$ if there exists no path from $V \setminus S$ to S in E . By extension, we call decomposition the pair of induced subautomata.*

► **Definition 12** (Σ_n and Π_n). *An alternating plane-walking automaton is Σ_{n+1} if it admits a decomposition into a Σ_1 -automaton and a Π_n -automaton. We denote Σ_{n+1} the class of subshifts X such that $X = L_\infty(A)$ for a Σ_{n+1} -automaton.*

The definitions for Π_{n+1} are symmetrical.

Equivalently, a Σ_n automata is such that the image by Q of any path starting from I is a word of $\exists^* \forall^* \dots$ with n blocks ($n - 1$ alternations). This justifies the following definition:

► **Definition 13** (Δ_n). *An alternating plane-walking automaton is Δ_n if the image by Q of any path starting from I is a word of $\{\exists, \forall\}^*$ with $n - 1$ blocks ($n - 2$ alternations). We denote by Δ_n the class of subshifts X such that $X = L_\infty(A)$ for some Δ_n PWA A .*

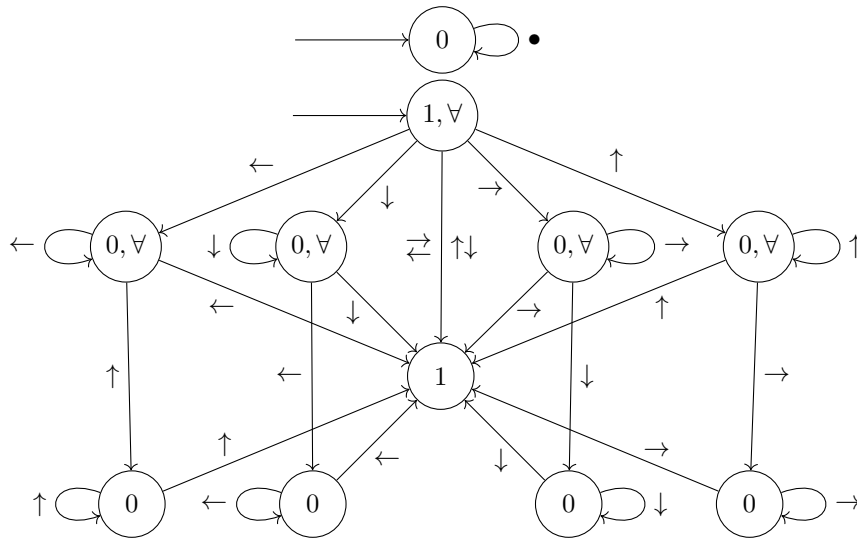
It is not difficult to see that $\Delta_n \subseteq \Sigma_n \subseteq \Delta_{n+1}$ and $\Delta_n \subseteq \Pi_n \subseteq \Delta_{n+1}$. We do not know whether $\Sigma_n \cap \Pi_n = \Delta_n$.

We are ready to state our main result:

► **Theorem 14.** $\Sigma_1 \not\subseteq \Pi_1$, and $\Pi_1 \not\subseteq \Sigma_1$. As a consequence, $\Delta_1 \subsetneq \Sigma_1$ and $\Delta_1 \subsetneq \Pi_1$.

In the next two subsections, we build two subshifts in $\Sigma_1 \setminus \Pi_1$ and $\Pi_1 \setminus \Sigma_1$, respectively. We begin by a technical lemma:

► **Lemma 15.** *Let $X \in \Sigma_n$ be a subshift and Y be a SFT. Then $X \cap Y \in \Sigma_n$. The same holds for Π and Δ .*



■ **Figure 2** A ∇ -automaton accepting X_{ssu} . A branch visiting the central state has no next move, so it rejects.

Proof. Given a PWA A , define A' as follows. From the initial state, check the area $\{\bullet, \uparrow, \rightarrow\}$ around the initial position. If a forbidden pattern is found, reject; this is done deterministically. Otherwise, come back to the initial position and execute a run of A . A' belongs to the same class as A and accepts all configurations in $L_\infty(A)$ where no forbidden pattern appears. ◀

4.2 Sunny Side Up

► **Definition 16** (Sunny side up). *The sunny side up subshift X_{ssu} is the set of configurations $x \in \{0, 1\}^{\mathbb{Z}^2}$ with at most one $i \in \mathbb{Z}^2$ such that $x_i = 1$.*

The sunny side up subshift can be easily accepted using a ∇ -automaton that has the ability of exploring an unbounded space and rejecting if any “problem” is found in any of the branches.

