Decidability and Universality of Quasiminimal Subshifts

Ville Salo

vosalo@utu.fi

Abstract

We introduce quasiminimal subshifts, subshifts having only finitely many subsystems. With $\mathbb{N}$-actions, their theory essentially reduces to the theory of minimal systems, but with $\mathbb{Z}$-actions, the class is much larger. We show many examples of such subshifts, and in particular construct a universal system with only a single proper subsystem, refuting a conjecture of [Delvenne, Kůrka, Blondel, '05].

Keywords:
quasiminimal, minimal, subshift, universality, decidability

1. Introduction

One of the most studied classes of subshifts[1] in the literature is that of minimal subshifts. These are precisely the nonempty subshifts containing no proper nonempty (sub-)subshifts. Some reasons that these subshifts are of great interest are that every subshift contains a minimal subshift, and many natural examples of subshifts, such as those generated by primitive substitutions and those generated by Toeplitz sequences, are minimal. We direct the reader to [1, 2] for a discussion of such systems. More generally, dynamical systems on compact metric spaces always contain minimal subsystems.

In [3], zero-dimensional dynamical systems with an $\mathbb{N}$-action and an effective presentation, called *symbolic systems*, are studied from the point of view of computational universality. These systems generalize one-sided subshifts whose language is recursive, and also many other symbolic systems

[1] Here, subshifts are sets $X \subset S^M$ which are topologically closed, and also closed under the shift action of M in the sense $X \cdot M \subset X$, where M is a monoid and S a finite set. In this paper, we consider the cases $M \in \{\mathbb{N}, \mathbb{Z}\}$.

such as Turing machines, counter machines and tag systems. Given any finite clopen partition of a space and an ω-regular language of infinite words with partition elements as letters, the *model-checking problem* is defined as the problem of checking whether one of the sequences in the language corresponds to a sequence of observations along an orbit. The model-checking problem for regular languages is the question of whether a finite sequence of observations in the language can be made in the system. In [3] a system is called *decidable* if the model-checking problem for ω-regular languages is decidable, and *universal* if its model-checking problem for regular languages is Σ^0_1-complete (that is, the halting problem many-one reduces to it). The definitions imply that a universal system cannot be decidable,[2] but a gap is left between the two definitions, so that both decidability and undecidability results are maximally strong.

In the case of subshifts, the model-checking problem for regular languages amounts to asking whether the intersection of the language of the subshift with a given regular language is empty, and we use this as the definition, omitting the details of computable presentations. The main results of this paper are constructions of subshifts which are universal, and thus ω-regular languages do not play a major role in this paper, and we only discuss them for context.

One of the main results of [3] is that if a minimal system is computable,[3] then it is decidable in the above sense. This shows in particular that many non-trivial things can be computed about minimal subshifts, as long as the forbidden patterns of the subshift can be enumerated. For example, given[4] a Turing machine enumerating the forbidden patterns of a (nonempty) minimal subshift X over the alphabet $\{0, 1, 2\}$, an algorithm can check whether there exists a subword w of a point of X where the number of 1s differs from the number of 0s by more than 7.

It is not particularly hard to show that if the forbidden words of X can be enumerated and X is minimal, then also the words that do occur in X can be enumerated (see Theorem 7), so that if such w exists, a rather simple

[2] This is because the model-checking problem of regular languages many-one reduces to that for ω-regular languages.

[3] In their paper, computable means recursive, or having $\Delta^0_1 = \Sigma^0_1 \cap \Pi^0_1$ language, but see Theorem 7.

[4] It is not explicitly stated in [3] that the algorithm is uniform in the description of the subshift, but this is clear from their proof.

algorithm can find it. The algorithm of [3] must be even smarter: if no w with the above properties exists, then after enumerating some finite number of forbidden patterns, the algorithm will state this fact.

The algorithm of [3] applies more generally to systems where every proper subsystem has nonempty interior, and to systems whose limit set is a finite union of minimal systems. It is interesting to ask where the precise border of decidability lies, by extending the family of subshifts further, and to this end the authors also make the following conjecture.

Conjecture 1 ([3]). *A universal symbolic system has infinitely many minimal subsystems.*

Conversely, this conjecture would imply that a system with finitely many minimal subsystems has to be non-universal (even if not necessarily decidable). Note that this conjecture talks about infinitely many *minimal* subsystems, but allows us to have any number of subsystems in general. It turns out that this is not a very strong condition, and as such, the conjecture is false.

Proposition 1. *There exists a recursive universal subshift with finitely many minimal subsystems.*

The proof of this is given in Section 3, Proposition 3. Our example is a $\mathbb{Z}$-subshift, and we interpret subsystems in the sense of being closed under the $\mathbb{Z}$-action, while subsystems in the sense of [3] need to be closed under the induced $\mathbb{N}$-action only. It is easy to see that this does not change the number of minimal subsystems, although it may change the number of subsystems in general (see Proposition 4 for details), so that our example also provides an $\mathbb{N}$-system with the desired properties. Alternatively, one can directly modify our example to be one-sided.

The subshift is very simple, and we offer multiple variations of it. At its simplest, the number of minimal subsystems is one, and this subsystem is just a single point.[5] The subshift is countable, and in fact contained in a countable sofic shift. Alternatively, the subshift can be contained in a countable SFT, although with slightly more minimal subsystems. The

[5] The one-point subshift is the simplest minimal subshift, but any minimal subshift can be used here, though naturally at the expense of countability.

Cantor-Bendixson rank of the enveloping countable sofic shift is 4. CB-rank 4, for a countable sofic shift, means that each point contains at most 3 disturbances to periodicity.[6] In addition to the subshift being very simple, also the universality is very strong: not only are intersections with regular languages Σ^0_1-complete, but even the *undirected halting problem*, the question of whether two given words u and v occur in the same point, is Σ^0_1-complete.

While X has only finitely many minimal subsystems, it has infinitely many subsystems altogether. In fact, the set of subshifts of X has the cardinality of the continuum (which is the maximal possible). In the abstract and introduction of [3], the authors state the conjecture in a weaker form, asking if a universal system should have infinitely many subsystems, without any mention of minimality. While this was presumably just meant as a shortened form of the statement of Conjecture 1, it is a natural next question whether at least this is true.

Question 1. *Must universal systems have infinitely many subsystems?*

We name this class for easier reference.

Definition 1. *A subshift is* quasiminimal *if it has finitely many subshifts.*

The question for us is then whether a recursive quasiminimal subshift can be universal. One might guess that if X has only finitely many subshifts, then it must be a quite simple extension[7] of the one or more minimal subshifts it necessarily contains, or perhaps even essentially just a union. It turns out that, unlike in Proposition 1, whether this is true depends on whether the acting monoid is $\mathbb{N}$ or $\mathbb{Z}$, that is, on whether our subshifts are one- or two-sided.[8]

[6] More precisely, there exists p such that every point x of the subshift satisfies $x_i = x_{i+p}$ for all $i \in \mathbb{Z} \setminus A$, where A is a union of three intervals which all have uniformly bounded length.

[7] Here, we use the term extension in the sense of containment (monomorphisms), and not in the sense of factoring (epimorphisms). This usage is very nonstandard in the theory of dynamical systems, but it is fitting for quasiminimal systems, since they are inductively built, in finitely many steps, from smaller quasiminimal systems by adding new points.

[8] Strictly speaking, we could also consider $S^{\mathbb{Z}}$ with an $\mathbb{N}$-action (obtaining a rather unnatural definition of a subshift), but the convention is that S^M uses the natural shift action of M. This is also required for the dynamical characterization of subshifts as expansive systems.

In the case of $\mathbb{N}$-actions, we show that the extensions of minimal subshifts to quasiminimal ones are rather trivial, and the answer to Question 1 is "yes".

Theorem 1. *A recursive quasiminimal* $\mathbb{N}$*-subshift is decidable.*

This is shown in Corollary 4. The proof is quite short: we characterize this class, and show that it fits one of the decidability results proved in [3]. Our main interest is in the case of $\mathbb{Z}$-actions, where the extensions can be quite complicated. For example, while primitive substitutions give rise to minimal subshifts, all substitutions satisfying a technical property (in particular all aperiodic substitutions) give rise to quasiminimal ones, see Proposition 2. These examples already show that quasiminimality is indeed quite different from minimality. In fact, in the setting of $\mathbb{Z}$-subshifts, the answer to Question 1 is "no".

Theorem 2. *There exists a recursive universal quasiminimal* $\mathbb{Z}$*-subshift.*

This follows for example from Theorem 3.

While this answers Question 1 completely in the case of $\mathbb{Z}$-actions, it turns out to be a bit more intricate than Conjecture 1 – for example, we will see that the halting problem (whether a clopen set is reachable from another one) is decidable for a recursive quasiminimal subshift. In fact, given any ordered finite tuple of words, an algorithm can check whether there exists a point in the subshift containing those words, in the given order (see Theorem 10). In particular (though not quite equivalently) we can solve the model-checking problem for regular languages of the form $w_1S^*w_2S^* \cdots S^*w_k$ where S is the alphabet of the subshift and the w_i are arbitrary words. This class of regular languages is related to the piecewise testable regular languages.

On the other hand, in Theorem 3 we show that the halting problem along a clopen set is Σ^0_1-complete, that is, the halting problem where we restrict the path between the given clopen sets to stay in a third clopen set. This corresponds to the fact that the model-checking problem is undecidable for local languages, a well-known subclass of regular languages.

We also study countable quasiminimal subshifts. For these, we believe the model-checking problem for local languages is in fact decidable (Conjecture 2), which would show a great difference between the countable and uncountable cases. We prove the undecidability of the model-checking problem for piecewise testable languages, star-free languages and what we call renewal languages (languages obtained as concatenations of a set of words, starting and ending in given sets of words), in Theorem 5 and Theorem 4. The model-checking problem for piecewise testable languages and star-free ones is dealt with by showing that the "halting problem along a clopen set except on exactly k steps" (which we call the counting problem) is Σ_1^0-complete, and the model-checking problem for renewal languages by showing that the "halting problem in time m mod k" is Σ_1^0-complete.

An obvious question is how many proper subshifts exactly do we need to have to achieve universality, since the result of [3] shows we need at least one. In the uncountable case, our example has only one minimal subshift (which can be chosen to be any infinite minimal subshift). This is clearly optimal. For countable quasiminimal subshifts, the number of nontrivial (nonempty and non-full) subshifts is two in our example for renewal systems, and we show this to be optimal in Proposition 8, since countable systems with only one subsystem are in fact decidable in general. For the particular case of piecewise testable languages, the optimal number of subshifts is shown to be 4.

In addition to proving a small gap between decidability and undecidability, these constructions are interesting as first explicit examples of quasiminimal $\mathbb{Z}$-subshifts. In particular, they show that a quasiminimal $\mathbb{Z}$-subshift can be quite far from a union of minimal $\mathbb{Z}$-subshifts, and even countable quasiminimal $\mathbb{Z}$-subshifts can be rather complicated objects. In fact, these constructions are in some sense representative of the general case, in that a full characterization of quasiminimal subshifts can be obtained by generalizing and iterating the constructions of this article. We defer the full characterization to a later work [4], and prove here only enough structural results to solve our decidability problems of interest in Section 4.

In addition to results about (non-minimal) quasiminimal subshifts, we also prove some new undecidability results about minimal subshifts, which complement the results of [3] in other directions: There is a recursive minimal subshift whose model-checking problem for context-free languages is undecidable (Theorem 6) and a minimal subshift whose language is recursively enumerable but not recursive (Proposition 7).

Our main results about subshifts with restricted subsystems are summarized in Table 1. See 2.3 for more information about model-checking and halting problems.

Restriction	$\mathbb{N}/\mathbb{Z}$	EPT	PT	Loc	SF	Ren	Reg	CF
one minimal subsystem	$\mathbb{N}/\mathbb{Z}$	U1	U1	U1	U1	U1	U1	U1
substitutive	$\mathbb{N}/\mathbb{Z}$	D1	D1	D1	D1	D1	D1	?1
$\mathrm{Q}(X) \in \mathbb{N}$	$\mathbb{N}$	D2	D2	D2	D2	D2	D2	U5
$\mathrm{Q}(X) \leq 2$	$\mathbb{N}/\mathbb{Z}$	D2	D2	D2	D2	D2	D2	U5
$\mathrm{Q}(X) \leq 3$	$\mathbb{Z}$	D5	U2	U2	U2	U2	U2	U2
$\mathrm{Q}(X) \in \mathbb{N}$	$\mathbb{Z}$	D5	U2	U2	U2	U2	U2	U2
$\mathrm{Q}(Y) \leq 3$	$\mathbb{Z}$	D5	D3	D3	D3	D3	D3	D3
$\mathrm{Q}(Y) \leq 4$	$\mathbb{Z}$	D5	D4	?2	D4	U3	U3	U3
$\mathrm{Q}(Y) \leq 5$	$\mathbb{Z}$	D5	D4	?2	D4	U3	U3	U3
$\mathrm{Q}(Y) \leq 6$	$\mathbb{Z}$	D5	U4	?2	U4	U3	U3	U3
$\mathrm{Q}(Y) \in \mathbb{N}$	$\mathbb{Z}$	D5	U4	?2	U4	U3	U3	U3

Table 1: Summary of our results on subshifts with subsystem restrictions. On the left, X is any subshift with a recursive language, and Y is any countable subshift with a recursive language. By Q we mean the number of subsystems, including the subshift itself and the empty set. The second column specifies whether we are talking about $\mathbb{N}$- or $\mathbb{Z}$-subshifts (or both). Other columns are model-checking problems for classes of languages, and the abbreviations stand for 'elementary piecewise testable', 'local', 'star-free', 'renewal', 'regular' and 'context-free' languages, respectively. In the entries, U followed by a number means the problem is undecidable, and D means decidable. The numbers are explained in Table 2.

U1	Halting problem is undecidable.	Proposition 3
U2	Halting problem along a clopen set is undecidable.	Theorem 3
U3	Modular halting problem is undecidable.	Theorem 4
U4	Counting problem is undecidable.	Theorem 5
U5		Theorem 6
D1		Lemma 3
D2	These follow from results of [3].	Corollary 8
D3	These systems are eventually periodic.	Proposition 8
D4		Theorem 12
D5		Theorem 10
?1		Question 2
?2		Conjecture 2

Table 2: Explanations of the uppercase entries.

2. Definitions and basic observations

2.1. More or less standard definitions

Remark 1. *In this article,* $\mathbb{N} = \{0, 1, 2, 3, \dots\}$*, and words are* 0*-indexed. The empty word is denoted by* ϵ*. We give the definitions of subshifts and related concepts for* $\mathbb{Z}$*-actions in this section. It is easy to modify them in the case of* $\mathbb{N}$*-actions for the few places where we need them, the crucial difference being that we do not require surjectivity for* $\mathbb{N}$*-actions. A subset of* $S^{\mathbb{Z}}$*, when considered a dynamical system, will always be considered a system with a* $\mathbb{Z}$*-action given by the shift, and a subset of* $S^{\mathbb{N}}$ *is an* $\mathbb{N}$*-system.*

Definition 2. *A* subshift *is a compact subset* X *of* $\mathbb{N}^{\mathbb{Z}}$*, where* $\mathbb{N}$ *has the discrete topology and* $\mathbb{N}^{\mathbb{Z}}$ *the product topology, and which is shift-invariant, in the sense that* $\sigma(X) = X$*, where* $\sigma : \mathbb{N}^{\mathbb{Z}} \to \mathbb{N}^{\mathbb{Z}}$ *is the shift map* $\sigma(x)_i = x_{i+1}$*.*

For standard references see [5, 2, 6].

