# Multidimensional tilings and MSO logic

Rémi PALLEN[1][0009−0006−6708−6904] and Ilkka TÖRMÄ[2][0000−0001−5541−8517]

[1] ENS Paris-Saclay, Gif-sur-Yvette, France remi.pallen@loria.fr
[2] Department of Mathematics and Statistics, University of Turku, Turku, Finland
iatorm@utu.fi

**Abstract.** We define sets of coulourings of the infinite discrete plane using monadic second order (MSO) formulas. We determine the complexity of deciding whether such a formula defines a subshift, parametrized on the quantifier alternation complexity of the formula. We also study the complexities of languages of MSO-definable sets, giving either an exact classification or upper and lower bounds for each quantifier alternation class.

**Keywords:** MSO logic · subshifts · symbolic dynamics · tilings

## 1 Introduction

A *tiling* or *configuration* is a colouring of the two-dimensional plane $\mathbb{Z}^2$ with finitely many colours. A *subshift* is the set of tilings such that no pattern from a set of *forbidden patterns* appears. These notions were first introduced by Wang [10] to study first order logic, and to find an algorithm which computes whether a formula is a tautology or not. Berger proved in 1966 [1] the undecidability of the domino problem, and many other problems were proved undecidable too thanks to this result in the following decades.

This article continues the line of research in [3,9]. The idea is to define sets of configurations using *monadic second order* (*MSO* for short) logic. MSO definability is a classical and broad area of research; see [8] for an overview in the context of finite and infinite languages. MSO formulas on finite words correspond to regular languages, and those on infinite words correspond to $\omega$-regular languages. Subshifts can be seen as languages of two-dimensional infinite words, so it makes sense to ask which of them can be defined by MSO formulas. In the formalism of [3], an MSO formula always defines a translation invariant set, but if it has a first order existential quantifier, it may not define a topologically closed set, i.e. a subshift. This is why [3,9] mostly restrict their study to the case where the MSO formulas do not have first order existential quantifiers. In this article, we study the case where they are allowed.

Our first result is to determine the exact complexity of deciding whether an MSO formula defines a subshift or not, depending on the complexity (number of second-order quantifier alternations) of the formula. We also study the complexity of languages of subshifts and sets these formulas can define.

One motivation for this research is *Griddy* (formerly called Diddy), a Python library for discrete dynamical systems research developed by Ville Salo and the second author [7,6]. Griddy allows the definition and manipulation of multidimensional subshifts defined by first order logical formulas. Even though most properties of multidimensional subshifts are undecidable, partial algorithms can often give good solutions or approximations in "naturally occurring" cases. This raises the question whether the same holds for subshifts defined by MSO formulas, or whether the complexity is too high to obtain even partial algorithms.

## 2 Preliminaries

### 2.1 Subshifts

We review some basic facts from multidimensional symbolic dynamics. See [5], especially Section A.6, for a more comprehensive overview.

Let $\mathcal{A}$ be a finite set of colours called *alphabet*. A *pattern* is a map $P : D \to \mathcal{A}$ where $D \subset \mathbb{Z}^2$. $D$ is called *support* of the pattern and is denoted $D(P)$. A *configuration* is a pattern $x$ such that $D(x) = \mathbb{Z}^2$. We say that a pattern $P$ *appears* in a configuration $x \in \mathcal{A}^{\mathbb{Z}^2}$ if there exists $\boldsymbol{v} \in \mathbb{Z}^2$ such that $P_{\boldsymbol{u}} = x_{\boldsymbol{u}+\boldsymbol{v}}$ holds for all $\boldsymbol{u} \in \mathrm{supp}(P)$. The number of occurrences of $P$ in $x$ (which may be infinite) is denoted $\#_x(P)$. A *cell* or *position* is an element $\boldsymbol{v} \in \mathbb{Z}^2$.

For a set of patterns $\mathcal{F}$, we define $X_{\mathcal{F}} \subseteq \mathcal{A}^{\mathbb{Z}^2}$ as the set of those $x \in \mathcal{A}^{\mathbb{Z}^d}$ in which no pattern from $\mathcal{F}$ appears. Such a set is called a *subshift*. When $\mathcal{F}$ is finite, $X_{\mathcal{F}}$ is a *subshift of finite type (SFT)*.

*Property 1.* For each Turing machine $M$, there exists a SFT (see Figure 1 for the colours, a pattern of size $1 \times 2$ or $2 \times 1$ is forbidden if the tiles do not match), such that the *initial tile* may appear in a tiling if and only if $M$ does not halt on the empty input. There exists a slightly different version for which an input is written on the computation tape. In this case, if the initial tile appears on a configuration with a finite input $x$ written on it, $M$ does not halt on $x$.

For $\boldsymbol{v} \in \mathbb{Z}^2$ and $x \in \mathcal{A}^{\mathbb{Z}^2}$, denote by $\sigma_{\boldsymbol{v}}(x)$ the configuration $\sigma_{\boldsymbol{v}}(x)_{\boldsymbol{u}} = x_{\boldsymbol{u}+\boldsymbol{v}}$ for all $\boldsymbol{u} \in \mathbb{Z}^2$. The function $\sigma_{\boldsymbol{v}} : \mathcal{A}^{\mathbb{Z}^2} \to \mathcal{A}^{\mathbb{Z}^2}$ is called the *shift by* $\boldsymbol{v}$. A set of configurations $E$ is *shift-invariant* if $\sigma_{\boldsymbol{v}}(x) \in E$ for all $x \in E$ and $\boldsymbol{v} \in \mathbb{Z}^2$.

We give a distance function $d$ on the space $\mathcal{A}^{\mathbb{Z}^2}$. For two configurations $x \neq y$, let $d(x, y) = 2^{-\min\{|a|+|b| \mid x_{(a,b)} \neq y_{(a,b)}\}}$ and $d(x, x) = 0$ for all $x$. The set $\mathcal{A}^{\mathbb{Z}^2}$ is therefore a metric space and has a corresponding topology, which is generated by the clopen *cylinder sets* $[P] = \{x \in \mathcal{A}^{\mathbb{Z}^2} \mid \forall \boldsymbol{v} \in \mathrm{supp}(P)\ x_{\boldsymbol{v}} = P_{\boldsymbol{v}}\}$ for finite patterns $P$.

*Property 2.* A set of configurations $X$ is a subshift if and only if it is shift-invariant and topologically closed.

The *language* $\mathcal{L}(X)$ of a set of configurations $X$ is the set of finite patterns $P$ such that $P$ appears in at least one $x \in X$.

*Property 3.* If $X$ and $Y$ are subshifts such that $\mathcal{L}(X) = \mathcal{L}(Y)$, then $X = Y$.