► **Proposition 17.** $X_{ssu} \in \Pi_1$.

Proof. Let A be the automaton represented in Figure 2. Every run starting from a position containing 0 accepts. Therefore every configuration with no 1 is accepted.

If A starts on a 1 at position i , it nondeterministically picks a quarter-plane to explore and rejects if this branch encounters another 1. A run never visits a given position twice, so if x contains a single 1, all runs accept.

If x contains two 1 at positions i and j , draw a path from i to j using at most two directions in $\{\rightarrow, \uparrow, \leftarrow, \downarrow\}$. This path yields a rejecting run starting from i . ◀

► **Proposition 18.** $X_{ssu} \notin \Sigma_1$.

The following proof is related to Proposition 1 in [14], which only holds for deterministic PWA (and a larger class of subshifts). Intuitively, Σ_1 -automata are ill-fitted to explore unbounded spaces, as a run may reject from a given starting point only if all branches reject, but every branch may only visit a small part of the space.

In the next proof, we use the notation $p \pm K = \{p \pm i : i \in K\}$ for $p \in \mathbb{Z}^2, K \subset \mathbb{Z}^2$.

Proof. Let A be a Σ_1 PWA with C states; we show that $X_{ssu} \neq L_\infty(A)$. For $\vec{v} \in \mathbb{Z}^2$ and $r \in \mathbb{N}$, we denote by $B^+(\vec{v}, r)$ the set $\{j \in \mathbb{Z}^2 : \exists k \in \mathbb{R}^+, d(j, k\vec{v}) \leq r\}$ (for the Manhattan distance on \mathbb{R}^2), and $B(\vec{v}, r) = B^+(\vec{v}, r) \cup -B^+(\vec{v}, r)$. In other words, this is the band of width r around the (half-)line of direction \vec{v} starting from 0; if $\vec{v} = 0$, $B(0, r) = B^+(0, r)$ is a ball of radius r . When r is not indicated, we assume that $r = C$.

Let A be a Σ_1 -automaton that accepts all configurations in X_{ssu} . Define $x \in X_{ssu}$ such that $x_i = 0$ for all i ; $y \in X_{ssu}$ such that $y_0 = 1$ and $y_i = 0$ for all other positions i ; and, for any $p \neq 0$, $z^p \notin X_{ssu}$ such that $z_0^p = z_p^p = 1$ and $z_i^p = 0$ for all other positions i . We find some p such that A accepts z^p , showing that A does not accept X_{ssu} .

To find some accepted z^p , we use the following property: if there is an accepting branch in x that visits neither 0 nor p , the same branch is accepting in z^p . Similarly, if an accepting branch in y does not visit p or an accepting branch in $\sigma^p(y)$ does not visit 0, then the same branch is accepting in z^p .

Let $S_1(r) \subset V \times B(0, r)$ be such that $(s, k) \in S_1$ if and only if there is an accepting run on y from (s, k) . For each $(s, k) \in S_1$, pick an arbitrary accepting branch b . It is described by a sequence $(s_n, p_n)_{n \in \mathbb{N}} \in (V \times \mathbb{Z}^2)^\mathbb{N}$. If b stays in $B(0)$, we find two steps $i < j$ such that $(s_i, p_i) = (s_j, p_j)$ and, by a pumping argument, we build another accepting branch b_p which is eventually periodic and stays in $B(0)$. If b leaves $B(0)$, we find two steps $i < j$ such that $s_i = s_j$ and $d(0, p_j) \geq d(0, p_i)$ and, by a pumping argument, we build another accepting branch b_p which is eventually periodic up to translation by the pumping vector $v^{(s,k)} = p_j - p_i$ every $j - i$ steps; in other words, b_p stays in some band $B(v^{(s,k)})$. Denote $P_1(r) = \{v^{(s,k)} : (s, k) \in S_1(r)\}$ the set of all such pumping vectors.

Let $S_0 \subset V$ be the set of states reachable from i_0 through a path labelled by 0. Since x is accepted, there must be a cycle that stays in S_0 . For any such simple cycle (without repeated states), the associated pumping vector is the sum of all directions. Denote P_0 the finite set of all such pumping vectors.

We distinguish two cases that we illustrate in Figure 3.