Note that this indeed corresponds (up to symbol renaming) to the usual definition where a finite set is used in place of $\mathbb{N}$, and closed shift-invariant subsets are called subshifts. Namely, for any finite $S \subset \mathbb{N}$, $X \subset S^{\mathbb{Z}}$ is a subshift if and only if X is closed and shift-invariant, since $S^{\mathbb{Z}}$ is compact and Hausdorff, so that its closed subsets are precisely its compact subsets. On the other hand, no shift-invariant set $X \subset \mathbb{N}^{\mathbb{Z}}$ containing infinitely many letters is compact, since the open cover $\{[a]_0 \mid a \in \mathbb{N}\}$ has no finite subcover. Accordingly, when we write that $X \subset S^{\mathbb{Z}}$ is a subshift, we imply that S is chosen to be some finite alphabet. The induced topology of $S^{\mathbb{Z}}$ for finite S is generated by the *cylinders* $[w]_i = \{x \in S^{\mathbb{Z}} \mid x_{[i,i+|w|-1]} = w\}$ for $w \in S^*$ and $i \in \mathbb{Z}$, which are clopen in $S^{\mathbb{Z}}$. The clopen sets are precisely the finite unions of cylinders, and can all be represented as $[C]_i = [w_1]_i \cup [w_2]_i \cdots \cup [w_k]_i$ for some $i \in \mathbb{Z}$, $n \in \mathbb{N}$ and $C = \{w_1, \dots, w_k\} \subset S^n$. We write $[C] = [C]_0$.

For words $u, v \in \mathbb{N}^*$ (that is, two finite words over $\mathbb{N}$), we write $u \cdot v$ or uv for their concatenation. For words $u, v, v', w \in \mathbb{N}^*$, we write ${}^{\infty}uv.v'w^{\infty}$ for the point $x \in \mathbb{N}^{\mathbb{Z}}$ with $x_{[0,|v'|-1]} = v'$, $x_{[-|v|,-1]} = v$, and for all $i \in \mathbb{N}$,

$$x_{[-|v|-(i+1)|u|,-|v|-1-i|u|]} = u$$

and

$$x_{[|v|+i|w|,|v|+(i+1)|w|-1]} = w.$$

Note that coordinate 0 is to the right of the decimal point. Often, however, this position is not relevant, and the decimal point is omitted. We sometimes write $\cdots u.v \cdots$ when the continuations from u to the left and from v to the right are easy to guess. For example, $x = \cdots 0000.0123 \cdots$ is the point $x \in \mathbb{N}^{\mathbb{Z}}$ where for all $i \in \mathbb{N}$ we have $x_{-i} = 0$ and $x_i = i$. Similar notations are used for points $x \in \mathbb{N}^{\mathbb{N}}$. Writing $-\mathbb{N} = \{-n \mid n \in \mathbb{N}\}$, for $x \in \mathbb{N}^{-\mathbb{N}}$ and $y \in \mathbb{N}^{\mathbb{N}}$ we define $x.y = z$ where $z_i = x_{i+1}$ when $i < 0$ and $z_i = y_i$ otherwise.

A *central pattern* of $x \in \mathbb{N}^{\mathbb{Z}}$ is a word $x_{[-n,n]} \in \mathbb{N}^*$ for some $n \in \mathbb{N}$. For $x \in \mathbb{N}^{\mathbb{Z}}$, we write $u \sqsubset x$ if $x_{[i,i+|u|-1]} = u$ for some $i \in \mathbb{Z}$. We also write $u \sqsubset v$ for two words $u, v \in \mathbb{N}^*$ if u is a subword of v. This extends to sets $A \subset \mathbb{N}^{\mathbb{Z}}$ by $u \sqsubset A$ if $u \sqsubset x$ for some $x \in A$.

A subshift can also be defined by a set $F \subset S^*$ of *forbidden words*, in the sense that if $X \subset S^{\mathbb{Z}}$ is a subshift, then there exists $F \subset S^*$ such that

$$X = \{x \in S^{\mathbb{Z}} \mid \forall u \in F : u \not\sqsubset x\}.$$

If F can be taken to be finite, then X is called an *SFT*, and if it can be taken to be a regular language (a language accepted by a finite-state automaton [7]), then X is called a *sofic shift*. Every SFT is sofic, but the converse is not true. The *SFT approximation of order k* of a subshift $X \subset S^{\mathbb{Z}}$ is the SFT Y_k whose forbidden patterns are precisely the words $S^k \setminus \mathcal{L}_k(X)$. Thus, for all k and $m \geq k$, we have $\mathcal{L}_k(Y_m) = \mathcal{L}_k(X)$.

If X and Y are subshifts, a continuous function $f : X \to Y$ between them is called a *morphism* if $f \circ \sigma = \sigma \circ f$. A surjective morphism is called *factor map*, and a bijective one is called a *conjucagy*, or a *recoding*.

For a point $x \in \mathbb{N}^{\mathbb{Z}}$, we write $\mathcal{L}_n(x)$ for length-n subwords of x, that is, $\mathcal{L}_n(x) = \{u \in \mathbb{N}^n \mid u \sqsubset x\}$, and $\mathcal{L}(x) = \bigcup_n \mathcal{L}_n(x)$. This is called the *language* of x. For any set $A \subset \mathbb{N}^{\mathbb{Z}}$, we write $\mathcal{L}_n(A) = \bigcup_{x \in A} \mathcal{L}(x)$, and define $\mathcal{L}(A)$ as before. A subshift $X \subset S^{\mathbb{Z}}$ is uniquely determined by its language $\mathcal{L}(x)$, which is always *factor-closed* ($vwv' \in \mathcal{L}(X) \implies w \in \mathcal{L}(X)$) and *extendable* ($u \in \mathcal{L}(X) \implies \exists a, b \in S : aub \in \mathcal{L}(X)$).

We often define languages through regular expressions: A word w denotes the language $\{w\}$, and if $L_1, L_2 \subset \Sigma^*$, $L_1 L_2 = \{w_1 \cdot w_2 \mid w_i \in L_i\}$, $L^* = \{w_1 \cdot w_2 \cdots w_n \mid \forall i : w_i \in L\}$ (including the empty word) and $L_1 + L_2 = L_1 \cup L_2$. A *regular* language is one obtained from finite languages with these three operations. We write L^+ as a shorthand for LL^*.

We also define some standard notions of regularity for infinite words, though mainly for context, as they are used in the definition of decidability in [8]. If $L \subset S^*$ we write $L^\omega = \{w_1 w_2 w_3 \cdots \mid \forall i : w_i \in L\}$, and a subset of $S^{\mathbb{N}}$ is *ω-regular* if it is a finite union of languages of the form LK^ω where L

and K are regular. A subset of $S^{\mathbb{Z}}$ is *ζ-regular* if it is a finite union of sets of the form $\{y.x \mid y \in Y^R, x \in X\}$ where X and Y are ω-regular and $Y^R \subset S^{-\mathbb{N}}$ is the set of reversals of points in Y ($-\mathbb{N} = \{-n \mid n \in \mathbb{N}\}$). See [9] for more information.[9]

Note that regular languages, ω-regular languages and ζ-regular languages are all described by a finite expression, and thus we can talk about decision problems where such languages are given as input. All these languages can also be represented by (various kinds of) finite-state automata, and there are conversion algorithms between the representations [9].

Accordingly, for a factor-closed and extendable language L, we write $\mathcal{L}^{-1}(L)$ for the unique subshift X with $\mathcal{L}(X) = L$. We extend this notation to all extendable languages L by $\mathcal{L}^{-1}(L) = \mathcal{L}^{-1}(\mathrm{Fact}(L))$, where $\mathrm{Fact}(L) = \{w \mid \exists u \in L : w \sqsubset u\}$ is the *factor closure* of L. For example, we can write the (sofic) *even shift* [5] with forbidden patterns $1(00)^*01$ as $\mathcal{L}^{-1}((1(00)^*)^*)$, and the (SFT) *golden mean shift* with the single forbidden pattern 11 as $\mathcal{L}^{-1}((100^*)^*)$. The *orbit* of a point $x \in \mathbb{N}^{\mathbb{Z}}$ is $\mathcal{O}(x) = \{\sigma^i(x) \mid i \in \mathbb{Z}\}$. The *orbit closure* $\overline{\mathcal{O}(x)}$ of $x \in S^{\mathbb{Z}}$ is the (topological) closure of the orbit of x in $S^{\mathbb{Z}}$.[10] It is easy to see that $\overline{\mathcal{O}(x)}$ is the smallest subshift containing $x \in S^{\mathbb{Z}}$.

A *substitution* is a function $\tau : A \to B^*$, where $A, B \subset \mathbb{N}$. In our applications, usually $A = \mathbb{N}$, and $B = S$ where S is the alphabet of a subshift being discussed. For $\phi \in \mathbb{N}^{\mathbb{Z}}$ and $\tau : \mathbb{N} \to S^*$, we write

$$\tau(\phi) = \cdots \tau(\phi_{-2})\tau(\phi_{-1}).\tau(\phi_0)\tau(\phi_1)\tau(\phi_2) \cdots .$$

A *weakly transitive point (for X)* is a point $x \in X$ such that $w \sqsubset x$ holds for all $w \sqsubset X$. If X contains a weakly transitive point, we say X is *weakly transitive.* A *doubly transitive point (for X)* is a point $x \in X$ such that every word $w \sqsubset X$ occurs infinitely many times in both tails of x. If X contains a doubly transitive point, then we say X is *transitive.* A weakly transitive subshift need not be transitive. For example, a transitive subshift is either finite or uncountable (since a countable subshift contains an *isolated point*, that is, a singleton open set [10]), but the infinite countable

[9] In [9], they write 'rational' rather than 'regular', and their ω-rational languages can include both finite and infinite words, while they would call our ω-regular languages ω-rational subsets of $S^{\mathbb{N}}$. Also, their languages ζ-rational languages $X \subset S^{\mathbb{Z}}$ are really unions of σ-orbits. Here it makes no difference, as subshifts are also closed under the shift.

[10] See Section 2.2 for the case where x has infinite alphabet.

subshift $\mathcal{L}^{-1}(0^*10^*) = \overline{\mathcal{O}({}^\infty010^\infty)}$, called the *sunny-side-up subshift* is weakly transitive since it is the orbit closure of the point ${}^\infty010^\infty$.

A subshift $X \subset S^{\mathbb{Z}}$ is *minimal* if we have $Y \in \{\emptyset, X\}$ for all subshifts $Y \subset X$. This is equivalent to the fact that X is *uniformly recurrent*, that is,

$$w \sqsubset X \implies \exists n : \forall u \in \mathcal{L}_n(X) : w \sqsubset u.$$

We write $a^{\mathbb{Z}}$ for the point x with $\forall i \in \mathbb{Z} : x_i = a$. We have $\overline{\mathcal{O}(a^{\mathbb{Z}})} = \{a^{\mathbb{Z}}\} = \mathcal{L}^{-1}(a^*)$. This is an example of a (finite) minimal subshift. Typically, minimal subshifts are uncountable. Such examples can be found in [2].

There is a canonical way to remove the isolated points of a space (in our case, a subshift): The *Cantor-Bendixson derivative* of a subshift X is the subshift $X^{(1)}$ obtained by forbidding all words w from X such that $[w]_0$ is a singleton. That is, $X' = X^{(1)}$ is X without its isolated points. This process can be repeated up to any ordinal in an obvious way by transfinite induction, and the *Cantor-Bendixson rank* of a subshift X is the least ordinal λ such that $X^{(\lambda)} = X^{(\lambda+1)}$, where $X^{(\beta)}$ denotes the βth Cantor-Bendixson derivative of X. The Cantor-Bendixson rank of X is denoted by $\mathrm{CB}(X)$, and we call the set $X^{(\mathrm{CB}(X))}$ the *Cantor-Bendixson center* of X. The Cantor-Bendixson center is the unique maximal perfect subspace of X, which is automatically a subshift. A subshift X is countable if and only if its Cantor-Bendixson center is empty. We refer to [10] for details.

We need the first few levels of the *arithmetical hierarchy* [11]. Namely, Σ^0_1 is the set of recursively enumerable languages, that is, languages such that there is an algorithm that halts on words in the language, but not on the ones outside it. The set of complements of Σ^0_1 languages is Π^0_1. A language in $\Delta^0_1 = \Sigma^0_1 \cap \Pi^0_1$ is called *recursive*. Properties that are recursive when encoded into languages (in some natural way, usually safe to leave implicit) are called *decidable*.

We say that a subshift is Π^0_1 or Σ^0_1 if its language is. There are other possible definitions for Π^0_1 subshifts, and we give a few below.

Lemma 1. *For a subshift $X \subset S^{\mathbb{Z}}$, the following are equivalent:*

- *X is Π^0_1.*
- *The set $S^* \setminus \mathcal{L}(X)$ is Σ^0_1.*
- *There exists a Turing machine M which enumerates an infinite list of words $w_1, w_2, \ldots$ such that if $x \in S^{\mathbb{Z}}$, then $x \in X$ if and only if $w_i \not\sqsubset x$ for any $i \in \mathbb{N}$.*

- *There exists a Turing machine M such that the machine M^x (M with oracle $x \in S^{\mathbb{Z}}$) halts if and only if $x \notin X$.*

We call a point $x \in S^{\mathbb{Z}}$ *computable* if there is an algorithm that, given $i \in \mathbb{Z}$, computes x_i. It is well-known that not all Π^0_1 subshifts contain computable points, see for example [12].

2.2. The ruler sequence

The set $\mathbb{M} = \mathbb{N} \cup \{\infty\}$ is the *one-point compactification* of $\mathbb{N}$, where $U \subset \mathbb{M}$ is open if either $U \subset \mathbb{N}$, or $\mathbb{M} \setminus U$ is finite.

We give the space $\mathbb{M}^{\mathbb{Z}}$ the product topology, so that it becomes compact as well. A sequence $\phi \in \mathbb{N}^{\mathbb{Z}}$ is called *recurrent* if every word that appears in ϕ appears infinitely many times in both directions. As usual, if every word appears with bounded gaps, we say ϕ is *uniformly recurrent.* We say ϕ is *Toeplitz* if for all $i \in \mathbb{Z}$ there exists a *period* p such that $\phi_i = \phi_{i+kp}$ for all k. It is easy to see that a Toeplitz sequence is uniformly recurrent. We say ϕ has *unique singularities* if for each $m \in \mathbb{N}$ there exists $k \in \mathbb{N}$ such that

$$\phi_i \geq k \implies \forall j \in [i-m, i+m] \setminus \{i\} : \phi_j \leq k.$$

We give $\mathbb{M}^{\mathbb{Z}}$ the shift action, and for each $\phi \in \mathbb{M}^{\mathbb{Z}}$ obtain a compact space $\overline{\mathcal{O}(\phi)}$ analogously to the case of finite alphabets.[11]

It is easy to verify that ϕ is recurrent if and only if ϕ is doubly transitive in $\overline{\mathcal{O}(\phi)}$, and, more importantly, by Theorem 7 in [1] it is uniformly recurrent if and only if $\overline{\mathcal{O}(\phi)}$ is minimal. The unique singularities property for ϕ means precisely that each point of $\overline{\mathcal{O}(\phi)}$ contains at most one occurrence of ∞, as one can verify by a compactness argument. Of course, ϕ contains only finitely many distinct symbols if and only if $\infty \not\sqsubset \overline{\mathcal{O}(\phi)}$ if and only if $\overline{\mathcal{O}(\phi)}$ is a subshift. In particular, if there are finitely many symbols, having unique singularities is trivial, and recurrence and uniform recurrence correspond to the usual notions. The same observations hold for the space $\mathbb{M}^{\mathbb{N}}$.