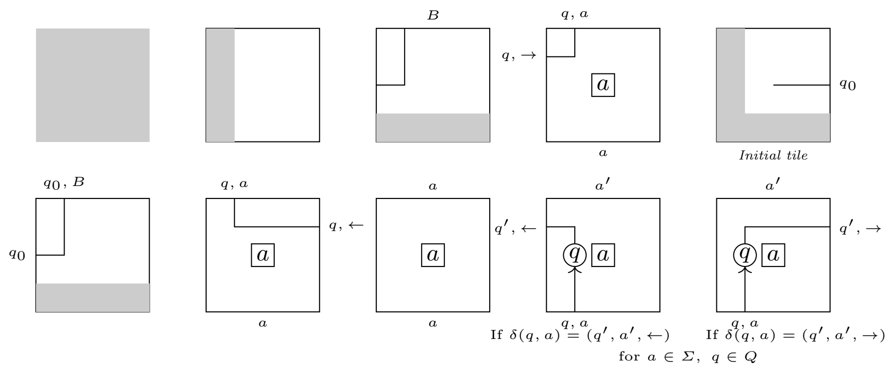


**Fig. 1.** Computation tiles

### 2.2 Computability

A decision problem is a predicate with some free variables (first or second order). The *arithmetic hierarchy* for decision problems is defined inductively as follows:

- $\Sigma_0^0$ and $\Pi_0^0$ are the set of decidable problems.
- $R$ is in $\Pi_{n+1}^0$ if there exists $S \in \Sigma_n^0$ such that $R \Leftrightarrow \forall x\ S(x)$.
- $R$ is in $\Sigma_{n+1}^0$ if there exists $S \in \Pi_n^0$ such that $R \Leftrightarrow \exists x\ S(x)$.

Here, quantifiers range over natural numbers. We define in a simililar way the analytical hierarchy:

- $\Sigma_0^1$ and $\Pi_0^1$ are the set of problems in the arithmetical hierarchy.
- $R$ is in $\Pi_{n+1}^1$ if there exists $S \in \Sigma_n^1$ such that $R \Leftrightarrow \forall_1 X\ S(X)$.
- $R$ is in $\Sigma_{n+1}^1$ if there exists $S \in \Pi_n^1$ such that $R(x) \Leftrightarrow \exists_1 X\ S(X)$.

Here, $\forall_1$ and $\exists_1$ quantify over sets of natural numbers. We often write simply $\exists$ and $\forall$ with a capital letter variable when the context is clear.

We use *many-one reductions* to compare the computational complexity of decision problems. We denote $Q \leq P$ is there exists a computable function $f : \mathbb{N} \to \mathbb{N}$ such that $Q(x) \Leftrightarrow P(f(x))$ for all inputs $x$. For a class $C$ and a problem $P$, if $Q \leq P$ holds for all $Q \in C$, then $P$ is $C$-*hard*. If $P$ is in $C$ and $C$-hard, we say that it is $C$-*complete*.

### 2.3 MSO Logic

We now review the formalism used in [3,9] for defining subshifts (and, more generally, shift-invariant sets of configurations) using logical formulas. Let $\mathcal{A}$ be a finite set of colours. A *term* is either a first order variable or one of East($t$), West($t$), North($t$) and South($t$), where $t$ is a term. An *atomic formula* is $t = t'$ or $P_c(t)$, where $t$ and $t'$ are terms and $c \in \mathcal{A}$ is a colour. A *monadic second order* (*MSO* for short) formula is constructed from terms using the logical connectives

$\neg$, $\wedge$ and $\vee$, plus first and second order quantification. In general, we will use uppercase letters for second order variables and lowercase letters for first order variables. Second order quantification is sometimes denoted $\forall_1$ and $\exists_1$ for clarity. The first order variables will range over $\mathbb{Z}^2$ and the second order variables over subsets of $\mathbb{Z}^2$. The *radius* of a formula is the maximal depth of terms $t$ in it. A formula is *closed* if all variables inside the formula are bound by a quantifier. It is *first order* (*FO* for short) if it contains no second order quantifier.

*Remark 1.* We concentrate on MSO formulas of the form $Q_1X_1Q_2X_2\cdots Q_nX_n\psi$, where the $Q_i$ are second order quantifiers and $\psi$ is first order, in order to discuss its number of second order quantifier alternations. A formula with mixed quantifier order can always be put into this form by replacing each out-of-order first order variable with a second order variable constrained to be a singleton. For example, $\forall x\ \exists Y\ \psi$ with $\psi$ first order becomes $\forall X\ \exists Y\ \neg(\exists n\ \forall m\ m \in X \Leftrightarrow m = n) \vee (\exists x\ x \in X \wedge \psi)$.

We define a hierarchy called *MSO hierarchy* of the complexity of formulas:

- $\bar{\Sigma}_0$ and $\bar{\Pi}_0$ are the set of first order formulas.
- $\phi \in \bar{\Sigma}_{n+1}$ if it is of the form $\exists X_1\ \ldots \exists X_n\ \psi$ with $\psi \in \bar{\Pi}_n$.
- $\phi \in \bar{\Pi}_{n+1}$ if it is of the form $\forall X_1\ \ldots \forall X_n\ \psi$ with $\psi \in \bar{\Sigma}_n$.

For any configuration $x \in \mathcal{A}^{\mathbb{Z}^2}$, we consider the model $\mathcal{M}_x = (\mathbb{Z}^2, \tau)$, whose domain is $\mathbb{Z}^2$ and whose signature $\tau$ contains

- four unary functions East, West, North and South, whose interpretations are $\mathrm{East}^{\mathcal{M}_x}(a,b) = (a+1,b)$, $\mathrm{West}^{\mathcal{M}_x}(a,b) = (a-1,b)$, $\mathrm{North}^{\mathcal{M}_x}(a,b) = (a,b+1)$ and $\mathrm{South}^{\mathcal{M}_x}(a,b) = (a,b-1)$, and
- a unary predicate symbol $P_c$ for each $c \in \mathcal{A}$, with the interpretation $P_c^{\mathcal{M}_x}(\boldsymbol{v}) = \top$ if and only if $x_{\boldsymbol{v}} = c$.

For a formula $\phi$, we write $x \vDash \phi$ if $\mathcal{M}_x \vDash \phi$, that is, $\phi$ holds in the model $\mathcal{M}_x$. For each MSO formula $\phi$, denote $X_\phi = \{x \in \mathcal{A}^{\mathbb{Z}^2} \mid x \vDash \phi\}$. The set $X_\phi$ is always shift-invariant but not always closed, and so not always a subshift (see Example 3). For a set of configurations $X$, if $X = X_\phi$ for some $\phi$, we say that $X$ is *MSO-definable*, and *C-definable* for a class of formulas $C$ if $\phi \in C$.

Every SFT is definable by an MSO formula of the form $\forall v\ \psi$, where $\psi$ is quantifier-free. This is essentially the format used by Griddy [7], with syntactic support for easily defining complex quantifier-free formulas.

If $X_\phi = X_\psi$, we say $\phi$ and $\psi$ are *equivalent*. If $Q$ is a quantifier, then $\bar{Q} = \forall$ if $Q = \exists$, $\bar{Q} = \exists$ otherwise. We say that a quantifier $Q$ in a formula $\phi$ is *irrelevant* if $\phi$ is equivalent to the formula where that quantifier has been replaced by $\bar{Q}$.