All vectors in P_0 are colinear to some v

We choose p such that $p \notin B(v) \cup \bigcup_{v_1 \in P_1(0)} B(v_1)$. This avoids a finite set of one-dimensional subsets, so such a choice is possible.

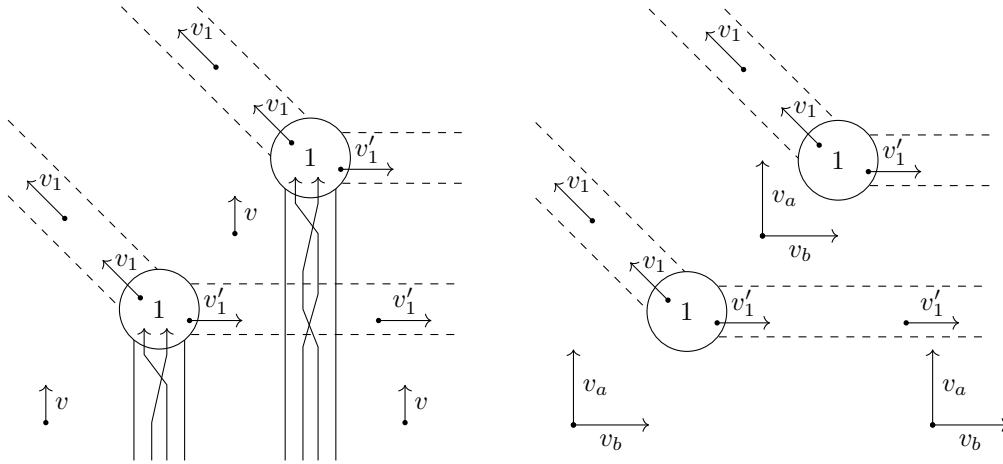
Take any position $k \in \mathbb{Z}^2$ and assume that $k \notin p - B^+(v)$. Pick an accepting branch b in y from k . As long as b never visits the 1 in position 0, it stays in a state of S_0 , so we can pump to build a periodic accepting branch that stays in $k + B^+(v)$ and does not visit p . If b visits 0, then it is in some state s such that $(s, 0) \in S_1(0)$, and so we can pump to build another accepting branch b_p that stays in $B(0)$ or in $B^+(v_1)$ for some $v_1 \in P_1(0)$. Either way, we built an accepting branch in y from k that does not visit p , so it is accepting in z^p .

If $k \in p - B^+(v)$, then $k \notin 0 - B^+(v)$ since we chose p in such a way that the sets are disjoint. With a similar argument, we build an accepting branch that does not visit 0 on $\sigma^p(y)$.

Every starting position in z^p has an accepting branch, so z^p is accepted by A .

There are two noncolinear vectors v_a, v_b in P_0

This means that, in x , the accepting run from $(i_0, 0)$ has two accepting branches that stay in $B^+(v_a)$ and $B^+(v_b)$, respectively. Since v_a and v_b are not colinear, there is $r > 0$ large enough that $B(v_a) \cap B(v_b) \subset B(0, r)$.



■ **Figure 3** Left: the first case when all vectors in P_0 are colinear to v . Right: the second case when $\{v_a, v_b\} \subset P_0$. In both cases, $P_1 = \{v_1, v'_1\}$ and the circles represent positions 0 and p . The proof shows that each initial position admits a path that never visits 0 or p and accepts in y or $\sigma^p(y)$.

We choose p such that $p \notin \bigcup_{v_1 \in P_1} B(v_1, r + C)$ and such that p is in the quarterplane generated by v_a, v_b , at distance at least $r + 2C$ from the border.

Take a position $k \in \mathbb{Z}^2$. If $k \in B(0, r)$, then we pump to build an accepting branch b_p in y that stays either in $B(0)$ or in $k + B^+(v_1)$ for some $v_1 \in P_1(r)$, so that b_p does not visit p . If $k \in B(p, r)$, the same argument applies on $\sigma^p(y)$.

If $k \notin B(0, r) \cup B(p, r)$, then one of the two bands $B_a = k + B^+(v_a)$ and $B_b = k + B^+(v_b)$ contains neither 0 nor p . Indeed,

- $B_a \cap B_b$ contains neither position by definition of r ;
- If both positions appeared, we would have $|p - 0 - \pm(av_a - bv_b)| \leq 2C$ for some $a, b \geq 0$, which contradicts the choice of p .