We note that neither of uniform recurrence and unique singularities implies the other, as one can easily verify by examples.

We will essentially just need one recurrent sequence with unique singularities in our constructions. The proofs are particularly easy for the *ruler sequence* $\phi \in \mathbb{N}^{\mathbb{N}}$,

$$0\,1\,0\,2\,0\,1\,0\,3\,0\,1\,0\,2\,0\,1\,0\,4\,0\,1\,0\,2\,0\,1\,0\ldots,$$

[11] We note that this is not the usual way to compactify subshifts with infinite alphabets.

that is, $\phi = n_0 n_1 n_2 \ldots$ where n_i is the largest integer k with $2^k | i+1$. This is sequence A007814 in the OEIS database [13].

Note that for all j such that $j \sqsubset \phi$ we have that $\{i \in \mathbb{N} \mid \phi_i = j\}$ is a semilinear set, that is, a finite union of translates of submonoids of $\mathbb{N}$. More precisely, for the ruler sequence ϕ, we have

$$\{i \in \mathbb{N} \mid \phi_i = j\} = \{n 2^{j+1} + 2^j - 1 \mid n \in \mathbb{N}\}.$$

It follows that the ruler sequence is Toeplitz (thus uniformly recurrent) and has unique singularities.

We note some basic combinatorial properties of the subwords of the ruler sequence ϕ, whose proofs we omit. If $\phi_i = \ell = \phi_j$ where $i < j$, then there exists $k \in [i, j]$ such that $\phi_k = \ell + 1$. It follows that if $w \sqsubset X = \overline{\mathcal{O}(\phi)}$, then there exists a unique maximal symbol $w_i = k$ in w. For each $k \in \mathbb{N}$, there is a unique maximally long word $u \in \mathcal{L}(X)$ with $u_{2^k - 1} = k$ and $\forall i : u_i \leq k$, and this u satisfies $|u| = 2^{k+1} - 1$.

Given a word $w \in \mathcal{L}(X)$ with $\infty \not\sqsubset w$, all possible extensions to longer words of X are obtained by the following process: First, take the maximally long word $u \in \mathcal{L}(X)$ with $\max_i u_i = k = \max_i w_i$. We have $u = vwv'$ for a unique choice of v, v', and v (v', resp.) is the unique extension of w in $\mathcal{L}(X)$ of length $|v|$ to the left (of length $|v'|$ to the right, resp.). If $w = u$, then the extensions of w to words in $\mathcal{L}(X)$ of the form awb with $a, b \in \mathbb{N}$ are precisely those of the forms $(k+1)w\ell$ and $\ell w(k+1)$, where $\ell > k+1$.

For example, consider the word $w = 3$. To obtain its extensions, we first deterministically extend it to $u = 010201030102010$. Here, we have $v = v' = 0102010$. We now choose a side on which to write 4. Choosing the left side, we obtain 4010201030102010. On the other side, we can choose any number greater than or equal to 5. Choosing 5 and performing the deterministic extensions, we obtain the word

$$010201030102010\ 4\ 0102010\ 3\ 0102010\ 5\ 0102010301020104010201030102010$$

(where some spaces have been added for clarity). This is the unique maximally long word with maximal symbol 5.

Although $\phi \in \mathbb{M}^{\mathbb{N}}$, we can take its limit points also in $\mathbb{M}^{\mathbb{Z}}$ in the obvious way, and in the orbit closure of ϕ in $\mathbb{M}^{\mathbb{Z}}$, there is a computable point ψ where ∞ does not occur. That is, $\psi \in \mathbb{N}^{\mathbb{Z}}$. One such example is

$$\psi = \ldots 01020103010201040102.010301020105010201030102010 \ldots,$$

which is constructed by starting with 0 and repeating the following steps, keeping the original 0 at coordinate 0:

- Extend on the right by the smallest symbol not yet found in it.
- Extend deterministically, as much as possible.
- Extend on the left by the smallest symbol not yet found in it.
- Extend deterministically, as much as possible.

2.3. Model-checking and halting problems

The following definitions are those from [3], adapted to the case of subshifts and adapted to the two-sided case.[12]

Definition 3. *Let $X \subset S^{\mathbb{Z}}$ be a subshift. The* model-checking problem of X for ζ-rational languages *is to decide, given a ζ-regular language $Y \subset S^{\mathbb{Z}}$, whether $X \cap Y \neq \emptyset$. The* model-checking problem of X for regular languages *is to decide, given a regular language $L \subset S^*$, whether $\mathcal{L}(X) \cap L \neq \emptyset$.*

These are chosen as the natural problems for defining decidability and universality in [3], and we take the same view.

Definition 4. *A subshift $X \subset S^{\mathbb{Z}}$ is* decidable *if its model-checking problem for ζ-regular languages is decidable. It is* universal *if its model-checking problem for finite-state automata is Σ^0_1-complete.*

In the few places where needed, in the one-sided case we use the same definitions, with ω-regular languages in place of ζ-regular ones. In [3], the definition of decidability for $\mathbb{N}$-actions is through Muller automata, which define precisely the ω-regular languages. In an earlier version of this article, we stated that we use Muller automata also over $\mathbb{Z}$, but in fact as noted in [9], Muller automata in their usual form do not really generalize to two-sided points (though there are other automata that do). Thus, we use the standard notions of ω- and ζ-regularity instead.

[12] In [3], these definitions are given not just for subshifts, but for the more general class of “effective symbolic systems”, which in the case of subshifts correspond to ones with a recursive language. Although we do not require recursivity a priori, our examples have recursive languages.

A difference in the definition of [3] is that the regular languages are over clopen sets instead of letters of the alphabet. One way to formulate this is that, instead of asking whether $\mathcal{L}(X) \cap L \neq \emptyset$ for a given regular language L, we ask whether $\mathcal{L}(\phi(X)) \cap L \neq \emptyset$ for a factor map $\phi : X \to Y$ where $Y \subset T^{\mathbb{Z}}$ is a subshift and $L \subset T^*$ is a regular language. If X is recursive, it is easy to see that this does not change the definitions of universality and decidability, since a finite-state machine is smart enough to look at the subshift through a factor map (we skip the easy proof).

It is worth noting that the complexity of a subshift is often defined to be the complexity of its language, or coincides with it. For example, a Π^0_1 *subshift* is a subshift whose language is Π^0_1. For decision problems, "decidable" is a synonym for "recursive", but "decidable subshifts" in the sense of the previous definition form a much smaller class than "recursive subshifts" (decidability implies that the language is recursive, but not vice versa). We will refer to subshifts whose language is recursive exclusively as *recursive subshifts*.

Finally we note that we make no claim that a subshift $X \subset S^{\mathbb{Z}}$ is decidable if and only if the corresponding one-dimensional subshifts $Y \subset S^{\mathbb{N}}$ and $Z \subset S^{-\mathbb{N}}$ consisting of left and right tails of points in X (see Proposition 4) are individually decidable. (In fact we believe this is not the case.)

In practice, to show undecidability, one (many-one) reduces the halting problem of Turing machines to the model-checking problem for regular languages, and usually the reduction will be far from surjective, in the sense that the regular languages produced by the reduction are of a specific form. By considering these different forms, one can get more fine-grained information on the degree to which the subshift is decidable. For this general definition, it is useful to include clopen sets, since, when restricting to small classes of regular languages, we may not be able to simulate factor maps (or even conjugacies), leading to dynamically unnatural decision problems.

Definition 5. *Let $\mathcal{C} = \bigcup_i \mathcal{C}_i$ be a class of regular languages, where each $\mathcal{C}_m$ is over the alphabet $[1, m] \subset \mathbb{N}$. The* model-checking problem for $\mathcal{C}$ and a subshift X *is, given a language L in $\mathcal{C}_m$ for some m and a presentation of X as a union of clopen sets $X = [C_1] \cup \cdots \cup [C_m]$, to decide whether there exists a word $w \in L$ and point $x \in X$ such that $\sigma^i(x) \in [C_{w_i}]$ for all $i \in [0, |w| - 1]$.*

When the union $[C_1] \cup \cdots \cup [C_m]$ is disjoint, this is equivalent to checking $\mathcal{L}(\phi(X)) \cap L \neq \emptyset$ for a factor map $\phi : X \to Y$ where Y is some subshift, and universality of a subshift means that its model-checking problem for the whole class of regular languages is Σ^0_1-complete. It is convenient to allow the union to be non-disjoint, and to even allow $C_i = C_j$ for some $i \neq j$, as this makes it easier to show that the halting problems we define correspond to model-checking problems. One could also define these problems for partitions only (as was done in an earlier draft of the paper), and in the subshift setting it would also make sense to consider expansive partitions only.

Definition 6. *The* star-free languages over $[1,m]$ *are the closure of finite languages over* $[1,m]$ *under concatenation, complementation (with respect to* $[1,m]^*$*) and union.*[13] *By an* elementary piecewise testable language over $[1,m]$*, we mean a language of the form* $a_1S^*a_2S^*a_3S^* \cdots S^*a_k$ *where* $S = [1,m]$*,* $a_i \in S$ *for all* $i \in [1,k]$*, and a* piecewise testable language over $[1,m]$ *is any Boolean combination of elementary piecewise testable languages. A* local language over $[1,m]$ *is a language of the form* $(AS^* \cap S^*B) \setminus S^*FS^*$*, where* $A, B \subset S$ *and* $F \subset S^2$*. By a* renewal language *we mean one of the form* UW^*V *where* $U, W, V \subset S^*$ *are finite sets of words.*

We occasionally use the following shorthands for these: 'EPT', 'PT', 'Loc', 'SF', 'Ren', 'Reg' and 'CF' stand for 'elementary piecewise testable', 'piecewise testable', 'local', 'star-free', 'renewal', 'regular' and 'context-free' languages, respectively.

In the case of star-free languages and piecewise testable languages, it is important to parametrize the class of languages with the alphabet $[1,m]$, and consider languages over $[1,m]$ when the space is fully covered by m clopen sets, because the alphabet $S = [1,m]$ plays a special role in the definitions – for local languages and renewal languages, this is not necessary. Because $S^* = \emptyset^C$, piecewise testable languages and local languages are both star-free. Because $S^*a = \bigcup_{b \in S} bS^*a \cup a$, we see that also languages of the form $S^*a_1S^*a_2S^*a_3S^* \cdots S^*a_kS^*$ are piecewise testable.

Example 1: An example of a star-free language over the alphabet $[1,2]$ is $L = 21^*2 = 2(\emptyset^C 2 \emptyset^C)^C 2$. The language L is not local, but 21^*3 over the alphabet $[1,3]$ is, by setting $A = \{2\}$, $B = \{3\}$, $F = \{12, 22, 31, 32, 33\}$. Neither of these is elementary piecewise testable, but $2(1+2)^*2$ is over the alphabet $[1,2]$. These are all renewal languages (over any alphabet), while $2(1+2)^*2(1+2)^*2$ is elementary piecewise testable and star-free but not local

[13] That is, they are defined by generalized regular expressions without using Kleene star.

$$\begin{tikzcd}[column sep=small] & & \mathsf{Loc} \arrow[d, phantom, "\rotatebox{-90}{$\subseteq$}"] & \mathsf{Ren} \arrow[d, phantom, "\rotatebox{-90}{$\subseteq$}"] \\ \mathsf{EPT} \arrow[r, phantom, "\subseteq"] & \mathsf{PT} \arrow[r, phantom, "\subseteq"] & \mathsf{SF} \arrow[r, phantom, "\subseteq"] & \mathsf{Reg} \end{tikzcd}$$

Figure 1: Inclusions between the classes of languages we define.

or a renewal language. The language $2(11)^*2$ is renewal, but is not EPT, PT, Loc or SF. △

From the observations above, we obtain the inclusions in Figure 1; note that $\mathcal{C} \subset \mathcal{D}$ for language classes $\mathcal{C}$ and $\mathcal{D}$ implies that the model-checking problem for $\mathcal{C}$ many-one reduces to that of $\mathcal{D}$.

The star-free languages are a well-studied class of regular languages, and were also essentially studied in the context of dynamical systems in [8] (an earlier version of [3]), as definability in temporal logic corresponds to star-freeness [14] (although this connection was not made explicit). "Elementary piecewise testable" is not a standard term, but piecewise testable languages are again well-studied (although there are many variations of the basic idea in the literature). Local languages are important in the theory of regular languages, as they are to regular languages what vertex shifts are to sofic shifts. We could have equally well used *locally testable languages*, where we specify the sets of legal length-k prefixes, suffixes and subwords. These languages correspond more directly to SFTs, but the difference is mostly irrelevant for dynamical considerations. We do not know of an occurrence of renewal languages in the literature, but they are an obvious "language version" of the well-studied class of sofic shifts called *renewal systems.*[14]

Another approach to decidability is to consider problems of reachability between clopen sets. We talk about the following problems:

Definition 7. *In the following problems, defined for a subshift X, we are given clopen sets $[C], [D], [E], [F]$ and numbers k, m, or a subset of these inputs, and need to decide whether there exists a point $x \in X$, $j \in \mathbb{Z}$ and $A \subset [1, j-1]$ with $|A| = k$, or a subset of these, with particular properties.*

- *In the* undirected halting problem*, properties $x \in [C]$ and $\sigma^j(x) \in [D]$.*
- *In the* halting problem*, properties $j \in \mathbb{N}$, $x \in [C]$ and $\sigma^j(x) \in [D]$.*

[14] One can imagine many other such language versions as well, and we do not claim this to be the "correct" one – for the purpose of this article, it is mainly a mnemonic.