Our second order variables range over subsets of $\mathbb{Z}^2$. It is equivalent to quantify over configurations in the following sense. Consider a formula $\exists X \in X_\phi\ \psi$, where $X$ ranges over configurations of $X_\phi$ and $\psi$ may contain subformulas $P_c(X(t))$ with the interpretation that the term $t$ is coloured with the colour $c$ in the configuration $X$. This formula be rewritten as an equivalent $\bar{\Sigma}_n$ formula if $\phi, \psi \in \bar{\Sigma}_n$, and analogously for $\bar{\Pi}_n$, by replacing $X$ with $|\mathcal{A}|$ subsets of $\mathbb{Z}^2$, one for each color, with the requirement that the sets form a partition of $\mathbb{Z}^2$. Thus, we allow quantification over configurations since it is often more convenient. Furthermore, we may restrict any such quantification to be over a set of configurations defined by a first order formula $\psi$ without changing the second-order quantifier complexity: $\forall_1 X \in X_\psi\ \phi$ can be implemented as $\forall_1 X(\neg\psi(X) \vee \phi)$, and $\exists_1 X \in X_\psi\ \phi$ as $\exists_1 X(\psi(X) \wedge \phi)$, after which the quantifiers of $\phi$ can be moved over $\psi(X)$ to bring it to prenex normal form.

*Example 1.* We define $\mathrm{fin}(E)$, a $\bar{\Sigma}_1$ formula with a free second order variable $E$ which is true if and only if the set $E \subset \mathbb{Z}^2$ is finite. The formula is $\mathrm{fin}(E) = \exists N \exists S\ \phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge \phi_5$, where

- $\phi_1 = \forall v\ (v \in N \Leftrightarrow \mathrm{West}(v) \in N \wedge \mathrm{South}(v) \in N)$
- $\phi_2 = \forall v\ (v \in S \Leftrightarrow \mathrm{East}(v) \in S \wedge \mathrm{North}(v) \in S)$
- $\phi_3 = \exists v\ v \in N \wedge \neg\,\mathrm{North}(v) \in N \wedge \neg\,\mathrm{East}(v) \in N$
- $\phi_4 = \exists v\ v \in S \wedge \neg\,\mathrm{South}(v) \in S \wedge \neg\,\mathrm{West}(v) \in S$
- $\phi_5 = \forall v\ v \in E \Rightarrow v \in N \wedge v \in S$

The idea is that $N \subset \mathbb{Z}^2$ is a nondegenerate southwest quarter-plane, $S \subset \mathbb{Z}^2$ is a nondegenerate northeast quarter-plane, and $E$ is contained in their intersection.

*Example 2.* We define $\mathrm{Stable}(X) = \forall n\ n \in X \Leftrightarrow \mathrm{East}(n) \in X$ where $X$ is a second order variable. The predicate $\mathrm{Stable}(X)$ is valuated to true if, whenever an element is in $X$, then all the elements of the same line are in $X$ too. We define also $\mathrm{Line}(X, Y)$ as the formula

$$\mathrm{Stable}(X) \wedge ((\mathrm{Stable}(Y) \wedge (\exists n\ n \in X \wedge n \in Y)) \Rightarrow (\forall n\ n \in X \Rightarrow n \in Y)).$$

Then $\forall Y\ \mathrm{Line}(X, Y)$ is true if and only if $X$ is exactly the set of elements of a line, or is empty. Finally, define the formula $\phi = \exists X\ \forall Y\ \mathrm{Line}(X, Y) \wedge (\forall n\ P_{\blacksquare}(n) \Rightarrow n \in X)$. The subshift $X_\phi$ consists of exactly those configurations where every symbol $\blacksquare$ appears in the same line.

*Example 3.* Define the formula $\phi = \exists x\ \forall y\ P_{\blacksquare}(y) \Leftrightarrow y = x$. Then $X_\phi$ is the set of configurations where there is exactly one position colored $\blacksquare$. It is not a subshift since it is not topologically closed. Indeed, there is a sequence of configurations in $X_\phi$ which converges to the all-$\square$ configuration, which is not in $X_\phi$.

## 3 First order formulas

In this section, we analyze the structure of FO-definable sets. The main result is that they are exactly the sets that can be defined by restricting the number of occurrences of a finite number of finite patterns up to some finite upper bound. Our techniques come from [2], where similar results were proved in the context of *picture languages*.

**Definition 1.** *Let $k \geq 0$. For $0 \leq a \leq k$ and a finite pattern $P$ on an alphabet $\mathcal{A}$, define $S_a^k(P) \subset \mathcal{A}^{\mathbb{Z}^2}$ as the set of configurations where $P$ appears exactly $a$ times if $a < k$, or at least $k$ times if $a = k$.*

*Let also $n \geq 0$. For a function $f : \mathcal{A}^{n \times n} \to [0, k]$, define $C_f = \bigcap_P S_{f(P)}^k(P)$. Then $\{C_f \mid f : \mathcal{A}^{n \times n} \to [0, k]\}$ is a partition of $\mathcal{A}^{\mathbb{Z}^2}$ into equivalence classes, and if two configurations $x, y \in \mathcal{A}^{\mathbb{Z}^2}$ are in the same class, we denote $x \sim_{(n,k)} y$.*

In other words, $x \sim_{(n,k)} y$ means that each $n \times n$ pattern either occurs exactly $t < k$ times in both $x$ and $y$, or occurs at least $k$ times in both.

**Lemma 1.** *Let $\phi$ be a first order formula. There exists a couple $(n, k) \in \mathbb{N}^2$ such that $X_\phi$ is a union of $\sim_{(n,k)}$-equivalence classes. Moreover, this union is computable from $\phi$.*

We stress that apart from the computability of the union, this result essentially appeared already in [3,2].

**Corollary 1.** *Let $\phi$ be a first order formula. We can compute two formulas of the form $\exists x_1 \ \ldots \ \exists x_n \ \forall y_1 \ \ldots \ \forall y_m \ \psi_1$ and $\forall x_1 \ \ldots \ \forall x_n \ \exists y_1 \ \ldots \ \exists y_m \ \psi_2$ with $\psi_1$ and $\psi_2$ quantifier-free which are equivalent with $\phi$.*

*Proof.* For all $n, k \in \mathbb{N}$, $0 \geq a \geq k$ and $P$ pattern of size $n \times n$, $S_a(P)$ is first-order definable by formulas of the form $\exists x_1 \ \ldots \ \exists x_n \ \forall y_1 \ \ldots \ \forall y_m \ \psi_1$ and of the form $\forall x_1 \ \ldots \ \forall x_n \ \exists y_1 \ \ldots \ \exists y_m \ \psi_2$, with $\psi_1$ and $\psi_2$ quantifier-free, and these formulas are computable from $S_a(P)$. It follows that it is also the case for every first-order formula using Lemma 1.

## 4 MSOSUB

In this section, we are interested in the following problems:

**Definition 2 (MSOSUB).** *Let $C$ be a class of MSO formulas. We define the following problem, called $C$-SUB: given a formula $\phi \in C$, is $X_\phi$ a subshift?*

If $C$ is $\bar{\Sigma}_0$ or $\bar{\Pi}_0$, we prefer to call this problem FOSUB.