Assuming that it is B_a , we pump to build an accepting branch in x from k that stays in B_a and visits neither 0 nor p .

Every position in z^p has an accepting branch, so z^p is accepted by A . ◀

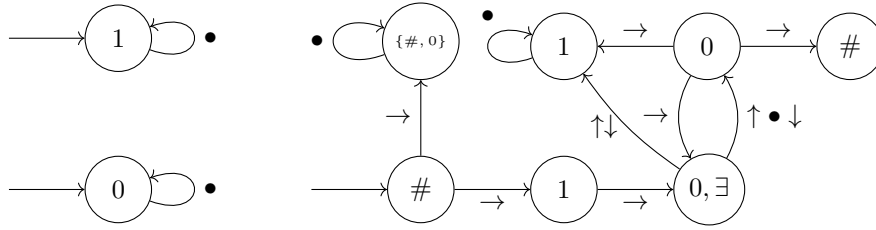
4.3 The Cone Labyrinth

► **Definition 19.** Let $\Sigma = \{0, 1, \#\}$. The cone labyrinth subshift (denoted X_{cl}) is the set of configurations x which contains none of the forbidden patterns $010, 11, \#1\#, \begin{smallmatrix} 0 \\ \# \end{smallmatrix}, \begin{smallmatrix} 1 \\ \# \end{smallmatrix}, \begin{smallmatrix} \# \\ 0 \end{smallmatrix}, \begin{smallmatrix} \# \\ 1 \end{smallmatrix}$ and such that from any position with a pattern $\#1$, there is a path to a position with a pattern $1\#$ using steps $\nearrow, \rightarrow, \searrow$ that only visits positions with symbols 0.

In a configuration $x \in \{0, 1, \#\}^{\mathbb{Z}^2}$ of X_{cl} , every column contains either only $\#$ symbols, or only 0 and 1 symbols. The $\#$ marks the outside areas, the 0 the inside areas, and 1 corresponds to entrances / exits to change areas. In particular, a 1 can only be between a 0 and a $\#$. Furthermore, every entrance $\#1$ can be matched to at least one exit $1\#$, in the sense that one can walk from $\#1$ to $1\#$ through zeroes using steps $\nearrow, \rightarrow, \searrow$.

In other words, if the width of the inside area is k , then every entrance must be matched to an exit at a vertical distance at most k .

► **Proposition 20.** $X_{cl} \in \Sigma_1$.



■ **Figure 4** \exists -automaton accepting X_{cl} . We assumed for readability that the patterns from Definition 19 have already been forbidden.

Proof. We build a \exists automaton that accepts X_{cl} assuming that patterns from Definition 19 do not appear; we can do this assumption by Lemma 15. The automaton is represented in Figure 4.

If the initial position is not the entrance of a labyrinth $\#1$, accept (by looping). Otherwise, walk nondeterministically in all directions $\nearrow, \rightarrow, \searrow$. Keep going as long as you see 0, accept if you find a 1, reject if you find a $\#$. Therefore the automata accepts if and only if, starting from every entrance, one branch found a matching exit. ◀

► **Proposition 21.** $X_{cl} \notin \Pi_1$.

Intuitively, in order for a run to reject a labyrinth with no exit, some individual branch must reject, which requires visiting a region of unbounded size (depending on the width on the labyrinth). By making the width large enough, we disorient this run into rejecting a valid configuration because it cannot check all required cells.

Proof. Let A be a Π_1 -automaton that rejects all configurations not in X_{cl} . We build a configuration $y \in X_{cl}$ that A rejects, the process being illustrated in Figure 5.

First define a configuration $x^n \notin X_{cl}$ for some $n > |Q|$.