- *In the* halting problem along a clopen set*, properties* $j \in \mathbb{N}$*,* $x \in [C]$*,* $\sigma^j(x) \in [D]$ *and* $\forall i \in [1, j-1] : \sigma^i(x) \in [E]$*.*
- *In the* counting problem*, properties* $j \in \mathbb{N}$*,* $x \in [C]$*,* $\sigma^j(x) \in [D]$*,* $\forall i \in [1, j-1] \setminus A : \sigma^i(x) \in [E]$ *and* $\forall i \in A : \sigma^i(x) \in [F]$*.*
- *In the* modular halting problem*, properties* $j \in \mathbb{N}$*,* $j \equiv k \bmod m$ *with* $x \in [C]$ *and* $\sigma^j(x) \in [D]$*.*
- *In the* modular halting problem along a clopen set*, properties* $j \in \mathbb{N}$*,* $j \equiv k \bmod m$ *with* $x \in [C]$*,* $\sigma^j(x) \in [D]$ *and* $\forall i \in [1, j-1] : \sigma^i(x) \in E$*.*

The halting problem was also defined in [3], and for Turing machines, considered as a dynamical system (with moving tape or moving head, see [15]), it essentially corresponds to the usual halting problem. It is easy to see that the halting problem many-one reduces to the model-checking problem for all of EPT, PT, Loc, SF, Ren and Reg, since, using a clopen cover with m clopen sets, of which the first corresponds to $[C]$ and second to $[D]$, it amounts to the model-checking problem for $1[1, m]^*2$, which is in all of the language classes.

The halting problem along a clopen set many-one reduces to the model-checking problem for any of our classes, except EPT.[15] This is because it corresponds to the model-checking problem for the language 12^*3, by again naming clopen sets suitably, and this language is Loc, Ren and Reg trivially, and a bit of work shows that it is also PT and thus SF: We have

$$12^*3 = \{1, 2, 3\}^* \cap 1[1, m]^* \cap [1, m]^*3 \cap ([1, m]^+([1, m] \setminus \{2\})[1, m]^+)^C,$$

and on the right, every individual expression can be easily written as a Boolean combination of EPT languages.

The counting problem is a special case of the model-checking problem for PT as well (and thus reduces to SF): For $a_1, a_2, a_3, a_4 \in S$, the language $L \cap L' \cap L_k \cap L_{k+1}^C$ is piecewise-testable, where $L = \{a_1, a_2, a_3, a_4\}^*$, $L' = a_1(S^+(a_1 + a_2)S^+)^C a_2$ (shown to be PT similarly as above), and $L_i = (S^*a_4)^i S^*$ is the language of words containing at least i occurrences of the symbol a_4.

Finally, the modular halting problem (along a clopen set or not) is a special case of the model-checking problem for renewal languages: For $a_1, a_2, a_3 \in$

[15] Theorem 3 shows it does not reduce to EPT.

S, set $u = a_1 a_3^k$, $v = a_2$ and $w_1 = a_3^m$. Then uw_1^*v is a renewal language. We note that this is not a special case of the model-checking problem for star-free languages, and indeed should not be: a well-known characterization of star-freeness of a language is that it is recognized by a finite state machine that does no modular counting [16].

2.4. The generating order

The following object is very useful in studying subshifts, and was introduced in [10] in the context of countable multidimensional SFTs.

Definition 8. *The* pattern preorder *of a subshift $X \subset S^{\mathbb{Z}}$ is the order $x \leq y \iff \mathcal{L}(x) \subset \mathcal{L}(y)$.*

We will, instead, talk about the *subpattern poset*, meaning the preordered set with elements X and the pattern preorder as the preorder. If $x \leq y \leq x$, we write $x \sim y$.

We wish to extend this relation to finite words as well. An obvious way to do this would be to define $u \leq v \iff \mathcal{L}(u) \subset \mathcal{L}(v)$, so that $u \leq v \iff u \sqsubset v$. However, the following definition is more useful:

Definition 9. *The* generating (pre-)order *of a subshift X has elements $\mathcal{L}(X)$ and preorder $u \leq_X v \iff (\forall x \in X : v \sqsubset x \implies u \sqsubset x)$.*

By compactness, $u \leq_X v$ means that there exists $k \in \mathbb{N}$ such that

$$\forall x \in X : x_{[0,|v|-1]} = v \implies u \sqsubset x_{[-k,|v|+k-1]}.$$

We again talk about the generating poset to mean the preordered set $\mathcal{L}(X)$ with the generating preorder.

Abusing notation, for a word u and a point x, we write $x \leq_X u$ if $v \leq_X u$ for all $v \sqsubset x$. When X is clear from context, in particular when u and v are explicitly chosen from X, we write $u \leq v$ for $u \leq_X v$.

Lemma 2. *Let X be a Π_1^0 subshift. Then, given u, v it is semidecidable whether $u \leq_X v$. In fact, there is an algorithm that, given u, v with $u \leq_X v$, computes $h_{u,v}$ such that*

$$|w| \geq |v| + 2h_{u,v} \wedge w_{[h_{u,v},h_{u,v}+|v|-1]} = v \implies u \sqsubset w.$$

Proof. If $u \leq v$, then all long enough legal patterns of X containing v in the center contain u as well. By compactness, the same is true for some SFT approximation of X, which we eventually find by enumerating forbidden patterns of X. This yields $h_{u,v}$ as a side-product. □

The poset induced by the $\leq_X$-order on words is not a conjugacy invariant. However, $\leq_X$ extends naturally to clopen sets by $[C] \leq_X [D] \iff x \in [D] \implies \mathcal{O}(x) \cap [C] \neq \emptyset$, and the poset of clopen sets is obviously conjugacy invariant. It is a direct corollary of the previous lemma that $[C] \leq_X [D]$ is semidecidable for given cylinders $[C]$ and $[D]$.

3. Quasiminimality and undecidability

In this section, we present our constructions of universal subshifts mentioned in Section 1 and prove their correctness. The decidability results complementing them are given in Section 4. To get up to speed, we begin with some simpler examples. First, we give a few examples of quasiminimal but non-minimal subshifts which are in a sense already well-known: subshifts generated by letter-to-word substitutions. Then, we give a universal subshift with finitely many minimal subshifts, but infinitely many subshifts in total, already proving Proposition 1 and refuting Conjecture 1. Then, we present our main constructions: universal quasiminimal subshifts. We conclude this section with a minimal subshift whose model-checking problem for context-free languages is Σ^0_1-complete.

3.1. Non-universal quasiminimal examples

In this section, we consider classical symbol-to-word substitutions on a finite alphabet S. It is well-known that a *primitive* substitution, that is, a substitution $\tau : S \to S^*$ such that

$$\exists n : \forall a, b \in S : b \sqsubset \tau^n(a),$$

generates a minimal subshift (in the sense defined below). A non-primitive substitution does not necessarily generate a minimal subshift, but it usually[16] generates a quasiminimal subshift. We show some examples, and sketch the proofs of their quasiminimality. These examples are very similar to the universal examples we construct later.

[16] Below, we give a proof under some assumptions on the substitution.

Definition 10. *Let $\tau : S \to S^*$ be a substitution, define the subshift $X_\tau \subset S^{\mathbb{Z}}$ that it* generates *by*

$$x \in X_\tau \iff \forall j, k \in \mathbb{N} : \exists a \in S, \ell \in \mathbb{N} : x_{[j,k]} \sqsubset \tau^\ell(a).$$

First, we give an example of a countable quasiminimal system which is not a union of minimal systems.

Example 2: Let τ be the substitution $(0 \mapsto 0; 1 \mapsto 010)$. Then we have $\tau^n(0) = 0$ and $\tau^n(1) = 0^n10^n$ for all $n \in \mathbb{N}$. Clearly, the substitution then generates the sunny side up subshift $\overline{\mathcal{O}({}^\infty 010^\infty)}$. Since this subshift is the union of the orbits of ${}^\infty 010^\infty$ and ${}^\infty 0^\infty$, it is countable, and its only proper nontrivial subsystem is $\mathcal{L}^{-1}(0^*)$. $\triangle$

The points need not be eventually periodic:

Example 3: Let τ be the substitution $(0 \mapsto 00; 1 \mapsto 11; 2 \mapsto 20; 3 \mapsto 2301)$. Clearly, the only points where infinitely many central patterns are subwords of $\tau^n(0)$, $\tau^n(1)$ or $\tau^n(2)$ are $0^{\mathbb{Z}}$ and $1^{\mathbb{Z}}$. Thus, we only need to find out the limit points of $\tau^n(3)$. We have

$$\tau^n(3) = 20^{n-1}20^{n-2}2 \cdots 202\ 3\ 0100110^{2^2}1^{2^2}0^{2^3}1^{2^3} \cdots 0^{2^{n-1}}1^{2^{n-1}}.$$

Since there is only one occurrence of the symbol 3 in this word, there is a unique point in X_τ where 3 occurs. In a proper subsystem, then, a finite subset of the symbols $\{0, 1, 2\}$ occurs. The proper subshifts of X_τ can be seen to be the countable sofic shifts

$$\mathcal{L}^{-1}(0^*), \mathcal{L}^{-1}(1^*), \mathcal{L}^{-1}(0^*20^*), \mathcal{L}^{-1}(0^*1^*), \mathcal{L}^{-1}(1^*0^*)$$

and their finite unions. In particular, X_τ itself is countable. $\triangle$

The following uncountable example is the basis of the more general construction in Lemma 4.

Example 4: Let τ be the substitution $(0 \mapsto 00; 1 \mapsto 101)$. Then one can show by induction that $\tau^n(0) = 0^{2^n}$ and

$$\tau^n(1) = \tau^{n-1}(1)0^{2^n}\tau^{n-1}(1) = 10^{2^{\phi_0}}10^{2^{\phi_1}}10^{2^{\phi_2}}1 \cdots 10^{2^{\phi_k}}1$$

where ϕ is the ruler sequence and $k = 2^n - 2$. Clearly, $\mathcal{L}^{-1}(0^*)$ is a subsystem. We claim that it is in fact the only subsystem.

Namely, we show that if $x \in X_\tau$ contains the symbol 1, then it is weakly transitive. Suppose $x_0 = 1$ (by shifting x if necessary). For any k, by definition of X_τ we must have that $x_{[-k,k]} \sqsubset \tau^n(1)$ for some n. We note that the ruler sequence has unique singularities, and thus does not contain subwords of the form ab where $a, b \in \mathbb{N}$ and both are arbitrarily large – in fact, every second symbol in it is 0, and thus every second gap between two symbols 1 is of length 1. Taking a suitable k and considering the word $\tau^n(1)$ such that $x_{[-k,k]} \sqsubset \tau^n(1)$, we see that either $x_{-2} = 1$ or $x_2 = 1$. For concreteness, suppose we are in the second case. Again, due to unique singularities, the ruler sequence does not contain words of the form $a0b$ where both a and b are arbitrarily large, and in fact one of them is always 1. This means, again taking suitable k and looking at the corresponding $\tau^n(1)$, that either $x_{-4} = 1$ or $x_{2+4} = 1$. For concreteness, suppose we are in the second case. The only extension of 01 to the right by one symbol in the ruler sequence is 010, so again looking at $\tau^n(1)$ for large n, we see that $x_8 = 1$, and $x_{[0,8]} = 101000101$.

We repeat this deduction infinitely many times: We look at the finite part of x already considered by the process, and observe, using the unique singularities of the ruler sequence, that it cannot be the case that all the 1s in x are in this part, and in fact the already filled part must be continued by $0^{2^\ell}1$ to the right or 10^{2^ℓ} to the left, for suitable ℓ. We then extend this word deterministically using the properties of the ruler sequence given in Section 2.2.

If we eventually fill the whole point x with this deducion process (so that we find the new 1 from both the left and the right side infinitely many times), then x corresponds in a one-to-one fashion to a point $\psi \in \mathbb{N}^\mathbb{Z}$ in the orbit closure of the ruler sequence. If only one side is filled, then one can check that x is in the orbit closure of either

$$^\infty 0.10^{2^{\phi_0}}10^{2^{\phi_1}}10^{2^{\phi_2}}10^{2^{\phi_3}}1\ldots$$

or

$$\ldots 10^{2^{\phi_3}}10^{2^{\phi_2}}10^{2^{\phi_1}}10^{2^{\phi_0}}1.0^\infty,$$

and in either case it generates the whole subshift X_τ. △

We believe that all substitutions generate quasiminimal systems. In the following, we show this under an additional condition: Let $\tau : S \to S^+$ be a substitution and let $S_\ell = \{a \in S \mid |\tau^n(a)| \to \infty\}$, the set of *long symbols*. If for some m,

$$w \sqsubset X_\tau \wedge |w| \geq m \implies S_\ell \cap \mathcal{L}_1(w) \neq \emptyset,$$

then we say *long symbols are syndetic in* X_τ.

Proposition 5.5 of [17] shows that the above condition holds automatically if X_τ contains no periodic points, and in Proposition 5.6 they show that if this condition holds, then the number of minimal subshifts contained in X_τ is at most $|S|$. We show a similar result for the set of all subshifts.

Proposition 2. *If $\tau : S \to S^+$ is a substitution and long symbols are syndetic in X_τ, then X_τ is quasiminimal.*

Proof. For each $n \geq 1$ and $x \in X_\tau$ there exists $y \in X_\tau$ such that $x = \tau^n(y)$ (up to shifting x). Namely, each subword of x occurs in $\tau^k(a) = \tau^n(\tau^{k-n}(a))$ for arbitrarily large k and a long symbol a, and we obtain y as a limit point of $\tau^{k-n}(a)$. Let m be such that every word of length m in X_τ contains a long symbol. Let $Z \subset X_\tau$ be a subshift, and for each $n \in \mathbb{N}$, associate to Z the following set:

$$W_n(Z) = \{w \in S^{m+1} \mid \exists x, y \in X_\tau : x \in Z \wedge x = \tau^n(y) \wedge w \sqsubset y\}.$$

Note that for each n, W_n can have at most $2^{|S|^{m+1}}$ distinct values. Thus, if $Z_1, Z_2, \ldots, Z_{2^{|S|^{m+1}}+1}$ are subsystems of X_τ, we must have two indices $i \neq j$ such that $W_n(Z_i) = W_n(Z_j)$ for infinitely many n. It is enough to show that this implies $\mathcal{L}(Z_i) = \mathcal{L}(Z_j)$, since subshifts with the same language are equal.

Thus, suppose that Z, Y are subshifts of X_τ with $W_n(Z) = W_n(Y)$ for arbitrarily large n. Let $u \sqsubset Z$ be arbitrary. Let k be such that $|\tau^k(a)| \geq |u|$ whenever a is a long symbol, and choose $n \geq k$ such that $W_n(Z) = W_n(Y)$. Since $\tau(S) \subset S^+$, τ can only increase the length of words, and thus we have $|\tau^n(a)| \geq |u|$ for all long symbols a.

Choose a point $z \in Z$ such that $u \sqsubset z$, and let $z = \tau^n(y)$ where $y \in X_\tau$. Since long symbols are syndetic in X_τ and $|\tau^n(a)| \geq |u|$ for long symbols a, it is easy to see that there is a subword w of y of length $m+1$ with $u \sqsubset \tau^n(w)$. By the assumption $W_n(Z) = W_n(Y)$, w is also a subword of some $y' \in X_\tau$ such that $\tau^n(y) \in Y$, so $u \sqsubset Y$.