### 4.1 FOSUB

**Theorem 1.** FOSUB *is $\Pi_4^0$-complete.*

We split the proof into Lemmas 2 and 3.

**Lemma 2.** FOSUB *is $\Pi_4^0$.*

*Proof (sketch).* Let $\phi$ be a first order formula. Using Lemma 1 we can compute $n, k \in \mathbb{N}$ and disjoint $\sim_{(n,k)}$-equivalence classes $C_1, \ldots, C_r$ such that $X_\phi = \bigcup_{i=1}^r C_i$. We want to prove that knowing whether $\bigcup_{i=1}^r C_i$ is closed is $\Pi_4^0$.

For a $\sim_{(n,k)}$-equivalence class $C = \bigcap_{P \in \mathcal{A}^{n^2}} S_{a_P}(P)$, define the function $f_C : \mathcal{A}^{n^2} \to \{0, \ldots, k\}$ by $f_C(P) = a_P$. A *map* of $C$ is a function $M : \mathcal{A}^{n^2} \to \mathcal{P}_{\text{fin}}(\mathbb{Z}^2)$. For $\ell \in \mathbb{N}$ and two maps $M_1, M_2$ of $C$, we say that $(C, \ell, M_1, M_2)$ *tiles the plane* if there exists $x \in \mathcal{A}^{\mathbb{Z}^2}$ such that for each $P \in \mathcal{A}^{n^2}$:

- if $f_C(P) < k$, then the pattern $P$ appears at positions $M_1(P) \cup M_2(P)$ and no other positions,
- if $f_C(P) = k$, then the pattern $P$ appears on positions $M_1(P) \cup M_2(P)$ (with $|M_1(P) \cup M_2(P)| \geq k$), and if $|M_1(P)| < k$, the only positions of $[-\ell, \ell-1]^2$ at which $P$ appears are in $M_1(P)$.

To decide whether a given quadruple tiles the plane is $\Pi_1^0$.

Now let us consider the following arithmetical formula $F$: there exist $m \in \mathbb{N}$, an index $i \in \{1, \ldots, r\}$ and two functions $I_1, I_2 : \mathcal{A}^{n^2} \to \{0, \ldots, k\}$ such that $I_1 + I_2 = f_{C_i}$, and for all $\ell > m$ there exist two maps $M_1, M_2$ of $C_i$ such that

- $|M_j(P)| = I_j(P)$ for each $j = 1, 2$ and $P \in \mathcal{A}^{n^2}$,
- $M_1(P) \subseteq [-m, m]^2$ and $M_2(P) \cap [-\ell, \ell-1]^2 = \emptyset$ for each $P \in \mathcal{A}^{n^2}$,
- $(C_i, \ell, M_1, M_2)$ tiles the plane, and
- there is no $j \in \{1, \ldots, r\}$ such that $(C_j, 0, M_1, P \mapsto \emptyset)$ tiles the plane.

By its form, $F$ is $\Sigma_4^0$. One can prove that $F$ holds if and only if $\bigcup_{i=1}^r C_i$ is not closed. The idea is that if $F$ holds, then one obtains a sequence $x_\ell$ of configurations in which the maps $M_1$ and $M_2$ control the positions of each pattern $P \in \mathcal{A}^{n^2}$, and the tiling conditions make sure that at least one pattern "escapes to infinity" in such a way that no limit point of the sequence is in $\bigcup_{i=1}^r C_i$.

**Lemma 3.** FOSUB *is $\Pi_4^0$-hard.*

*Proof (sketch).* We reduce the $\Pi_4^0$-complete problem $\forall COF$: given a Turing Machine $M$ which takes two inputs, does $M(v, \_)$ have a cofinite language for all $v$? For this, we define an SFT $X$ and add on it a first order formula to get a formula $\phi$. Then, we will prove that $M \in \forall COF$ if and only if $\phi \in$ FOSUB.

Let $M$ be a turing machine. The SFT $X$ is the one described by Figure 2 (the only allowed patterns are those of size $2 \times 2$ in this figure, but other inputs $v$ and $w$ are possible, even the infinite one). The input $v$ is copied to the southeast until it hits the gray diagonal signal emitted by the $*$-symbol at the end of the second input $w$, where it is erased. The diagonal turns black after the point of erasure. The machine $M$ is not allowed to halt, so in all configurations containing two finite inputs $v$ and $w$, $M(v, w)$ does not halt.

Let $\phi'$ be the formula which defines this SFT, and let $\phi = \phi' \wedge ((\exists x\ \exists y\ P_1(x) \wedge P_2(y)) \Rightarrow \exists z\ P_3(z))$, where tiles 1, 2 and 3 are showed in Figure 2). Intuitively, $\phi$ requires the configuration to satisfy $\phi'$, and if there is a diagonal with a finite input $v$ written on it, then there also exists a finite input $w$.

Now $M \in \forall COF$ if and only if $\phi \in MFOSUB$, since the only reason for $X_\phi$ not to be closed is that for some input $v$, there are infinitely many inputs $w$ such that $M(v, w)$ does not halt, and a limit point of the correspondng configurations violates the second condition of $\phi$.

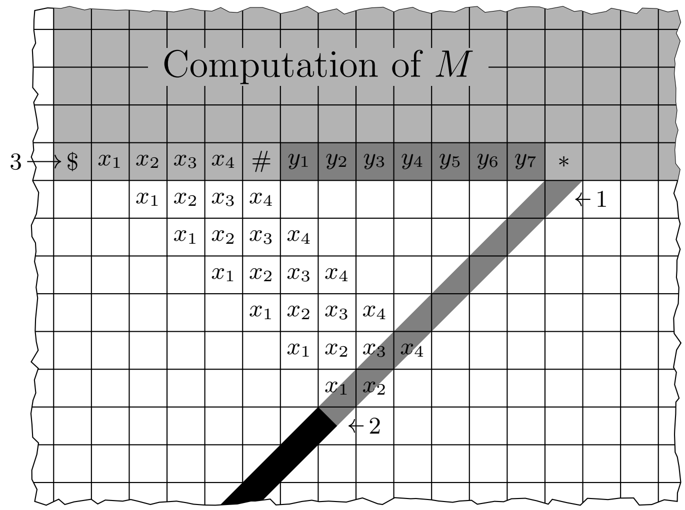


**Fig. 2.** A configuration of the SFT $X$ in the proof of Lemma 3.

### 4.2 Higher order cases

**Theorem 2.** *For all $n \geq 1$, $\bar{\Pi}_n$-SUB and $\bar{\Sigma}_n$-SUB are $\Pi^1_n$-complete.*

We split the proof into Lemmas 4, 5 and 6.

*Remark 2.* All problems of the form $\exists_1 X_1 \exists x_1 \forall x_2\ R$ with $R$ a decidable predicate, are in $\Sigma^0_2$. Dually, all problems of the form $\forall_1 X_1 \forall x_1 \exists x_2\ R$ with $R$ a decidable predicate, are in $\Pi^0_2$.