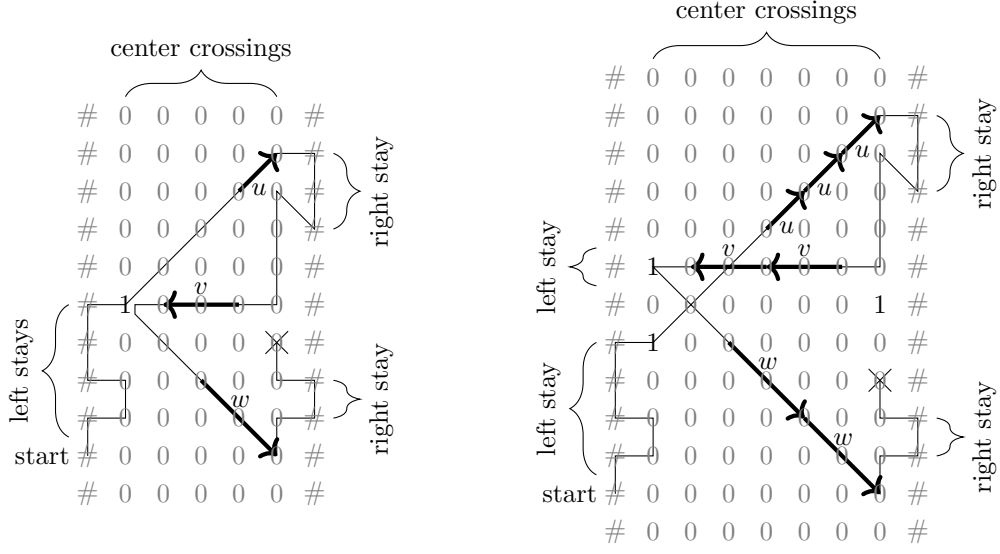
$$x^n_{(i,j)} = \begin{cases} 1 & \text{if } (i, j) = (0, 0) \\ 0 & \text{otherwise, if } 0 \leq i \leq n \\ \# & \text{otherwise} \end{cases}$$

Since A rejects x^n , there exists a position from which there is no accepting run of A ; that is, some branch b from some position p rejects (in finite time). The configuration would be valid if we switched the symbols at positions $(0, 0)$ or (n, k) for any $-n \leq k \leq n$; therefore b must visit all these positions, since it would otherwise reject a valid configuration.

Within x^n , we call “the left” the half-plane $i \leq 0$, “the right” the half-plane $i \geq n$, and “the center” the band $0 < i < n$. For simplicity, we assume that b starts in the left and ends in the right. We decompose b as $\ell_0 c_0^{\ell \rightarrow r} r_0 c_0^{r \rightarrow \ell} \dots \ell_T c_T^{\ell \rightarrow r} r_T$, where for all $k = 0, \dots, T$:

- every subsequence is nonempty;
- ℓ_k starts in the left, stays in the left and center, and ends in the left (*left stays*);
- r_k starts in the right, stays in the right and center, and ends in the right (*right stays*);
- $c_k^{\ell \rightarrow r}$ and $c_k^{r \rightarrow \ell}$ stay in the center (*center crossings*).

Consider the first crossing $c_0^{\ell \rightarrow r}$. Crossing takes at least n steps, so we find two positions (i, j) and $(i + a_0, j + b_0)$ with $a_0 > 0$ that $c_0^{\ell \rightarrow r}$ visited in that order in the same state; (a_0, b_0) is the pumping vector. Similarly, we find such pumping vectors (a_k, b_k) with $a_k > 0$ for all $c_k^{\ell \rightarrow r}$ and (α_k, β_k) with $\alpha_k < 0$ for all $c_k^{r \rightarrow \ell}$.



■ **Figure 5** On the left, a rejecting run over the invalid labyrinth x^4 , which has no exit. Bolded parts \vec{u} , \vec{v} and \vec{w} represent pumping vectors which start and end in the same state. These vectors are repeated on the right to provide another rejecting run of the same automata over the valid labyrinth y .

By pumping on these vectors on each crossing, we build a branch b' that will be valid on some other configuration $y \neq x^n$; for the time being, we only pay attention to the positions in the branch and not to the underlying configuration.

Let C be the lowest common multiple of a_k and $-\alpha_k$ and let $m > 0$ to be fixed later. We define $b' = \ell'_0 c_0^{\ell \rightarrow r} r'_0 c_0^{r \rightarrow \ell} \dots \ell'_T c_T^{\ell \rightarrow r} r'_T$ step by step.

- $\ell'_0 = \ell_0$ is unchanged;
- $c_0^{\ell \rightarrow r}$ is $c_0^{\ell \rightarrow r}$ that we pump $\frac{mC}{a_0}$ times along the vector (a_0, b_0) .
- $r'_0 = \sigma^{(mC, \frac{mC}{a_1} b_1)}(r_0)$; in other words, r'_0 is shifted relative to r_0 so that the positions are consistent with the previous part.
- $c_0^{r \rightarrow \ell}$ is $c_0^{r \rightarrow \ell}$ shifted by $(mC, \frac{mC}{a_1} b_1)$ and pumped $\frac{mC}{-\alpha_0}$ times along the vector (α_0, β_0) .