This shows $u \sqsubset Y$, and since $u \sqsubset Z$ was arbitrary, we have $\mathcal{L}(Z) \subset \mathcal{L}(Y)$. Symmetrically, we obtain $\mathcal{L}(Y) \subset \mathcal{L}(Z)$, which concludes the proof. □

The upper bound we obtain for the number of subsystems $2^{S^{m+1}}$, where m is the bound for the length of the gap between two long symbols. In the case that all symbols are long, we have $m = 1$, and obtain the upper bound

$2^{|S|^2}$ for the number of subsystems. The next example shows that this is in the right ballpark.[17]

Example 5: Let $k \in \mathbb{N}$, let $S = \{a, b_1, \ldots, b_k\}$, and define the substitution τ with $\tau(b_i) = b_i^2$ for all $i \in [1,k]$ and $\tau(a) = w$, where $w \in \{b_1, \ldots, b_k\}$ is such that $b_i b_j \sqsubset w$ for all $i, j \in [1,k]$. It is easy to see that $X_\tau = \mathcal{L}^{-1}(\bigcup_{i,j} b_i^* b_j^*)$.

Choose a subset $K \subset [1,k]$, and a subset

$$J \subset \{(i,j) \in [1,k]^2 \mid i \neq j\}$$

such that $(i,j) \in J \implies \{i,j\} \subset K$. For such (K,J), let

$$Y_{K,J} = \mathcal{L}^{-1}(\bigcup_{i \in K} b_i^* \cup \bigcup_{(i,j) \in J} b_i^* b_j^*).$$

We note that we can determine the sets K and J from $Y_{K,J}$, and if Y is a subshift of X_τ, then $Y = Y_{K,J}$ for some (K,J) with the above properties. Thus, to compute the number $B(k)$ of subsystems of X_τ, we only need to compute the number of pairs (K,J) with these properties. The number of such pairs is just the number of directed graphs whose vertices form a subset of $[1,k]$.

Letting $A(j) = 2^{j(j-1)}$ be the number of directed graphs with vertices $[1,j]$, we have

$$B(k) = \binom{k}{0} A(0) + \binom{k}{1} A(1) + \binom{k}{2} A(2) + \cdots + \binom{k}{k} A(k).$$

Of course, $B(k) \geq A(k) = 2^{j(j-1)}$, so there exists a substitution on an alphabet of k symbols with $B(k-1) \geq 2^{(k-1)(k-2)}$ subsystems. $\triangle$

In the OEIS database, $B(i)$ is the sequence A135756, and the first few values are

$$B(0) = 1, B(1) = 2, B(2) = 7, B(3) = 80, B(4) = 4381, B(5) = 1069742.$$

We note that the upper bound for the number of subshifts of X_τ which are unions of minimal subshifts is $2^{|S|}$ by the result of [17]. For $|S| = 6$, the

[17] We only look at the case $m = 1$ for simplicity, but one can add short symbols between long symbols in the word $\tau(a) = w$ in the example, to get roughly $2^{|S_\ell|^2 |S \setminus S_\ell|^{m-1}}$ subshifts, for a partition $S = S_\ell \cup (S \setminus S_\ell)$.

number of such subsystems of X_τ is at most $2^6 = 64$, but one can have $B(5) = 1069742$ subsystems in total by the previous example.

Example 3 and Example 4 both have the property of syndetic long symbols, as all symbols are long. Example 2 does not have this property, as 0 is not a long symbol, but clearly the subshift does not change if the image of 0 is changed to 00. However, we believe there is no substitution with syndetic long symbols generating a subshift conjugate to X_τ where

$$\tau = (0 \mapsto 0; 1 \mapsto 10; 2 \mapsto 021),$$

where the point generated by the symbol 2 is

$$...0000002110100100010000100000...$$

Of course, even though long symbols are not syndetic, it is not very hard to show that this subshift is quasiminimal. In general, one might be able to obtain a proof for general substitutions by analysing the proof of Proposition 5.5 in [17] in more detail.

We show that the model-checking problem for regular languages is decidable for these systems. It seems plausible that this can be extended considerably, see Section 5.

Lemma 3. *Given a substitution $\tau : S \to S^*$, a symbol $a \in S$ and a regular language L, it is decidable whether $\tau^n(a) \in L$ for some $n \in \mathbb{N}$. In particular, it is decidable whether $\mathcal{L}(X_\tau) \cap L = \emptyset$ for a given regular L.*

Proof. We may assume $L \subset S^*$. Let A be a nondeterministic finite state automaton for L with state set Q, initial state q_s, final state q_t and transition function $\delta \subset Q \times S \times Q$, and extend δ to a relation $\delta \subset Q \times S^n \times Q$ for all n in the usual way:

$$(q, w, q') \in \delta \iff (q, w_0, q_0), (q_0, w_1, q_1), \cdots, (q_{n-2}, w_{n-1}, q') \in \delta$$

For all $s \in S$, let $R \in ((2^{Q \times Q})^S)^{\mathbb{N}}$ be defined by

$$(q, q') \in (R_i)_s \iff \delta(q, \tau^i(s), q').$$

We can compute R_i for each i easily from the definition of δ, and R_i takes its value in the finite set $(2^{Q \times Q})^S$. The set $(2^{Q \times Q})^S$ is finite, so let t, p be such that $R_t = R_{t+p}$. Then, $R'_t = R_{t'+p}$ for all $t' > t$ as well: writing $w = \tau^{t'-t}(s)$, we have

$$\begin{aligned}(q, q') \in (R_{t'})_s &\iff \delta(q, \tau^{t'}(s), q') \\ &\iff \delta(q, \tau^{t}(w), q') \\ &\iff \delta(q, \tau^{t+p}(w), q') \\ &\iff \delta(q, \tau^{t'+p}(s), q') \\ &\iff (q, q') \in (R_{t'+p})_s,\end{aligned}$$

where the second $\iff$ follows because $R_t = R_{t+p}$, so that

$$\delta(q, \tau^t(a), q') \iff \delta(q, \tau^{t+p}(a), q')$$

for all $a \in S$, and the third $\iff$ because this extends to all words $w \in S^*$ by the inductive definition of $\delta \subset Q \times S^n \times Q$.

Now, to check whether $\tau^n(a) \in L$ for some n, it is enough to check whether $(q_s, q_t) \in (R_i)_a$ for some $i \in [0, p + t - 1]$. The second claim is proved by checking whether we have $\tau^n(a) \in S^*LS^*$ for some n and $a \in S$. □

3.2. Universal non-quasiminimal examples

We present our (non-quasiminimal) example of a universal recursive subshift with finitely many minimal subsystems.

Proposition 3. *There exists a recursive subshift $X \subset \{0, 1, 2, 3\}^{\mathbb{Z}}$ which is contained in a countable SFT, has finitely many minimal subsystems, and has a Σ^0_1-complete halting problem. Every minimal subsystem of X consists of a single unary point.*

Proof. Enumerate the deterministic Turing machines as $T_1, T_2, T_3, \ldots$. We define a subshift by $X = \overline{\mathcal{O}(\{x_i \mid i \geq 1\})}$, where

$$x_i = {}^\infty 0.1^i 2^{i+h(i)} 3^\infty,$$

if the machine T_i halts exactly after $h(i) \in \mathbb{N}$ steps, and $x_i = {}^\infty 0.1^i 2^\infty$ if it never halts.

This is a recursive subshift: First, all words $a^* b^*$ for $a \leq b \in \{0, 1, 2, 3\}$ are in $\mathcal{L}(X)$. Let $i, j, k, \ell \geq 1$ be arbitrary. The word $0^i 1^j 2^k$ is in $\mathcal{L}(X)$ if and only if T_j does not halt in the first k steps, which is decidable. The word $0^i 1^j 2^k 3^\ell$ is in $\mathcal{L}(X)$ if and only if T_j halts exactly after $k - j$ steps, which is decidable. The word $1^j 2^k 3^\ell$ is in $\mathcal{L}(X)$ if and only if some word of the form $01^{j'} 2^k 3^\ell$ is in $\mathcal{L}(X)$, where $j \leq j' \leq k$, which we can check by running all the machines $T_{j'}$ for at most k steps.

The subshift X is contained in the countable SFT $\mathcal{L}^{-1}(0^* 1^* 2^* 3^*)$, because each of the points x_i is in this subshift. The only periodic points in X are the points ${}^\infty a^\infty$ for $a \in \{0, 1, 2, 3\}$. Since a minimal subshift is either finite or uncountable, the only minimal subsystems of X are these singleton subshifts.

The undirected halting problem of X is Σ^0_1-complete because solving the halting problem for the clopen sets $[01^i 2]_0$ and $[3]_0$ is equivalent to solving the halting problem of T_i. □

We make a few remarks about this subshift, omitting the easy proofs.

1. This subshift has Cantor-Bendixson rank 3. Countable subshifts of Cantor-Bendixson rank 1 are finite, and those of rank 2 are sofic [18], so this is the minimal possible rank for a countable universal subshift.
2. By forbidding the single letter 3 from the subshift X, we obtain a system Y whose language is not decidable, since $01^k 2 \sqsubset Y$ if and only if T_k never halts. Thus, while X is recursive, one could say it is not *hereditarily recursive*, as it contains a Π^0_1 subshift which is not recursive. (Compare this with Corollary 2.)
3. The subshift X has only finitely many minimal subshifts, but more than one. By using the points
$$x_i = {}^\infty 0.10^i 20^{i+h(i)} 30^\infty$$
in the proof, the enveloping countable SFT of Cantor-Bendixson rank 4 changes into a countable sofic shift with the same CB-rank, and there will be only one minimal subsystem (which is a fixed point).
4. We do not have to use minimal subshifts consisting of a single fixed point – we can use a periodic point, or more generally any minimal subshift: If, in the points $x_i = {}^\infty 0.10^i 20^{i+h(i)} 30^\infty$, we replace the maximal subwords of the form 0^k by (any!) subwords of length k of a minimal subshift Y over an alphabet disjoint with $\{1, 2, 3\}$, it is easy to check that the only minimal subsystem of $\overline{\mathcal{O}(\{x_i \mid i \geq 1\})}$ is Y. Choosing the minimal subshift and the words suitably, the subshift can be made recursive as well.
5. A classical tool for studying minimal systems are the Bratteli-Vershik systems. It was proved in [19] that every minimal system is conjugate

to such a system. In fact, the result of [19] applies more generally to systems containing exactly one minimal subshift, and thus to our example. Such systems are called *essentially minimal systems.* The classes of essentially minimal systems and quasiminimal systems are incomparable.

6. In Proposition 9 of [3], it is shown that if the limit set of a symbolic dynamical system is a finite union of minimal systems, then it is decidable. As we are concerned with two-way subshifts, the limit set of a subshift is equal to the subshift itself. We note, however, that the *asymptotic set* (the union of limit points of individual configurations) and its *nonwandering set* (the points whose neighborhoods all return to themselves) of X are both finite unions of minimal systems: in fact, these sets are finite, and can again be taken to be singletons.

As discussed in the introduction, our system is a $\mathbb{Z}$-system, and one could also ask whether Conjecture 1 is true for $\mathbb{N}$-systems, where a priori there are more subsystems. Since minimal systems are clearly surjective, and minimality is a property of the language of a subshift (uniform recurrence), the following proposition shows that Proposition 3 resolves the case of $\mathbb{N}$-actions too. If $X \subset S^{\mathbb{N}}$ is a one-sided subshift, we say it is *surjective* if the left shift $\sigma : X \to X$ is surjective.

Proposition 4. *For any subshift $X \subset \mathbb{N}^{\mathbb{Z}}$ let*

$$c(X) = \{y \in \mathbb{N}^{\mathbb{N}} \mid \exists x \in \mathbb{N}^{-\mathbb{N}} : x.y \in X\}.$$

For any surjective subshift $X \subset \mathbb{N}^{\mathbb{N}}$, let

$$e(X) = \{x \in \mathbb{N}^{\mathbb{Z}} \mid \forall i \in \mathbb{Z} : x_{[i,\infty)} \in X\}.$$

These operations preserve the language of the subshift, and thus $c(e(X)) = X$ for any surjective $\mathbb{N}$-subshift, and $e(c(X)) = X$ for any $\mathbb{Z}$-subshift.

The operations c and e and their correspondence are well-known, although we do not know an explicit reference for the precise statement above.

3.3. Uncountable universal quasiminimal examples

We now move on to our quasiminimal examples in the case of $\mathbb{Z}$-actions, from which one can obtain results in the case of $\mathbb{N}$-actions from the previous proposition, when only surjective subsystems are considered. The case of $\mathbb{N}$-actions and non-surjective subsystems is dealt with in Section 4.2. First, we give our example of a universal (uncountable) quasiminimal subshifts, giving our first proof of Theorem 2. In this case, we can construct a subshift whose halting problem along a clopen set is undecidable, and thus so is the model-checking problem for local languages, piecewise testable languages, star-free languages and renewal languages. Our example has only one proper subshift, which can be chosen rather freely.

Theorem 3. *For any recursive infinite minimal subshift Y, there exists a recursive transitive uncountable quasiminimal subshift X for which the halting problem along a clopen set is Σ^0_1-complete, such that the only nontrivial subsystem of X is Y.*

To prove Theorem 3, we need a few lemmas. The first is a method of constructing quasiminimal subshifts.

Lemma 4. *Let $Y \subset S^{\mathbb{Z}}$ be a quasiminimal subshift. Let $\phi \in \mathbb{N}^{\mathbb{Z}}$ be uniformly recurrent with unique singularities, and let $\tau : \mathbb{N} \to a\mathcal{L}(Y)$ be a substitution where $|\tau(n)| \overset{n\to\infty}{\longrightarrow} \infty$, where a is a symbol not in S. Then $X = \overline{\mathcal{O}(\tau(\phi))} \subset (S \cup \{a\})^{\mathbb{Z}}$ is transitive and quasiminimal, and in fact every proper subsystem of X is a subsystem of Y.*

Proof. If Z is a subshift of X where a does not occur, then clearly it is also a subshift of Y. Thus, we only need to show that there are finitely many subshifts of X where a does occur. In fact, we show the stronger fact that $a \in x \implies \overline{\mathcal{O}(x)} = X$. For this, suppose that $a \in x$.

If a occurs infinitely many times in both tails of x (that is, $\{i \in \mathbb{Z} \mid x_i = a\}$ is unbounded from both above and below), then by $|\tau(n)| \overset{n\to\infty}{\longrightarrow} \infty$ and a compactness argument, we have $x \in \mathcal{O}(\tau(\phi'))$ for some $\phi' \in \overline{\mathcal{O}(\phi)} \cap \mathbb{N}^{\mathbb{Z}}$. Namely, since $x \in \overline{\mathcal{O}(\tau(\phi))}$, for any j, j' such that $x_j = x_{j'} = a$, we find i, i' such that $x_{[j,j']} = \tau(\phi_{[i,i']})a$. There are finitely many choices for the word $\phi_{[i,i']}$ for each pair j, j', so letting $j \to -\infty$ and $j' \to \infty$ and passing to a suitable subsequence, we obtain $x \in \mathcal{O}(\tau(\phi'))$ for some $\phi' \in \overline{\mathcal{O}(\phi)}$.

Because ϕ is uniformly recurrent, ϕ' contains all its finite patterns, and thus $\overline{\mathcal{O}(x)}$ contains all finite patterns of X, which implies $\overline{\mathcal{O}(x)} = X$.