**Lemma 4.** *For all $n \geq 1$, $\bar{\Pi}_n$-SUB and $\bar{\Sigma}_n$-SUB are in $\Pi^1_n$.*

*Proof.* Let $\phi \in \bar{\Pi}_n$ with $m$ second order quantifiers (the $\bar{\Sigma}_n$ case is similar). Denote by $\phi^*$ the first order formula on $\mathcal{A} \times \{0,1\}^m$ obtained from $\phi$ by removing second order quantifiers and replacing each $v \in X_i$ with $\bigvee_{c \in \pi_i^{-1}(1)} P_c(v)$, where $\pi_i(c_0, \ldots, c_m) = c_i$. For a configuration $x \in \mathcal{A}^{\mathbb{Z}^2}$, we have $x \vDash \phi$ if and only if

$$\forall X_1 \ \ldots \ QX_m \ x \times X_1 \times \ldots \times X_m \vDash \phi^* \tag{1}$$

for $Q = \exists$ if $n$ is even, $Q = \forall$ otherwise. From Corollary 1 we deduce that $x \times X_1 \times \ldots \times X_m \vDash \phi^*$ is a $\Delta^0_2$ condition, when $\phi^*$ is given as input and $x$ and the $X_i$ are given as oracles. With Remark 2 it follows that (1), and hence $x \vDash \phi$, is $\Pi^0_2$ for $n = 1$, and $\Pi^1_{n-1}$ for $n \geq 2$, with $\phi$ as input and $x$ as an oracle.

The set $X_\phi$ is closed if and only if for all sequences $x_n$ of configurations such that $x_n \vDash \phi$ for all $n$ and $x_n|_{[-n,n]^2} = x_m|_{[-n,n]^2}$ for all $m \geq n$, we have $\lim_n x_n \vDash \phi$. By the above, this condition is $\Pi^1_n$.

**Lemma 5.** *For all $n \geq 1$, $\bar{\Pi}_n$-SUB is $\Pi^1_n$-hard.*

*Proof.* We reduce an arbitrary problem in $\Pi^1_n$. Let $R$ be a computable predicate. We construct a formula $\phi \in \bar{\Pi}_n$ such that $\forall A_1 \dots Q A_n\ \bar{Q}k\ Qm\ R$ holds if and only if $X_\phi$ is a subshift, with $Q = \exists$ if $n$ is even and $Q = \forall$ otherwise.

Let $\phi'$ be the formula which describes a tiling where a northeast quarter-plane is coloured with 0 and 1 such that each row is coloured in a same way, and the other tiles are blank. Since the quarter-plane must appear, $X_{\phi'}$ is not a subshift. The infinite binary sequence repeated on each line is interpreted as the oracle $A_1$ in the following.

Let $\phi = \forall X_2 \in X_{\phi'} \dots\ \bar{Q} X_n \in X_{\phi'}\ \bar{Q} Z\ \phi' \wedge \psi$, where $\psi$ is the formula stating the following:

- each of $X_2, \dots, X_n$ has a northeast quarter-plane at the same position as the main configuration $x \in X_{\phi'}$,
- the configuration $Z$ contains, in the same quarter-plane, a simulated computation of a Turing machine $M$ that can use the binary sequences of $x$ and the $X_i$ as oracles,
- if $\bar{Q} = \exists$, then $M$ visits a special state $q_s$ an infinite number of times (a $\bar{\Pi}_1$ condition by Example 1), and
- if $\bar{Q} = \forall$, then $M$ visits the state $q_s$ a finite number of times.

Notice that the configuration $Z$ is determined by what was quantified before, so its quantifier is irrelevant. Hence $\phi$ is $\bar{\Pi}_n$.

We specify the machine $M$. Starting with $k = 0$, it enumerates $m = 0, 1, 2, \dots$ until it finds one such that $R(A_1, \dots, A_n, k, m)$ does not hold. Whan this happens, it visits the state $q_s$, increments $k$, and starts the enumeration again.

Notice that $x \vDash \phi$ if and only if $\forall A_2 \dots\ \bar{Q} A_n\ \ Qk\ \bar{Q}m\ \neg R$ holds. Hence, if $\forall A_1 \exists A_2 \dots Q A_n\ \bar{Q}k\ Qm\ R$ holds, then $X_\phi$ is empty, and so a subshift. If the formula does not hold, then at least one configuration with a quarter-plane is in $X_\phi$. But the blank configuration is not in $X_\phi$, so $X_\phi$ is not a subshift.

**Lemma 6.** *For all $n \geq 1$, $\bar{\Sigma}_n$-SUB is $\Pi^1_n$-hard.*

The proof is similar to that of Lemma 5, but instead of emptiness, we use a separate binary layer with at most one 1-symbol.

## 5 Complexity of the MSO languages

In this section, we consider the complexity of the $C$-definable sets and subshifts for $C$ classes of the MSO hierarchy.

### 5.1 Maximal complexity

Here we determine the maximal complexity of languages of formulas in the different classes of the MSO hierarchy.

**Theorem 3.** *Languages of FO and $\bar{\Sigma}_1$ sets are $\Sigma^0_2$. Languages of $\bar{\Sigma}_n$ sets for $n \geq 2$ are $\Sigma^1_{n-1}$. Languages of $\bar{\Pi}_n$ sets for $n \geq 1$ are $\Sigma^1_n$.*

*Proof.* Let $\phi \in \bar{\Sigma}_n$. For a pattern $P$, we have $P \in \mathcal{L}(X_\phi)$ if there exists $x \in \mathcal{A}^{\mathbb{Z}^2}$ such that $x \vDash \phi$ and $P$ appears in $x$. Given $x$ as an oracle, $P$ appearing in $x$ is a $\Sigma^0_1$ predicate, and $x \vDash \phi$ is $\Sigma^1_{n-1}$ if $n \geq 2$, and $\Sigma^0_2$ if $n \leq 1$ thanks to Corollary 1 and Remark 2. We conclude that $\mathcal{L}(X_\phi)$ is $\Sigma^1_{n-1}$ if $n \geq 2$ and $\Sigma^0_2$ if $n \leq 1$ (we need once more Remark 2 for the case $n = 1$).

Let $\phi \in \bar{\Pi}_n$ for $n \geq 1$. Given $x \in \mathcal{A}^{\mathbb{Z}^2}$ as an oracle, $x \vDash \phi$ is $\Pi^1_{n-1}$ if $n \geq 2$, and $\Pi^0_2$ if $n = 1$ thanks to Corollary 1 and Remark 2. We conclude that $\mathcal{L}(X_\phi)$ is $\Sigma^1_n$ for all $n \geq 1$.

**Theorem 4.** *There exists an FO-definable subshift with a $\Sigma^0_2$-hard language.*

*Proof.* Let $\phi'$ be the first order formula which defines an SFT $X$ containing, in a northeast quarter-plane, a simulated computation of a turing machine $M'$. Figure 3 contains asample configuration. The machine $M'$ takes the code $[M]$ of another Turing machine $M$, and starting with $k = 0$, simulates $M$ with input $k$. If $M$ halts on $k$, then $M'$ enters a special state $q_H$, increments $k$ and starts again. The tile containing $q_H$ sends a signal (dark gray in Figure 3) to the west, which turns south when it encounters the left side of the quarter-plane, merging with another southward signal if one is present. We forbid to have such a signal under the diagonal of the quarter-plane.