In a similar manner, we pump every crossing in the center and shift:

- ℓ'_k and $c_k^{\ell \rightarrow r}$ by $(0, o_k^\ell(m))$, where $o_k^\ell(m) = \sum_{i=0}^{k-1} \frac{mC}{a_i} b_i + \sum_{i=0}^{k-1} \frac{mC}{-\alpha_i} \beta_i$;
- r'_k and $c_k^{r \rightarrow \ell}$ by $(mC, o_k^r(m))$ where $o_k^r(m) = \sum_{i=0}^k \frac{mC}{a_i} b_i + \sum_{i=0}^{k-1} \frac{mC}{-\alpha_i} \beta_i$.

These values are chosen so that the positions are locally consistent with allowed transitions in A . Notice that $o_k^\ell(m) - o_{k'}^\ell(m)$ is either always 0 or tends to $\pm\infty$ as $m \rightarrow \infty$.

Denoting φ the footprint, we see that $\varphi(\ell'_k) = \varphi(\ell_k) + o_k^\ell(m)$. Because every $\varphi(\ell_k)$ is finite, we can find m large enough such that, for all k and k' , $\varphi(\ell'_k) \cap \varphi(\ell'_{k'}) = \emptyset$ or $\varphi(\ell'_k) \cap \varphi(\ell'_{k'}) = \varphi(\ell_k) \cap \varphi(\ell_{k'})$, and similarly for right stays. In other words, two stays either visit no cell in common or they cross in exactly the same manner as the original branch.

If needed, we increase m so that $m > |\varphi(b)|$.

Now we build a configuration y so that b' is a branch of the run starting at the same position p . Start by putting $y = x^{n+mC}$, and do the following modifications:

1. set $y_{(0, o_k^\ell(m))} = 1$ for all k ;
2. choose some i such that $|i| < n + Cm$ and $(n + Cm, i) \notin \bigcup_k \varphi(r'_k)$. Then set $y_{(n, i + o_k^\ell(m))} = 1$ for all k .

Condition 1 adds entrances at such positions that the left stay ℓ'_k “sees” an entrance exactly when the original left stay ℓ_k saw an entrance at position $(0, 0)$. This does not impact other left stays by the argument above.

Condition 2 adds exits at positions that are not visited by b' , which is possible because we chose m to be larger than the footprint of b , but satisfy the definition for a valid labyrinth. This ensures that $y \in X_{cl}$ and that b' rejects.

By construction, the branch b' is a branch for the run on y starting at p . It follows that $y \in X_{cl}$ is rejected by A , a contradiction. ◀

5 Conclusion

5.1 Summary and open questions

We proved that subshifts accepted by alternating plane-walking automata form a strict subset of sofic subshifts, and that the first level of the hierarchy with bounded alternations is strict: Σ_1 and Π_1 are incomparable, and thus the inclusion of Δ_1 in both of them is strict.

We sum up our open questions:

1. Is the hierarchy strict for all n or does it collapse at some level? For example, can we find a subshift in $\Sigma_2 \setminus \Sigma_1$?
2. If a subshift is accepted by a universal PWA and an existential PWA, is it also accepted by a deterministic PWA? More generally, is it the case that $\Delta_n = \Sigma_n \cap \Pi_n$?
3. Is there an equivalent definition for subshifts accepted by alternating plane-walking automata by forbidden patterns accepted by plane-walking automata that do not require taking a complement?

Pumping arguments are tedious even in the first floors of the hierarchy, and we would like to find other tools, for example (as suggested by Guillaume Theyssier and inspired by [17]) from communication complexity.

This is only one of multiple possible hierarchies on walking automata. We mention n -nested automata in the next section. Automata with the ability to leave up to n pebbles (non-moving marks used as memory) during a run have also been considered. Salo and Törma studied automata with multiple heads and this hierarchy collapses at $n = 3$ [14, 13].