Next, suppose a occurs infinitely many times in one tail of x, but not the other. These cases are (more or less) symmetric, so we suppose $x_0 = a$, $x_i \neq a$ for all $i < 0$, and $\{i \in \mathbb{N} \mid x_i = a\}$ is unbounded from above. Now, a compactness argument like the one above shows that there exists a point $\phi' \in \overline{\mathcal{O}(\phi)}$ such that $\phi'_{-1} = \infty$, $x_{\mathbb{N}} = \tau(\phi'_{\mathbb{N}})$, and $x_{(-\infty,-1]}$ is some limit point of the suffices of words $\tau(n)$. Because ϕ uniformly recurrent, all its finite words appear in the one-way point $\phi'_{\mathbb{N}}$, and thus also in x. Again, $\overline{\mathcal{O}(x)} = X$.

Finally, we show that a cannot occur finitely many times. Namely, suppose $x_j = a, x_\ell = a$, and $x_i \neq a$ for all $i \notin [j,\ell]$. Then a compactness argument shows that there is a point $\phi' \in \overline{\mathcal{O}(\phi)}$ with $|\phi'|_\infty \geq 2$. This is a contradiction, since ϕ was assumed to have unique singularities.

The transitivity of X is clear from the definition. □

Definition 11. *We say a subshift $X \subset S^{\mathbb{Z}}$ is* right-perfect *if the subshift $c(X) = \{y \mid \exists x : x.y \in X\}$ is perfect, that is, if it has no isolated points.*

In other words, X is right-perfect if every word of X has at least two incomparable extensions to the right.

Lemma 5. *Let $Y \in [0,k-1]^{\mathbb{Z}}$ be a nonempty right-perfect subshift with a recursive language. Then, there exists a recursively enumerable infinite prefix code $u_1, u_2, \ldots$ of words in Y such that $(u_j)_{[0,|u_j|-2]} = (u_{j+k})_{[0,|u_j|-2]}$ for all $j, k \geq 1$.*

Of course, symmetrically, there exists such a suffix code if Y is left-perfect.

Proof. Note that a right-perfect nonempty subshift is infinite (even uncountable). Enumerate the words of Y as $V_1 = v_1, v_2, \ldots$, first ordered by length, and then lexicographically among words of each length. Let $u_1 = ua$ be the first one-symbol extension of a word of Y on this list which has at least two one-symbol extensions, ua and ub. Let V_2 be the subsequence of V_1 of words beginning with ub. Having chosen $u_1, \ldots, u_j$ and restricted our list to V_{j+1}, choose again the lexicographically smallest one-symbol extension of a word of Y which has at least two one-symbol extensions in V_{j+1}, and restrict to V_{j+2} accordingly. Since Y is right-perfect, this process continues forever, and the resulting set of words is clearly a prefix code. The condition on compatible prefixes is automatic in the construction. □

Proof of Theorem 3. Suppose $Y \subset [1,k]^{\mathbb{Z}}$. The subshift X will be over the alphabet $[0,k]$.

Enumerate the deterministic Turing machines as $T_0, T_1, T_2, \ldots$. Let $\phi \in \mathbb{N}^{\mathbb{Z}}$ be a computable point in the orbit closure of the ruler sequence.

Using Lemma 5 and the fact that an infinite minimal subshift is left- and right-perfect, take recursively enumerable prefix and suffix codes of words u_i and v_i of Y, respectively. Let $h : \mathbb{N} \to \mathbb{N}$ be a computable infinite-to-one dovetailing of the natural numbers (for example, the ruler sequence). Now, let $x = \tau(\phi)$, where τ is the substitution

$$i \mapsto \begin{array}{ll} 0u_{h(i)}w_iv_{h(i)}, & \text{if } T_{h(i)} \text{ does not halt before step } i. \\ 0u_{h(i)}w_iv_{h(i)+1}, & \text{if } T_{h(i)} \text{ halts before step } i. \end{array}$$

where each w_i is a word of length at least i such that $u_{h(i)}w_iv_{h(i)} \sqsubset Y$, chosen in such a way that $i \mapsto w_i$ is computable, and as i runs over the natural numbers, all words of Y beginning with $u_{h(i)}$ appear as prefixes of words $u_{h(i)}w_i$ infinitely many times (and similarly for $w_iv_{h(i)}$). Let $X = \overline{\mathcal{O}(x)} \subset [0,k]^{\mathbb{Z}}$.

Clearly, X is transitive, and Y is its subsystem. By Lemma 4 it is quasiminimal and all nontrivial proper subsystems are subsystems of Y – thus equal to Y by minimality. The halting problem along a clopen set is undecidable for X because T_j eventually halts if and only if $\mathcal{L}(X) \cap L \neq \emptyset$, where $L = 0u_j[1,k]^*v_{j+1}0$.

To show that the subshift is recursive, note that given any word $0u0$ where $u \in [1,k]^*$, we can easily check whether there exists $n \in \mathbb{N}$ such that $\tau(n) = 0u$. Suppose then that we are given a word $t_00t_10t_20 \cdots 0t_k$, where $t_i \in [1,k]^*$. Using the properties of the sequence ϕ, we can compute two extensions of this word, one beginning with 0 and one ending in 0, such that every extension agrees with one of them. Thus, we may assume the given word is $w = 0t_00t_10t_20 \cdots 0t_k$.

Such a word is in the language of X if and only if $u = \tau^{-1}(0t_00t_10t_20 \cdots 0t_{k-1})$ is well-defined and $u \sqsubset \phi$ holds, and either t_k begins with one of the words $u_{h(i)}$ or it is a prefix of the unique one-way limit $x \in [1,k]^{\mathbb{N}}$ of the words u_i. These conditions are easily seen to be decidable. □

3.4. Countable universal quasiminimal examples

The case of countable quasiminimal subshifts is also interesting. For such subshifts, we show that both the modular halting problem and the counting problem are undecidable, so that the model-checking problems for piecewise testable languages, star-free languages and renewal languages are undecidable as well. In Section 4, we complement these results by showing that in each, the number of subsystems is optimal, and the model-checking problem for local languages (and thus the halting problem along a clopen set) is decidable.

First, we show that if nonzero symbols are asymptotically spaced far apart in x, then x generates a quasiminimal countable subshift.

Lemma 6. *Let $x \in (\{0\} \cup S)^{\mathbb{Z}}$ be such that*

$$\forall n : \exists m : (x_j = x_{j'} \neq 0 \wedge j \neq j' \wedge |j| \geq m \implies |j - j'| > n).$$

Then $X = \overline{\mathcal{O}(x)}$ is a countable weakly transitive quasiminimal subshift whose proper subshifts are among the subshifts of $\mathcal{L}^{-1}(0^*S0^*)$. If x is computable and $m = m(n)$ is computable in n, then X is recursive.

Proof. Every limit point of x is easily seen to be in $\mathcal{L}^{-1}(0^*S0^*)$, so X is contained in $\mathcal{O}(x) \cup \mathcal{L}^{-1}(0^*S0^*)$. Thus, X is countable and quasiminimal, and its proper subshifts are among the subshifts of $\mathcal{L}^{-1}(0^*S0^*)$.

Suppose then that x and m are computable. For each ${}^{\infty}0s0^{\infty} \notin X$, let k_s be the maximal $|i|$ such that $x_i = s$ (given to the algorithm by a look-up table). Given w, if w is not a word in $\mathcal{L}^{-1}(0^*S0^*)$, then w contains two nonzero symbols spaced n apart, so $w \sqsubset X$ if and only if w occurs in $x_{[-m(n)-|w|,m(n)+|w|]}$. If $w \in 0^*$, or $w \in 0^*s0^*$ and ${}^{\infty}0s0^{\infty} \in X$, then $w \sqsubset X$. If $w \in 0^*s0^*$ and ${}^{\infty}0s0^{\infty} \notin X$, then $w \sqsubset X$ if and only if $w \sqsubset x_{[-k_s-|w|,k_s+|w|]}$. □

We begin with the case of modular halting problem. We first show the result for the modular halting problem along a clopen set, as the proof illustrates the main idea, but is easier.

Proposition 5. *There exists a recursive countable weakly transitive quasiminimal subshift $X \subset \{0,1\}^{\mathbb{Z}}$ for which the modular halting problem along a clopen set is Σ^0_1-complete, and which has exactly two subsystems $\mathcal{L}^{-1}(0^*10^*)$ and $\mathcal{L}^{-1}(0^*)$.*

Proof. Enumerate the deterministic Turing machines as $T_0, T_1, T_2, \ldots$. Let $h : \mathbb{N} \to \mathbb{N}$ be the ruler sequence and let p_i denote the ith odd prime number (so $p_1 = 3$). Let τ be the substitution

$$i \mapsto \begin{array}{ll} 10^{2^i}, & \text{if } T_{h(i)} \text{ does not halt before step } i. \\ 10^{p_{h(i)}2^i}, & \text{if } T_{h(i)} \text{ halts before step } i. \end{array}$$

We let $x = {}^{\infty}0.\tau(0123...)$ and $X = \overline{\mathcal{O}(x)}$.

The modular halting problem of this subshift is clearly undecidable, as the distance of two symbols 1 along symbols 0 can be divisible by p_j if and only if T_j eventually halts. The required properties of X follow from Lemma 6. □

Theorem 4. *There exists a recursive countable weakly transitive quasiminimal subshift $X \subset \{0,1\}^{\mathbb{Z}}$ for which the modular halting problem is Σ^0_1-complete, and which has exactly two subsystems $\mathcal{L}^{-1}(0^*10^*)$ and $\mathcal{L}^{-1}(0^*)$.*

Proof. Enumerate the deterministic Turing machines as $T_0, T_1, T_2, \ldots$. Let $h : \mathbb{N} \to \mathbb{N}$ be the ruler sequence. Let $p(i)$ denote the ith odd prime number, and define the *primorial* of $n \geq 2$ to be $n\# = 2p(1)p(2)\cdots p(\ell)$ where ℓ is maximal such that $p(\ell) \leq n$. Thus, $n\#$ is the product of primes up to n. Let $f : \mathbb{N} \to \mathbb{N}$ be an increasing recursive function satisfying $p(f(i-1))^{p(f(i-1))} \leq p(f(i))$ for all i. Let τ be the substitution

$$i \mapsto \begin{array}{ll} 10^{p(f(i))\#-1}, & \text{if } T_{h(i)} \text{ does not halt before step } i. \\ 10^{p(f(i))\#+p(f(i))\#/p(f(h(i)))\cdot k_i-1}, & \text{if } T_{h(i)} \text{ halts before step } i, \end{array}$$

where $0 < k_i < p(f(h(i)))$ is minimal such that

$$p(f(i))\#/p(f(h(i))) \cdot k_i \equiv 1 \bmod p(f(h(i))).$$

Note that such k_i exists because $p(f(i))\#/p(f(h(i)))$ is not divisible by $p(f(h(i)))$.

We let $x = {}^\infty 0.\tau(0123...)$ and $X = \overline{\mathcal{O}(x)}$. To show that the modular halting problem is Σ^0_1-complete, we show that there exist two symbols 1 with distance $\ell \equiv 1 \bmod p(f(j))$ if and only if T_j eventually halts. First, if T_j does halt, then $j = h(i)$ for some i such that T_j halts before step i. Then,

$$10^{p(f(i))\#+p(f(i))\#/p(f(j))\cdot k_i-1}1 \sqsubset x,$$

where $\ell = p(f(i))\# + p(f(i))\#/p(f(j)) \cdot k_i \equiv 1 \bmod p(f(j))$ by the choice of k_i and because $j \leq i$.

Let us show that if T_j never halts, then no such distance ℓ occurs. First, note that if T_j does not halt, then the distance between the ith and $(i+1)$th symbol 1 produced by the construction is divisible by $p(f(j))$ by construction whenever $i \geq j$. Thus, if there is a distance $\ell \equiv 1 \bmod p(f(j))$ between two symbols 1 in x, then it is among the first j symbols, that is, the distance must occur between two symbols 1 in the word $\tau(12\cdots(j-1)) \cdot 1$. The distance is never 1, so it must be at least $p(f(j)) + 1$. However, we have

$$|\tau(1\cdots(j-1))\cdot 1| \leq 2j|\tau(j-1)| = 2jp(f(j-1))\# \leq p(f(j-1))^{p(f(j-1))} \leq p(f(j)),$$

by the assumption on f.

This shows that the modular halting problem is undecidable. The other properties again follow from Lemma 6. □

Of course, the growth rate for the distances between consecutive symbols 1 is not optimal, but we are not aware of essentially simpler substitutions for which the proof is equally short.

We note that just like we could take an arbitrary infinite recursive minimal subshift as the unique minimal subshift in the proof of Theorem 3, one could of course use any infinite subshift of the form $\mathcal{L}^{-1}(u^*vu^*)$ in the place of $\mathcal{L}^{-1}(0^*10^*)$, although small additional complications arise if $|u| \nmid |v|$.

Next, let us consider the counting problem. For this, we need four subsystems.

Theorem 5. *There exists a recursive countable weakly transitive quasiminimal subshift X for which the counting problem is Σ^0_1-complete, and which has exactly four subsystems, $\mathcal{L}^{-1}(0^*(1+2)0^*)$, $\mathcal{L}^{-1}(0^*20^*)$, $\mathcal{L}^{-1}(0^*10^*)$ and $\mathcal{L}^{-1}(0^*)$.*

Proof. Let T_i be an enumeration of Turing machines and h and infinite-to-one computable mapping. Let τ be the substitution

$$i \mapsto \begin{array}{ll} 20^i, & \text{if } T_{h(i)} \text{ does not halt before step } i. \\ 2(0^i1)^{h(i)}0^i, & \text{if } T_{h(i)} \text{ halts before step } i, \end{array}$$

We let $x = {}^\infty 0.\tau(0123...)$ and $X = \overline{\mathcal{O}(x)}$. The counting problem is clearly Σ^0_1-complete, because there exists a point that travels from the cylinder [2] back to itself along [0] visiting [1] exactly j times if and only if T_j eventually halts. The other properties again follow from Lemma 6. □

3.5. Model-checking for context-free languages

In [3], it is shown that the model-checking problem of regular languages is decidable for minimal systems, and in this section, we have shown that this problem is undecidable for more complex subshifts. In another direction, we could ask how far we must step from the class of regular languages to find undecidable model-checking problems for recursive minimal subshifts.

We show that the model-checking problem is hard at least for context-free languages. Our example is a subshift of the *Dyck shift (with labels* $\{1, 2, 3\}$*)*, the subshift of $S = \{[_1, [_2, [_3,]_3,]_2,]_1\}^{\mathbb{Z}}$ where the parentheses are balanced, in the sense that the process of recursively erasing subwords of the forms $[_s]_s$ for $s \in \{1, 2, 3\}$ never introduces a subword of the form $[_s]_{s'}$ with $s \neq s'$.

The language of the Dyck shift is the set of factors of the *Dyck language* L generated by the context-free grammar $A \mapsto AA|[_aA]_a|[_bA]_b|[_cA]_c|\epsilon$. A deterministic push-down automaton M for it is obtained by pushing s on input

$[_s$, and popping the corresponding symbols on closing brackets (rejecting the word if these do not match). For two words u, v over S which occur as subwords of words in L, write $u \sim v$ if the two words correspond to the same element of the syntactic monoid of L. (In other words, they close the same parentheses, and leave the same parentheses open).