Now let $\phi = \phi' \wedge ((\exists u\ \exists v\ P_1(u) \wedge P_2(v) \wedge P_3(\mathrm{South}(v))) \Rightarrow \exists w\ \neg P_4(w) \wedge P_3(\mathrm{West}(w)))$ (tiles 1, 2, 3 and 4 are shown on Figure 3). Intuitively, $\phi$ requires that $\phi'$ is true and if the computation starts and the input is finite, then there exists a tile on the west border of the quarter-plane with no signal on it.

We prove first that $X_\phi$ is a subshift. Suppose for a contradiction that there is a sequence $(x_n)_{n\in\mathbb{N}}$ such that $x_n \vDash \phi$ for all $n$ and $x_n \overset{n\to\infty}{\longrightarrow} x \nvDash \phi$. Since $X_{\phi'}$ is a subshift, $x \vDash \phi'$. Hence $x$ shows a computation with a Turing machine $M$ on which the west border of the quarter-plane is covered by the signal. As $x_n$ converges to $x$, from some $n$ onward, the code $[M]$ is entirely written on the south border of the quarter-planes of the $x_n$ and the computation of $M'$ is determined by this. Since $x_n \vDash \phi$, the west borders are not covered by the signal. Then the the sequence $x_n$ must be eventually constant. With $x \nvDash \phi$ we reach a contradiction, so $X_\phi$ is a subshift.

Notice that $[M]$ is in $\mathcal{L}(X_\phi)$ if and only if $M \in \mathrm{coTOTAL}$, i.e. if $M$ does not halt on at least one input. This problem is known to be $\Sigma^0_2$-complete.

**Theorem 5.** *For all $n \geq 1$, there exists a $\bar{\Pi}_n$-definable set with a $\Sigma^1_n$-hard language.*

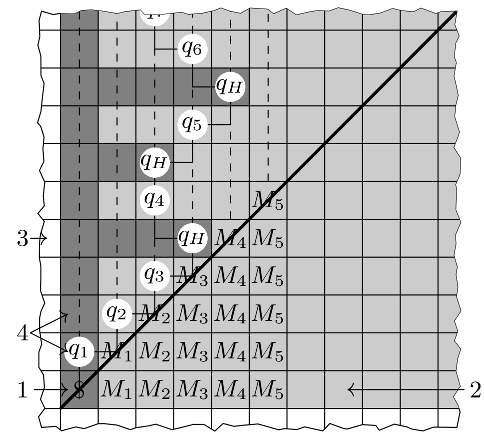


**Fig. 3.** Computation of $M'$.

*Proof.* Let $\phi'$ be the first order formula which defines the set of configurations that are blank except for an infinite binary word $w_1$, and the code $[M]$ of a total Turing machine $M$ written just above this word, at the left end. As no other configurations are allowed, $X_{\phi'}$ is not a subshift.

Let $\phi = \phi' \wedge \forall X_2 \ \exists X_3 \dots \ Q \ X_n \ Q \ Z \ \psi$, where $Q = \forall$ if $n$ is even and $Q = \exists$ otherwise, and $\psi$ specifies that each $X_i$ contains an infinite binary word $w_i$ at the same position as the main configuration $x$, and $Z$ contains a simulated computation which verifies $\exists k \ \forall m \ M(w_1, \dots, w_n, k, m)$ if $n$ is even, and $\forall k \ \exists m \ M(w_1, \dots, w_n, k, m)$ if $n$ is odd. We use the same technique as in Lemma 5 to design $\phi$ so that it is $\bar{\Pi}_n$.

Now $[M]$ is in $\mathcal{L}(X_\phi)$ if and only if

$$\exists w_1 \ \forall w_2 \ \dots \ Q \ w_n \ \bar{Q} \ k \ Q \ m \ M(w_1, \dots, w_n, k, m).$$

This condition is $\Sigma^1_n$-hard, so $\mathcal{L}(X_\phi)$ is $\Sigma^1_n$-hard too.

**Theorem 6.** *There exists a $\bar{\Pi}_1$-definable subshift with a $\Pi^0_3$-hard language.*

*Proof (sketch).* Let $X$ be the subshift in which all the configurations are blank except at most one infinite word, which contains the code $[M]$ of a Turing machine $M$ followed by an infinite sequence of tiles coloured with $\{0, 1, |\}$ (plus all limit configurations of these). Let $\psi$ be the first order formula such that $X = X_\psi$.

Let $\phi_1$ be a $\bar{\Pi}_1$ formula which requires that if $[M]$ is finite, then there are infinitely many $|$-tiles. We interpret this as $[M]$ being followed by an infinite list of natural numbers in binary.

Let $\phi_2$ be a $\bar{\Pi}_1$ formula which requires that if $[M]$ is finite, then for each natural number $n$ in the list, $M$ never halts on $n$. This can be done by universally quantifying on a configuration containing a simulated computation of $M(n)$.

Let $\phi_3$ be a $\bar{\Pi}_1$ formula which requires that if $[M]$ is finite, then for each natural number $n$, either $n$ occurs in the list, or $M(n) \downarrow$. This can be done by simulating $M(n)$ and simultaneously searching for $n$ in the list, and requiring that this computation halts at some point.

Let $\phi_4$ be a $\bar{\Pi}_1$ formula which requires that if $[M]$ is finite, the list of natural numbers written after $M$ is strictly increasing. Once again, we can do this using a single universal quantifier.

Then let $\phi$ be a $\bar{\Pi}_1$ formula equivalent with $\psi \wedge \phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4$ (just rewriting this formula in the prenex form). One can prove that $X_\phi$ is a subshift with a $\Pi^0_3$-hard language by reducing the language of Turing machines that diverge on infinitely many inputs, which is known to be $\Pi^0_3$-complete.

Thanks to Theorems 4 and 5, we conclude that there exists a $\bar{\Sigma}_1$-definable subshift with a $\Sigma^0_2$-hard language and for all $n \geq 2$, there exists a $\bar{\Sigma}_n$-definable subshift with a $\Sigma^1_{n-1}$-hard language.

### 5.2 Definable language classes

Here we are interested in knowing which language classes of the arithmetical and analytical hierarchy are included in the languages of sets or subshifts definable by MSO formulas according to their complexity.

**Theorem 7.** *Some decidable languages are not $\bar{\Sigma}_1$-definable by sets. Some decidable languages are not $\bar{\Pi}_1$-definable by sets.*

*Proof.* We prove here the $\bar{\Sigma}_1$ case; the $\bar{\Pi}_1$ case is similar. The proof is based on the technique of [4].