Definitions 2 and 4 (recognised / accepted subshifts) extend to higher dimensions \mathbb{Z}^d , $d \geq 2$, or any finitely generated group by replacing the set of directions by S and $S \cup S^{-1} \cup \{\bullet\}$, respectively, where S is a set of generators of the group. While our main results extend directly to \mathbb{Z}^d by considering lifts of our two-dimensional examples (see Definition 9), we make no guesses as to the situation in more complicated groups.

5.2 Strict Hierarchy and Tree-walking Automata

► **Conjecture 22.** *The hierarchy is strict, that is, $\Sigma_n \subsetneq \Sigma_{n+1}$ for all n (and the same is true for all combinations of Π, Σ and Δ).*

We present some elements supporting this conjecture. Tree-walking automata (TWA) is a similar model that recognise words written on (finite or infinite) trees. For example, Theorem 8 can be seen as the translation of [4].

In the context of tree-walking automata, we could not find any work on a similar Σ/Π hierarchy. However, [16] considers k -nested TWA, defined intuitively as follows: 0-nested TWA are (existential) nondeterministic TWA; k -nested TWA are nondeterministic TWA where, at each step, the set of available transitions is determined by foreseeing the behaviour of two $k - 1$ -nested TWA A and \bar{A} , in the following sense: the next transition is chosen nondeterministically among transitions after which A would accept and \bar{A} would reject.

The class of (tree-walking or plane-walking) k -nested automata seems to be related to Σ_{k+1} automata, although we do not have a proof of either direction. For tree-walking automata, Theorem 29 in [16] proves that the hierarchy of languages recognized by k -nested TWA is strict. This is to us a strong indication that the alternating hierarchy is strict on trees (free groups).

We believe that these results can be brought to plane-walking automata by considering subshifts where trees are drawn on a background on zeroes (this can be ensured with finitely many forbidden patterns), and every tree must belong to L_k , where L_k is a tree language that is Σ_{k+1} but not Σ_k (alternatively, k -nested and not $k-1$ -nested). This is not straightforward as plane-walking automata have the ability to walk out of the tree, which should not provide additional recognition power, but pumping arguments are very tedious for alternating plane-walking automata. We leave this as an open question for future research.

5.3 An Alternative Approach: Kari-Moore's Rectangles

We mention an alternative approach to prove that $\Sigma_1 \neq \Pi_1$ that was used in [10] for finite patterns. The following statements are only translations to our model and from finite patterns to periodic configurations, and we do not vouch for their correctness. This is another suggestion for future work generalising Theorem 14.

Let $f : \mathbb{N} \rightarrow \mathcal{P}(\mathbb{N})$. Let $X_f \subset \{0, 1\}^{\mathbb{Z}^2}$ be the smallest subshift containing the strongly periodic configurations generated by the rectangles:

$$\{r \in \{0, 1\}^{[0, n] \times [0, k]} : k \in f(n) \text{ and for all } i, j > 0, r_{0,0} = r_{i,0} = r_{0,j} = 1, r_{i,j} = 0\}.$$

If X_f is accepted by a Σ_1 plane-walking automaton with k states, then for every n , the language $\{1^j : j \in f(n)\}$ is recognised by a two-way nondeterministic finite automaton with kn states, and hence regular [10].

The following example of a function f is such that $X_f \in \Sigma_1 \setminus \Pi_1$, while $X_{f^c} \in \Pi_1 \setminus \Sigma_1$:

$$f(n) = \{in + j : i, j \in \mathbb{N}, j < i\}.$$

Indeed, $f^c(n)$ is a finite set whose largest element is $(n-2) \cdot n + (n-1) \approx n^2$. If it is accepted by an automaton with kn states, this would be a contradiction for $n > k$ by pumping.

References

- 1 Marcella Anselmo and Maria Madonia. Classes of two-dimensional languages and recognizability conditions. *RAIRO Theor. Informatics Appl.*, 44(4):471–488, 2010. doi:10.1051/ita/2011003.
- 2 Nathalie Aubrun and Mathieu Sablik. Simulation of effective subshifts by two-dimensional subshifts of finite type. *CoRR*, abs/1602.06095:35–63, 2016. doi:10.48550/arXiv.1602.06095.
- 3 Manuel Blum and Carl Hewitt. Automata on a 2-dimensional tape. In *8th Annual Symposium on Switching and Automata Theory, Austin, Texas, USA, October 18-20, 1967*, pages 155–160. IEEE, IEEE Computer Society, 1967. doi:10.1109/F0CS.1967.6.
- 4 Mikolaj Bojanczyk and Thomas Colcombet. Tree-walking automata do not recognize all regular languages. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 234–243. ACM, 2005. doi:10.1145/1060590.1060626.
- 5 Bruno Durand, Andrei Romashchenko, and Alexander Shen. Fixed-point tile sets and their applications. *Journal of Computer and System Sciences*, 78(3):731–764, 2012. doi:10.1016/j.jcss.2011.11.001.