Theorem 6. *There exists a recursive minimal subshift X whose model-checking problem for context-free languages is Σ^0_1-complete.*

Proof. For $s \in \{1, 2, 3\}$ let $[^0_i = [_i$, and $]^0_i =]_i$. Let $W_0 = \{[^0_1, [^0_2, [^0_3,]^0_3,]^0_2,]^0_1\}$. Inductively, suppose $W_i = \{[^i_1, [^i_2, [^i_3,]^i_3,]^i_2,]^i_1\}$ is defined, and each word in W_i is a concatenation of words of W_{i-1}, and contains all concatenations uv of words $u, v \in W_{i-1}$ such that uv is a subword of a word of L. We also inductively suppose that each $w \in W_i$ corresponds to one of the symbols $a \in S$ of the original alphabet in the sense that the push/pop action corresponding to a is performed on w when reading the very last symbol, and during the reading of proper prefixes, no existing data on the stack is popped, and all words put on the stack begin with 3. More precisely, we suppose the words in W_i are of the same length n_i, and for all $s \in \{1, 2, 3\}$, $([^i_s)_{[0,n_i-2]} \sim (]^i_s)_{[0,n_i-2]} \sim \epsilon$, $([^i_s)_{n_i-1} = [_s$ and $(]^i_s)_{n_i-1} =]_s$, and if the stack of M initially contains v, then after reading any proper prefix of a word $u \in W^i$, the stack contains either v or $v3w$ for some word w.

We define W_{i+1} as a concatenation of words of W_i, respecting the inductive assumptions. Let $T_1, T_2, T_3, \ldots$ be an enumeration of Turing machines, and let $h : \mathbb{N} \to \mathbb{N}$ be a computable infinite-to-one mapping. Let u_i be any word with $u_i \sim \epsilon$ containing all legal concatenations of pairs of words in W_i. For example,

$$u_i = \prod_{s,s' \in \{1,2,3\}} [^i_s [^i_{s'}]^i_{s'}]^i_s [^i_{s'}]^i_{s'}$$

is such a word, no matter what order the pairs s, s' are listed in.

Now, if $T_{h(i)}$ does not halt in i steps or less, we define

$$[^{i+1}_s = [^i_3 u_i]^i_3 [^i_s, \text{ and }]^{i+1}_s = [^i_3 u_i]^i_3]^i_s.$$

If $T_{h(i)}$ does halt in i steps or less, we define

$$[^{i+1}_s = [^i_3 ([^i_1)^{h(i)} [^i_2]^i_2 (]^i_1)^{h(i)} u_i]^i_3 [^i_s, \text{ and }]^{i+1}_s = [^i_3 ([^i_1)^{h(i)} [^i_2]^i_2 (]^i_1)^{h(i)} u_i]^i_3]^i_s.$$

We let

$$W_{i+1} = \{[^{i+1}_1, [^{i+1}_2, [^{i+1}_3,]^{i+1}_3,]^{i+1}_2,]^{i+1}_1\}.$$

The subshift X is defined as the set of limit points of (say) the words $[^j_1$ as $j \to \infty$.

Now, define $L_k \subset L$ as the language defined by the automaton A_k, which simulates the deterministic push-down automaton A for the Dyck language, but additionally inspects the $k+2$ top symbols of the stack after every step for the word 31^k2. The automaton accepts the word w if and only if $w \in L$, and 31^k2 was seen on the top of the stack during the execution.

We claim that X is minimal, recursive and for j sufficiently large, its language intersects L_j if and only if T_j eventually halts. For minimality, simply note that if $u \sqsubset X$, then $u \sqsubset [^i_1$ for some i (by definition). This word appears in every word of W_{i+1} and every point of X is a concatenation of these words.[18] To show X is recursive, we note that given any word w we can check whether $w \in \mathcal{L}(X)$ by computing the sets W_i until their lengths exceed that of w. Then by the choice of u_i, every word of length $|w|$ that appears in X appears in W_{i+1}.[19]

Now, note that if T_j eventually halts, then certainly L_j intersects the language of X because for any large enough i with $h(i) = j$, $[^i_3([^i_1)^{h(i)}[^i_2 \sqsubset [^{i+1}_a$ and $[^i_3([^i_1)^{h(i)}[^i_2 \sim [_3([_1)^{h(i)}[_2$, which pushes the word 31^j2 on top of the stack of M.

Conversely, suppose j is large. If there exists a subword of X where 31^j2 appears on top of the stack during the run of A on the word, then in particular there exists such a word which is a subword of some $[^{i+1}_s$ or $]^{i+1}_s$. Choose the minimal such i, and suppose that it occurs in $v = [^{i+1}_s$ (the case of a closing bracket being similar). Let k be minimal such that 31^j2 is on the top of the stack after reading $v_{[0,k]}$. If $v_{[0,k]}$ is not a concatenation of words of W_i, then the automaton has read a proper prefix of a word of W_i up to coordinate k. By the inductive assumption, it has then written a word beginning with 3 on the stack, or nothing. In the first case, the word 31^j2 was written on top of the stack while reading that word of W_i, so i is not minimal. In the second case, reading the prefix did not modify the stack, so k is not minimal.

In the remaining case, we have read some concatenation of words of W_i. It is clear by the construction that 31^j2 is written on top of the stack after

[18] We note that for this argument, there is no need to have all legal pairs of symbols in u_i, only all symbols, although the definition of X is more robust if all pairs occur.

[19] For this, on the other hand, having all pairs in u_i is essential.

reading some such prefix if and only if T_j halts or this happened because for some prefix $[^i_3 u$ of $[^i_3 u_i]^i_3 [^i_s$ wrote this on top of the stack. Since u is a concatenation of at most 54 words of W_i, $j > 54$ is enough to prevent this. □

4. Quasiminimality and decidability

In this section, we give various decidability results, which complement the results of the previous section. We take a rather high-level approach to decidability, and usually only describe the logical deductions the algorithm must make to determine the right answer.

4.1. Decidability in the general case

One could say that it is a folklore result that a minimal Π^0_1 subshift has a recursive language; we are not aware of a reference that states and proves this explicitly, although the proof is given in multiple places. The result is shown in [20] in the case of multidimensional SFTs, and the proof works in general. It is also essentially shown in [21], but the connection is not made very explicit. We give a proof below.

Theorem 7. *A minimal Π^0_1 subshift (over $\mathbb{N}$ or $\mathbb{Z}$) is recursive.*

Proof. Given an alphabet S, a word $w \in S^*$ and a Turing machine enumerating the forbidden patterns of a nonempty minimal subshift X over S, we show that it is decidable whether $w \sqsubset X$.

If $w \not\sqsubset X$, then an algorithm can eventually detect this by the assumption that X is Π^0_1. If $w \sqsubset X$, then by uniform recurrence, there exists n such that $w \sqsubset u$ for all $u \in \mathcal{L}_n(X)$. By compactness, there exists k such that the SFT Y defined by the first k forbidden patterns $v_1, v_2, \ldots, v_k$ enumerated by the given Turing machine defining X satisfies $\mathcal{L}_n(Y) = \mathcal{L}_n(X)$. It follows that after enumerating the first k forbidden patterns, the algorithm has found an SFT Y such that $X \subset Y$ and $w \sqsubset u$ for all $u \in \mathcal{L}_n(Y)$, and can deduce that $w \sqsubset X$ (on the assumption that X is nonempty). □

The algorithm is uniform in the given Turing machine and given word, though only when restricted to inputs that define nonempty subshifts. It is in fact easily seen to be undecidable whether a given Turing machine defines an empty subshift or a nonempty minimal subshift: the given Turing machine may output the forbidden patterns of any Π^0_1 minimal subshift as long as another simulated Turing machine does not halt, and forbid every symbol if it does halt.

The following is shown in [3]. The proof is based on analyzing the basins of attraction of clopen sets, and observing they are clopen. We believe the proof easily adapts to $\mathbb{Z}$-subshifts by considering basins of attraction for σ and σ^{-1} separately, but this is beyond the scope of the article.

Theorem 8. *A minimal Π_1^0 subshift $X \subset S^{\mathbb{N}}$ is decidable.*

The model-checking problem of regular languages reduces to the model-checking problem for ω-languages, so we have the following corollary.

Corollary 1. *If X is minimal Π_1^0 subshift over $\mathbb{N}$ or $\mathbb{Z}$, then its model-checking problem for regular languages is decidable.*

In the rest of this subsection, we restrict to $\mathbb{Z}$-subshifts.

We show that (a non-uniform version of) Theorem 7 holds for the class of quasiminimal subshifts as well, in Theorem 9 below. As a quasiminimal subshift contains only finitely many subshifts, it is natural to approach such subshifts by induction on the number of proper subshifts they have, and we give the following notation for this.

Definition 12. *For X quasiminimal, define*

$$\mathrm{Q}(X) = |\{Y \mid Y \subset X, Y \text{ is a subshift}\}|.$$

The empty subshift and the subshift itself are also counted: if X is nonempty and minimal, then $\mathrm{Q}(X) = 2$, and $\mathrm{Q}(\emptyset) = 1$. We define Q like this, instead of counting only the interesting (proper) subsystems, as it has a natural interpretation as the cardinality of the subsystem lattice.

We need the following basic structural results.

Definition 13. *If $w \sqsubset X$ and $u \leq_X w$ for all $u \sqsubset X$, then w is called a* generator *(for X).*

Lemma 7. *Let X be quasiminimal. Then X has a generator if and only if it is not the union of its proper subshifts.*

Proof. First, we suppose that X is the union of its proper subshifts. Then, whenever $w \sqsubset X$, there must exist a proper subshift $Y \subsetneq X$ with $w \sqsubset X$.

If $Y \subsetneq X$, then $u \not\sqsubset Y$ for some $u \sqsubset X$, which implies $u \not\leq_X w$. Thus, no $w \sqsubset X$ is a generator for X.

Now, suppose X is not the union of its proper subshifts. Let $Y \subsetneq X$ be the union of its proper subshifts. Note that since X has only finitely many proper subshifts, $Y = Y_1 \cup \cdots \cup Y_k$ for some k and subshifts $Y_i \subsetneq X$, and thus Y is itself a proper subshift of X. Let w be any word that occurs in X but not Y. Then there exists no proper subshift Z of X containing w. It follows that, for any $u \sqsubset X$, the subshift of X where u is forbidden does not contain w, which means u occurs in every point of X containing w. It follows that $u \leq_X w$. □

Note that if w is a generator for X, then since $u \leq_X w$ for any $u \sqsubset X$, any point x containing w must contain every word $u \sqsubset X$, that is, every point $x \in X$ containing w is weakly transitive (in particular, X is weakly transitive).

Lemma 8. *If X is quasiminimal and Y is a subshift of X, then there exists a finite list of words $v_1, v_2, \ldots, v_k$ such that Y is obtained from X by forbidding the words v_i. In particular, every subshift of a Π^0_1 quasiminimal subshift is Π^0_1.*

Proof. Let Y_i be the ith SFT approximation of Y, and consider the subshifts $Z_i = Y_i \cap X$. We have $\bigcap Y_i = Y$, and thus also $\bigcap_i Z_i = Y$. The Z_i are a decreasing sequence of subshifts of X, and thus for some j we have $Z_j = Z_{j+k}$ for all $k \geq 0$. It follows that $Y = Z_j$. Since Y_i is an SFT, $Y_i \cap X$ is obtained from X by forbidding a finite set of words. The latter claim is then trivial: the Turing machine outputting the forbidden words of a subshift $Y \subset X$ first outputs the finitely many words v_i, and then the forbidden words of X. □

Theorem 9. *A quasiminimal Π^0_1 subshift $X \subset S^{\mathbb{Z}}$ is recursive.*

Proof. We proceed by induction on $\mathrm{Q}(X)$. First, if $\mathrm{Q}(X) = 1$, then $X = \emptyset$ and there is nothing to prove. Suppose then that $\mathrm{Q}(X) > 1$, and let $u \in S^*$. We have to decide whether $u \sqsubset X$. By the induction hypothesis and the previous lemma, for any proper subshift $Y \subsetneq X$ it is decidable whether $u \sqsubset Y$. If $u \sqsubset Y$ for some proper subshift, then the algorithm may conclude $u \sqsubset X$. Also, if $u \not\sqsubset X$, we can semidecide this by enumerating the forbidden words of X by the assumption that X is Π^0_1. The case that is left is that $u \sqsubset X$ and $u \not\sqsubset Y$ for any proper subshift $Y \subset X$.

In such a case, the union Y of all proper subshifts of X does not equal X. By Lemma 7, there exists a generator $w \sqsubset X$ such that $w \sqsubset x \in X$ implies that x is weakly transitive in X. We may assume this w is known to the algorithm, by way of a look-up table. Thus, by compactness, there exists $n \in \mathbb{N}$ such that $x_{[0,|w|-1]} = w$ implies $x_{[i,i+|u|-1]} = u$ for some $i \in [-n, n]$, for all $x \in X$. This must also hold in a suitable SFT approximation of X, obtained by running the Turing machine defining X for sufficiently many steps. Once such n and an SFT approximation Z is found, the algorithm can conclude $u \sqsubset X$. □

Remark 2. *An important point is that the generator w was needed in the algorithm, and one cannot compute it directly from a Turing machine defining X (as one can verify by an easy counterexample). Indeed, one can hardly compute anything from such a representation, because it cannot even be decided if the Turing machine defines an empty subshift. Thus, the result is not uniform in X. However, like in the case of Theorem 7 (where we supplied the information of whether the subshift is empty), the algorithm is at least somewhat uniform, as we only need the lists of forbidden words (the lists $(v_1, \ldots, v_k)$ given by Lemma 8) defining the proper subshifts of X and the lists of generators for all subshifts of X. An easy example shows that in the case of general quasiminimal subshifts, knowing whether the subshift is empty is* not *enough to determine whether a given word u is in the subshift. However, it is an interesting question exactly what "shape information" is needed.*

A trivial corollary of the previous result and Lemma 8 is that recursive quasiminimal subshifts are *hereditarily recursive*, that is, all their subshifts are recursive.

Corollary 2. *All subshifts of a recursive quasiminimal subshift are recursive.*

Note that in the previous section, we essentially defined our Π^0_1 subshifts by giving a computational process that outputs a point, and we then showed that one has an algorithm for recognizing whether a given word occurs in the point. Just the fact that the point was given by a computational process does not automatically imply that there is a such an algorithm, but it does automatically follow that the subshift has at least a recursively enumerable language. Thus, we could not have hoped to find a Π^0_1 quasiminimal subshift which is not recursive with this technique, and indeed, by Theorem 9, there is no such subshift. We can strengthen this observation slightly, and prove Proposition 6, which states that all quasiminimal Π^0_1 subshifts are obtained as orbit closures of computable points. We first need a few lemmas.