Let $X$ be the subshift on $\mathcal{A} = \{0, 1, \#\}$ which is the closure of the set of configurations with two square $\{0, 1\}$-patterns of the same size aligned and at distance one where everything around these patterns is colored by # and where each pattern is the mirror image of the other one. We denote the mirror image of a pattern $P$ by $P^R$. It is clear that $\mathcal{L}(X)$ is decidable.

Suppose for a contradiction that there exists a $\bar{\Sigma}_1$ formula $\phi = \exists Y \in \Sigma^{\mathbb{Z}^2}\ \psi \in \bar{\Sigma}_1$, where $\psi$ is FO, such that $\mathcal{L}(X) = \mathcal{L}(X_\phi)$. For $N \geq 1$ and $P \in \{0, 1\}^{N^2}$, let $x_P \in X$ be the configuration where $x|_{[-N,-1]\times[0,N-1]} = P$ and $x|_{[0,N-1]^2} = P^R$. The pattern $\rho_P = x|_{[-N-1,N]\times[-1,N]}$, which contains $P$ and $P^R$ surrounded by #-symbols, is in $\mathcal{L}(X) = \mathcal{L}(X_\phi)$, so it appears in some configuration of $X_\phi$. We see that the only configuration of $X_\phi$ in which $\rho_P$ appears in the same position as in $x_P$ must be $x_P$ itself.

As $\psi$ is FO, by Lemma 1 there exist $n, k \in \mathbb{N}$ such that $X_\psi$ is a union of $\sim_{(n,k)}$-equivalence classes. For $P \in \{0, 1\}^{N^2}$, let $y_P \in \Sigma^{\mathbb{Z}^2}$ be such that $x_P \times y_P \vDash \psi$. For $N$ large enough there exist $P \neq Q \in \{0, 1\}^{N^2}$ such that the pattern $(x_P \times y_P)|_P$ has the same border of size $n$ as $(x_Q \times y_Q)|_Q$ and the same number of each $n \times n$ pattern counted up to $k$ (since $(2|\Sigma|)^{4Nn-4n^2}(k+1)^{(2|\Sigma|)^{n^2}} < 2^{N^2}$ for large $N$).

Now consider the configuration $z = (z_1, z_2)$ defined by $z_{\boldsymbol{v}} = (x_P \times y_P)_{\boldsymbol{v}}$ for $\boldsymbol{v} \notin [0, N-1]^2$ and $z_{\boldsymbol{v}} = (x_Q \times y_Q)_{\boldsymbol{v}}$ for $\boldsymbol{v} \in [0, N-1]^2$. In other words, we have replaced the $P^R$-patch of $x_P$ with $Q^R$, and made the corresponding replacement on the $y$-layer as well. Let us prove that $z \sim_{(n,k)} x_P \times y_P$. They are equal outside

$C_{\text{in}} = [n, N-n-1]^2$ (see Figure 4) and they have the same number of patterns of size $n \times n$ counted up to $k$ inside $C_{\text{out}} = \mathbb{Z}^2 \setminus [0, N-1]^2$. So if $z$ and $x_P \times y_P$ are not equivalent, this is due to a pattern of size $n \times n$ which overlaps $C_{\text{in}}$ and $C_{\text{out}}$, which is impossible.

Thus $z \in X_\psi$ and $z_1 \in X_\phi \subseteq X$. But $z_1$ consists of the patterns $P$ and $Q^R$ surrounded by #-symbols, so it is not in $X$, and we have a contradiction.

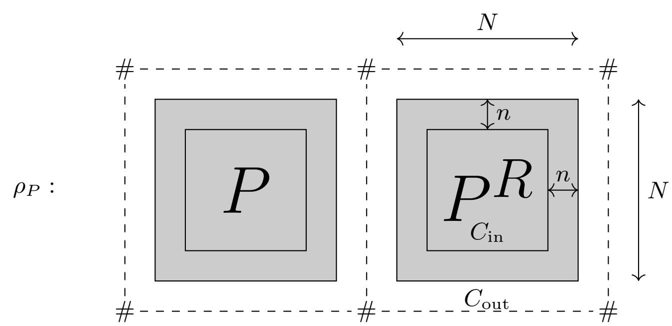


**Fig. 4.** Non $\bar{\Sigma}_1$-definability of a decidable language

**Theorem 8.** *For all $n \geq 2$, all $\Pi^1_{n-1}$ subshifts are $\bar{\Pi}_n$-definable, and all $\Sigma^1_{n-1}$ subshifts are $\bar{\Sigma}_n$-definable.*

*Proof (sketch).* We sketch the proof for $n = 3$. In the first case, we have a subshift $X \subseteq \mathcal{A}^{\mathbb{Z}^2}$ whose language is defined by a $\Pi^1_2$ formula $\forall_1 A_1 \exists_1 A_2 \forall k \exists m \xi$, where $\xi$ is computable. We construct an MSO formula $\phi = \forall X_1 \exists X_2 \forall X_3 \psi$, where the $X_i$ are constrained by FO conditions, that defines $X$.

We follow the technique of [9, Theorem 3]. The configuration $X_1$ contains a finite rectangle $R$, an encoding $w$ of a finite rectangular pattern over $\mathcal{A}$, and an infinite binary word $w_1$. The configuration $X_2$ contains either a gadget witnessing that $w$ is not the encoding of the pattern $x|_R$ in the main configuration $x \in \mathcal{A}^{\mathbb{Z}^2}$ (either due to having the wrong dimensions, or differing at a single coordinate), or an infinite binary word $w_2$. The configuration $X_3$ contains an infinite simulated computation of a Turing machine that can access the infinite words $w_1$ and $w_2$ as oracle tapes. The machine checks sequentially for $k = 0, 1, 2, \ldots$ that there exists $m \in \mathbb{N}$ such that $\xi$ holds. Whenever such an $m$ is found, the machine enters a special state $q_s$ and moves to the next value of $k$. The configuration also contains an upper half-plane $H$. The formula $F$ checks that if $X_2$ contains the word $w_2$, then $X_3$ contains a $q_s$ within the half-plane $H$.

Suppose $x \vDash \phi$. Given a rectangle $R$, take an $X_1$ that correctly encodes $x|_R$ into a word $w$ and contains some $w_1$. There exists an $X_2$ that necessarily contains a $w_2$ (because a witness for $w$ not encoding the pattern does not exist), and wherever we place the upper half-plane in $X_3$, it contains $q_s$. Hence the machine in $X_3$ enters $q_s$ an infinite number of times, and $x|_R \in \mathcal{L}(X)$.

Suppose then $x \in X$. Every valid configuration $X_1$ contains some rectangle $R$ and some words $w$ and $w_1$ above it. If $w$ does not encode $x|_R$, we can find a witness and choose it as $X_2$. If it does, we can choose $X_2$ to contain the word $w_2$ that corresponds to a suitable $A_2$. Then every $X_3$ satisfies $\psi$, since either it does not have the correct form, or the Turing machine enters $q_s$ an infinite number of times. Hence $x \vDash \phi$.