- 6 Dora Giammarresi and Antonio Restivo. Recognizable picture languages. *International Journal of Pattern Recognition and Artificial Intelligence*, 6(2&3):241–256, 1992. doi:10.1142/S021800149200014X.
- 7 Dora Giammarresi and Antonio Restivo. Two-dimensional languages. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages, Volume 3: Beyond Words*, pages 215–267. Springer, 1997. doi:10.1007/978-3-642-59126-6_4.
- 8 Nataša Jonoska and Joni B. Pirnot. Finite state automata representing two-dimensional subshifts. *Theoretical computer science*, 410(37):3504–3512, 2009. doi:10.1016/j.tcs.2009.03.015.
- 9 Jarkko Kari and Christopher Moore. New results on alternating and non-deterministic two-dimensional finite-state automata. In Afonso Ferreira and Horst Reichel, editors, *Annual Symposium on Theoretical Aspects of Computer Science*, volume 2010 of *Lecture Notes in Computer Science*, pages 396–406. Springer, 2001. doi:10.1007/3-540-44693-1_35.
- 10 Jarkko Kari and Christopher Moore. Rectangles and squares recognized by two-dimensional automata. In Juhani Karhumäki, Hermann A. Maurer, Gheorghe Paun, and Grzegorz Rozenberg, editors, *Theory is Forever: Essays Dedicated to Arto Salomaa on the Occasion of His 70th Birthday*, volume 3113 of *Lecture Notes in Computer Science*, pages 134–144. Springer, 2004. doi:10.1007/978-3-540-27812-2_13.
- 11 Jarkko Kari and Ville Salo. A survey on picture-walking automata. In Werner Kuich and George Rahonis, editors, *Algebraic Foundations in Computer Science – Essays Dedicated to Symeon Bozapalidis on the Occasion of His Retirement*, volume 7020 of *Lecture Notes in Computer Science*, pages 183–213. Springer, 2011. doi:10.1007/978-3-642-24897-9_9.
- 12 Douglas Lind and Brian Marcus. *An introduction to symbolic dynamics and coding*. Cambridge University Press, 1995.
- 13 Ville Salo. Four heads are better than three. In Hector Zenil, editor, *Cellular Automata and Discrete Complex Systems – 26th IFIP WG 1.5 International Workshop, AUTOMATA 2020, Stockholm, Sweden, August 10-12, 2020, Proceedings*, volume 12286 of *Lecture Notes in Computer Science*, pages 111–125. Springer, 2020. doi:10.1007/978-3-030-61588-8_9.
- 14 Ville Salo and Ilkka Törmä. Plane-walking automata. In Teijiro Isokawa, Katsunobu Imai, Nobuyuki Matsui, Ferdinand Peper, and Hiroshi Umeo, editors, *Cellular Automata and Discrete Complex Systems – 20th International Workshop, AUTOMATA 2014, Himeji, Japan, July 7-9, 2014, Revised Selected Papers*, volume 8996 of *Lecture Notes in Computer Science*, pages 135–148. Springer, 2014. doi:10.1007/978-3-319-18812-6_11.
- 15 Ville Salo and Ilkka Törmä. Group-walking automata. In Jarkko Kari, editor, *Cellular Automata and Discrete Complex Systems: 21st IFIP WG 1.5 International Workshop, AUTOMATA 2015, Turku, Finland, June 8-10, 2015. Proceedings 21*, volume 9099 of *Lecture Notes in Computer Science*, pages 224–237. Springer, 2015. doi:10.1007/978-3-662-47221-7_17.
- 16 Balder ten Cate and Luc Segoufin. Transitive closure logic, nested tree walking automata, and xpath. *J. ACM*, 57(3):18:1–18:41, 2010. doi:10.1145/1706591.1706598.
- 17 Véronique Terrier. Communication complexity tools on recognizable picture languages. *Theoretical Computer Science*, 795:194–203, 2019. doi:10.1016/j.tcs.2019.05.040.