In the second case, we have a subshift $X \subseteq \mathcal{A}^{\mathbb{Z}^2}$ whose language is defined by a $\Sigma^1_2$ formula $\exists_1 A_1 \forall_1 A_2 \exists k \forall m \xi$, where $\xi$ is computable. We construct an MSO formula $\phi = \exists X_1 \forall X_2 \exists X_3 \psi$ that defines $X$. Here, $X_1$ contains the infinite computation of a Turing machine that simply lists all finite rectangles $R$ and nondeterministically guesses a pattern $P \in \mathcal{A}^R$ and an infinite binary word $w_1$ for each. It also selects one cell $\boldsymbol{w} \in \mathbb{Z}^2$ as the "origin". The configuration $X_2$ selects one of these rectangles, and either selects a position $\boldsymbol{v} \in R$ and contains a witness gadget for $x_{\boldsymbol{w}+\boldsymbol{v}} = P_{\boldsymbol{v}}$, or contains an infinite binary word $w_2$ at the same position as the word $w_1$ of the encoded rectangle. The configuration $X_3$ is similar to the $X_3$ of the $\Pi^1_2$ case, except that the Turing machine looks for an $m$ such that $\xi$ does not hold. The formula $F$ is almost as before: if $X_2$ contains a $w_2$, then $X_3$ does not contain a $q_s$ within $H$.

Suppose $x \vDash \phi$, and choose an $X_1$ from $\phi$ with origin $\boldsymbol{w}$. Given a rectangle $R$, we can find it encoded in $X_1$ together with a pattern $P$ and a word $w_1$. For each $X_2$ that selects the encoding of $R$ and $P$ and a position $\boldsymbol{v} \in R$, we get a witness for $x_{\boldsymbol{w}+\boldsymbol{v}} = R_{\boldsymbol{v}}$. For those $X_2$ that contain a $w_2$ instead, we get an $X_3$ whose simulated Turing machine enters $q_s$ a finite number of times. Hence $x|_R \in \mathcal{L}(X)$.

Suppose then $x \in X$, and choose a $X_1$ with origin $\mathbf{0}$ that correctly encodes every rectangular pattern $P$ in $x$, and for each, contains a word $w_1$ corresponding to the set $A_1$ given by the $\Sigma^1_2$ formula of $\mathcal{L}(X)$. Every $X_2$ either contains a witness for the correctness of a single coordinate in some encoded pattern (which are all correct by definition) or a word $w_2$. Then the machine simulated in $X_3$ enters $q_s$ only a finite number of times, so we can place the half-plane $H$ above the last occurrence to satisfy $\psi$. Hence $x \vDash \phi$.

## 6 Conclusion

In Section 4 we determined the complexity of $C$-SUB for various classes $C$. FOSUB turned out to be $\Pi^0_4$-complete, while $\bar{\Pi}_n$-SUB and $\bar{\Sigma}_n$-SUB for $n \geq 1$ are both $\Pi^1_n$-complete.

The results of Section 5 are summarized in Table 1. Note that we have completely characterized the $\bar{\Sigma}_n$ subshifts with $n \geq 2$, as well as the languages of $\bar{\Sigma}_n$ sets. We have also pinpointed the maximal complexity of the languages of first order and $\bar{\Sigma}_1$ subshifts, which are both $\Sigma^0_2$, higher than the well known $\Sigma^0_1$ bound of SFTs and sofic shifts. The main remaining question concerns the maximal complexity of $\bar{\Pi}_1$ subshifts: is it $\Sigma^1_n$, the same as for general $\bar{\Pi}_1$ sets, or is it strictly lower?

**Table 1.** Summary of the complexity of MSO languages.

| | Sets | | Subshifts | |
|---|---|---|---|---|
| | Max complexity | Includes all in | Max complexity | Includes all in |
| FO | $\Sigma^0_2$ | $\Delta^0_1 \not\subseteq$ | $\Sigma^0_2$ | $\Delta^0_1 \not\subseteq$ |
| $\bar{\Sigma}_1$ | $\Sigma^0_2$ | $\Delta^0_1 \not\subseteq$ | $\Sigma^0_2$ | $\Delta^0_1 \not\subseteq$ |
| $\bar{\Pi}_1$ | $\Sigma^1_1$ | $\Delta^0_1 \not\subseteq$ | $[\Pi^0_3, \Sigma^1_1]$ | $\Delta^0_1 \not\subseteq$ |
| $\bar{\Sigma}_n, n \geq 2$ | $\Sigma^1_{n-1}$ | $\Sigma^1_{n-1}$ | $\Sigma^1_{n-1}$ | $\Sigma^1_{n-1}$ |
| $\bar{\Pi}_n, n \geq 2$ | $\Sigma^1_n$ | $\Pi^1_{n-1}$ | $[\Pi^1_{n-1}, \Sigma^1_n]$ | $\Pi^1_{n-1}$ |

**Acknowledgments.** Ilkka Törmä was supported by the Academy of Finland grant 359921.

**Disclosure of Interests.** The authors have no competing interests.

## References

1. R. Berger. *The Undecidability of the Domino Problem*. Memoirs of the American Mathematical Society. American Mathematical Society, 1966. URL: https://books.google.fi/books?id=mLfTCQAAQBAJ.
2. Dora Giammarresi, Antonio Restivo, Sebastian Seibert, and Wolfgang Thomas. Monadic second-order logic over rectangular pictures and recognizability by tiling systems. *Inf. Comput.*, 125(1):32–45, feb 1996. doi:10.1006/inco.1996.0018.
3. Emmanuel Jeandel and Guillaume Theyssier. Subshifts as models for MSO logic. *Information and Computation*, 225:1–15, 2013.
4. Steve Kass and Kathleen Madden. A sufficient condition for non-soficness of higher-dimensional subshifts. *Proceedings of the American Mathematical Society*, 141(11):3803–3816, 2013. doi:10.1090/S0002-9939-2013-11646-1.
5. Douglas Lind and Brian Marcus. *An introduction to symbolic dynamics and coding*. Cambridge Mathematical Library. Cambridge University Press, Cambridge, second edition, 2021. doi:10.1017/9781108899727.
6. Ville Salo and Ilkka Törmä. Diddy: a python toolbox for infinite discrete dynamical systems. In *Cellular automata and discrete complex systems*, volume 14152 of *Lecture Notes in Comput. Sci.*, pages 33–47. Springer, Cham, [2023] ©2023. URL: https://doi.org/10.1007/978-3-031-42250-8_3, doi:10.1007/978-3-031-42250-8_3.
7. Ville Salo and Ilkka Törmä. Griddy, 2023. GitHub repository. URL: https://github.com/ilkka-torma/griddy.
8. Wolfgang Thomas. Languages, automata, and logic. In *Handbook of formal languages, Vol. 3*, pages 389–455. Springer, Berlin, 1997.
9. Ilkka Törmä. Subshifts, MSO logic, and collapsing hierarchies. In Josep Diaz, Ivan Lanese, and Davide Sangiorgi, editors, *Theoretical Computer Science*, pages 111–122, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.
10. Hao Wang. Proving theorems by pattern recognition — II. *The Bell System Technical Journal*, 40(1):1–41, 1961. doi:10.1002/j.1538-7305.1961.tb03975.x.