Lemma 9. *Let X be a subshift whose language is recursively enumerable. Then the computable points are dense in X.*

Proof. Given any word $u \sqsubset X$, we construct a Turing machine that enumerates a sequence of words $u_1, u_2, u_3, \dots \sqsubset X$ such that $u_1 = u$ and for each i, there exists a and b such that $u_{i+1} = au_ib$. The Turing machine starts by setting $u_1 = u$, and constructs the sequence by enumerating words of X, and extends the last word of the sequence chosen so far, u_i, to $u_{i+1} = au_ib$, whenever a word $v \sqsubset X$ properly containing au_ib is enumerated. From such a Turing machine, one easily obtains a computable point in X with central pattern u. □

Lemma 10. *If X is a Π^0_1 quasiminimal subshift, then it is a finite union of orbit closures of computable points.*

Proof. Suppose that the claim fails for some Π^0_1 quasiminimal subshift X. Enumerate the words of X as $u_1, u_2, \dots$. Let x_1 be a arbitrary computable point in X. Inductively, let $u_{k(j)}$ be the first word on the list $u_1, u_2, \dots$ such that $u_{k(j)} \not\sqsubset Y_{j-1} = \overline{\mathcal{O}(x_1)} \cup \overline{\mathcal{O}(x_2)} \cup \dots \cup \overline{\mathcal{O}(x_{j-1})}$. Such $u_{k(j)}$ can always be found, since otherwise $X = Y_{j-1}$, a contradiction. Let then x_j be a computable point containing $u_{k(j)}$. It follows that $Y_1 \subsetneq Y_2 \subsetneq \cdots$ is an infinite increasing sequence of subshifts of X, which contradicts quasiminimality. □

The following proposition is subtly stronger.

Proposition 6. *If X is a Π^0_1 quasiminimal subshift, then there exists a finite set $\{x_1, x_2, \dots, x_k\}$ of computable points with distinct languages such that for every $x \in X$ we have $\mathcal{L}(x_i) = \mathcal{L}(x)$ for some $i \in [1, k]$.*

Proof. A Π^0_1 quasiminimal subshift has finitely many subshifts, which are all Π^0_1 and quasiminimal. We apply the previous lemma to each of them, and take the union of the finitely many sets of points obtained. To obtain distinct languages, repeatedly remove x_j from the set if $\mathcal{L}(x_i) \neq \mathcal{L}(x_j)$ for some $i \neq j$. Now, suppose that there is a point $x \in X$ such that $\mathcal{L}(x_i) \neq \mathcal{L}(x)$ for all $i \in [1, k]$. Then, for each i, either x contains a word u_i such that $u_i \not\sqsubset x_i$, or x_i contains a word v_i such that $v_i \not\sqsubset x$. Define a subshift Y of X by forbidding the v_i, and choose a subword w of x containing all the u_i. Then, x is a point of Y containing a word w which is not in any of the x_i. This is a contradiction, since we applied the previous lemma also to Y. □

This proposition suggests a small tangent to the discussion. The following proposition shows that the previous result is not a characterization of Π^0_1 quasiminimal subshifts, and also shows that there is no symmetric version of Theorem 7 for "Σ^0_1 subshifts". The result is presumably well-known, but we are not aware of a reference for it.

Proposition 7. *There exists a computable point $x \in \{0,1\}^{\mathbb{Z}}$ such that $\overline{\mathcal{O}(x)}$ is minimal and not Π^0_1.*

Proof. We only need to describe a minimal subshift whose subwords are recursively enumerable, as such a subshift contains a computable point, and every point in a minimal subshift generates it.

Let $W_0 = \{0,1\}$. For each i, we will define W_i to be a set of two words of the same length such that every (bi-)infinite concatenation of words of W_i has a unique partition into words of W_i, and both words of W_i are contained in both words of W_{i+1}. For any list of sets with these properties, we obtain a minimal subshift by taking the limit points of the words W_i in the obvious way. We denote by ℓ_i the maximal number such that for some $w \in W_i$, the word $w^{\ell_i} \in X$ occurs in some concatenation of words of W_{i+1}. We make sure ℓ_i is always finite, and strictly smaller than the length of the words $w \in W_{i+1}$. We will not define these words by a direct induction, but by a computational process, and the process may change the definition of W_i arbitrarily late in the process of constructing these sets – the reader should think of the W_i as variables in the programming sense of the word in the following.

The property of unique partitions is clearly true for $W_0 = \{0,1\}$. If $W_i = \{w_1, w_2\}$ (where $w_1 < w_2$ in lexicographical order) satisfies this property, and u, referred to as the *cover-up*, is any concatenation of words in W_i, then we set $W_{i+1} = \{w_1^i w_2^{q-i} w_1 u \mid i \in \{1,2\}\}$ where $q = 2|u| + 3$. These words will have the property of unique partitions, contain both words of W_i, and certainly $\ell_i < |w_1 w_2^{q-1} w_1 u|$, as required. Note also that $\ell_i > |u|$, and if u is the empty word ϵ, then $\ell_i = 3$.

Now, we describe the algorithm that constructs the sets W_{i+1}, if W_i is the last set of words computed so far. The algorithm sets $W_{i+1} = \{u_1, u_2\}$, where u_1 and u_2 are any two words constructed as above from the words of W_i and the cover-up $u = \epsilon$. Every time it computes a new set W_i, it outputs the words of W_i, and all their subwords, in lexicographical order, as words occurring in X. While constructing these words, the algorithm dovetails the computation of all Turing machines $T_1, T_2, \ldots$. If the kth machine halts and the machine has, up to that point, computed the sets W_i for $i \leq K$ (where we may assume $K > k$), then the algorithm changes the definition of W_k: If $W_{k-1} = \{w_1, w_2\}$ and $W_K = \{u_1, u_2\}$ (and $w_1 < w_2$ and $u_1 < u_2$ in lexicographical order), let $u = u_1u_2$, and define W_k as in the previous paragraph with cover-up u, so that every word of the new set W_k contains the word u. Furthermore, since we previously had $\ell_{k-1} \leq |w|$ for $w \in W_k$, the new value of ℓ_{k-1} is strictly larger, as $\ell_{k-1} > |u|$, $u = u_1u_2$ and $u_i \in W_K$.

The crucial point is that since the algorithm changed its mind about the contents of W_k, all the subwords of words in W_i for $i \geq k$ it has output so far have been 'wrong', in the sense that they were being output as subwords of W_i for the wrong i. Of course, the output does not actually contain the index i, so the output is still correct – these words are now words of W_k, and will still be legal words of the subshift X we obtain in the end. This is because we covered up the lie with a suitable choice of u.

It is clear that the sets W_j all converge, as once those Turing machines T_i with $i \leq j$ have halted which will eventually halt, the contents of W_j will never change. The sets obtained in the limit still satisfy the properties we asked for, and thus the limit of these sets of words is a minimal subshift by a standard argument. The subshift has a recursively enumerable language by construction.

Finally, let us show the language of the subshift is not Π^0_1. For this, simply observe that if we had a decision algorithm for the language of X, we could decide, for each i one by one, whether the machine T_i ever halts. Namely, assuming we know whether the machine T_i halts for all $i < j$, we can determine whether T_j eventually halts as follows: we run the algorithm described above until it has computed the sets W_i for all $i < j$ (and recomputed them as many times as necessary, whenever the machines T_i with $i < j$ halt). Let $W_{j-1} = \{w_1, w_2\}$, in lexicographical order. Now, the machine T_j eventually halts if and only if w_2^4 occurs in the language of X (due to the choice of the ℓ_i). □

We note that it does not follow from the previous proof that the language of the subshift constructed is Σ^0_1-complete for many-one reductions, only Turing reductions.

We can strengthen Theorem 9 further, and show that even the halting problem of a quasiminimal subshift is decidable (compare this with Proposition 3). More precisely, we show that the model-checking problem for elementary piecewise testable languages is decidable, and the halting problem follows as a special case. First, we state the obvious.

Lemma 11. *The generating poset and subpattern poset of a quasiminimal subshift are finite.*

Proof. Let X be quasiminimal. If $x \in X$, then $Y = \overline{\mathcal{O}(x)}$ is a subshift of X with $\mathcal{L}(Y) = \mathcal{L}(x)$, and X has finitely many subshifts, so the subpattern poset is finite. Let $\{x_1, x_2, \ldots, x_k\} \subset X$ be a finite set of representatives such that $\forall x \in X : \exists i : \mathcal{L}(x) = \mathcal{L}(x_i)$.

The inequality $u \leq_X v$ means that whenever $v \sqsubset x$ for $x \in X$, then also $u \sqsubset x$. Associate to each $u \in \mathcal{L}(X)$ the tuple $B(u) = (b_1, b_2, \ldots, b_k) \in \{0,1\}^k$ where $b_i = 1 \iff u \sqsubset x_i$. If $B(u) = B(v)$, then $u \leq_X v \leq_X u$ holds, so the generating poset of X is finite. □

Theorem 10. *If $X \subset S^{\mathbb{Z}}$ is a quasiminimal Π^0_1 subshift, then the model-checking problem of X for elementary piecewise testable languages is decidable.*

Proof. Restated in terms of clopen sets, we are given $[C_1], [C_2], \ldots, [C_\ell]$, and we need to decide whether all of these are visited in order by a single point. We can deal with each of the finitely many tuples $T = (t_1, \ldots, t_\ell) \subset (S^*)^\ell$ with $t_i \in C_i$ separately, and check for such a tuple whether there exist $x \in X$ and $i_1 < \ldots < i_\ell \in \mathbb{Z}$ such that $x_{[i_j, i_j+|t_j|-1]} = t_j$ for all $1 \leq j \leq \ell$. We denote this condition by $T \sqsubset X$.

Let $w_1, w_2, \ldots, w_k$ be a finite set of representatives for the elements of the generating poset. To each word u, we associate $h_u \in \mathbb{N}$ such that in every extension of u of length h_u, the unique w_i with $w_i \sim u$ occurs, given by Lemma 2.

We may suppose these words, and their mutual ordering, are known to the algorithm. Now, let $T = (t_1, \ldots, t_\ell)$ be given. We may again assume $\mathrm{Q}(X) > 1$, and prove the decidability of $T \sqsubset X$ by induction on $\mathrm{Q}(X)$. We do a similar case analysis as in Theorem 9. By the induction hypothesis, for any proper subshift $Y \subsetneq X$ it is decidable whether $T \sqsubset Y$, so we suppose $T \not\sqsubset Y$ for all proper subshifts Y of X. If X is the union of its proper subshifts, the algorithm can decide $T \sqsubset X$ by checking whether $T \sqsubset Y$ for all $Y \subsetneq X$, so we assume this is not the case. As above, denote by $Y \subsetneq X$

the union of all proper subshifts of X and let $w \in S^*$ be a generator for X (known to the algorithm). Now, choose a point $x \in X$ which contains w, so that x is weakly transitive in X. We may assume all of the words t_i appear in x, since if $t_i \not\sqsubset x$, then $t_i \not\sqsubset X$, which the algorithm can prove by enumerating forbidden patterns of X.

If w appears infinitely many times in x (which we may assume is known to the algorithm), the algorithm only needs to prove $t_i \leq_X w$ for all i (which is possible, since $\leq_X$ is semidecidable, and w is a generator): then the words t_i in fact appear in x in every possible order, and thus $T \sqsubset X$. Suppose then that w only appears once. Then x is an isolated point, so that the only points with the same subwords as x are in the orbit of x. Furthermore, w is an isolating pattern. Let Y_L and Y_R be the subshifts generated by the left and right tail of x.

Now, let $u = t_i$ for some i, let $u \sim w_j$, and let $h = h_u$. Now, if the unique w_j with $w_j \sim u$ occurs in Y_L, then u appears infinitely many times in the left tail of x. Otherwise, we can compute a bound on how far to the left of w the word u can appear: w_j occurs at most m steps to the left of w for some m (which we may assume is known to the algorithm by way of a look-up table), and then u cannot appear $m + h_u$ steps to the left of the origin. Similarly, we can semidecide that u appears infinitely many times to the right, or find a bound on how far to the right of w it can appear. Since the language of X is recursive, we can now easily decide whether $T \sqsubset X$. □

The halting problem is the subcase where there are only two clopen sets.

Corollary 3. *The halting problem of a quasiminimal Π^0_1 subshift is decidable.*

We give a simpler direct proof of this as well, using more external information. The simpler proof works for any fixed tuple length (with larger and larger look-up tables), but not in general.

Direct proof of Corollary 3. Let $w_1, \ldots, w_k$ be representatives of the generating poset, and let $H(i,j)$ be the answer of the halting problem for the cylinders $[w_i], [w_j]$. Given any two words u, v, we find w_i and w_j such that $u \sim w_i$ and $v \sim w_j$. Using the fact that X is recursive, we can check whether some $x \in X$ moves from $[u]$ to $[v]$ in at most

$$|w_i| + h_{u,w_i} + h_{w_i,u} + |w_j| + h_{v,w_i} + h_{w_i,v}$$

steps (and if yes, answer "yes"). If not, the answer to the halting problem is $H(i,j)$. □

Remark 3. *By Theorem 10, we can decide $T \sqsubset X$ for any tuple T, and thus we can certainly decide $P(P_1(X), P_2(X), \ldots, P_k(X))$ where P is an arbitrary Boolean formula and the $P_i(X)$ are statements of the form $T_i \sqsubset X$ for tuples of words T_i. In contrast, we saw in Theorem 5 that for piecewise testable languages, the Boolean closure of elementary piecewise testable languages, the model-checking problem is undecidable. This may seem contradictory, but it only means that*

$$P((\exists w \sqsubset X : P_1(w)), (\exists w \sqsubset X : P_2(w)), \ldots, (\exists w \sqsubset X : P_k(w)))$$

is decidable for Π^0_1 quasiminimal subshifts, while

$$\exists w \sqsubset X : P(P_1(w), P_2(w), \ldots, P_k(w))$$

is undecidable, where we write $P_i(w) \iff T_i \sqsubset w$.

4.2. Quasiminimal subshifts with $\mathbb{N}$-actions

Most of our study has been about quasiminimal subshifts $X \subset S^{\mathbb{Z}}$ with the $\mathbb{Z}$-action given by the shift map. We now show that quasiminimal $\mathbb{N}$-subshifts are in fact decidable (when subsystems are not required to be surjective). See Figure 2 for an illustration.

Theorem 11. *Let $X \subset S^{\mathbb{N}}$ be a subshift. Then X is quasiminimal if and only if there exist finitely many minimal subshifts $Y_1, Y_2, \ldots, Y_k$, finitely many points $x_1, \ldots, x_\ell$ and m such that $X = \bigcup_i Y_i \cup \{x_1, \ldots, x_\ell\}$ such that for all $i \in [1, \ell]$, either $\sigma(x_i) \in \bigcup_i Y_i$ or $\sigma(x_i) \in \{x_{i+1}, \ldots, x_\ell\}$.*

Proof. The "if" direction is easy to verify. Suppose then that X is quasiminimal. We show that X is the of required form.

Since each $\sigma^i(X)$ is a subsystem, we must have $\sigma^{m+1}(X) = \sigma^m(X)$ for some m, and we write $Y = \sigma^m(X)$. We claim that there are only finitely many points outside Y. Namely, if $x \in X$, then $\sigma^m(x) \in Y$. If ℓ is the least number such that $\sigma^\ell(x) \in Y$, then

$$\overline{\mathcal{O}(x)} = \{x, \sigma(x), \ldots, \sigma^{\ell-1}(x)\} \cup Y'$$

is a subshift of X for some subshift $Y' \subset Y$ (it is closed as a union of $\ell + 1$ closed sets, and shift-invariant by definition). Thus, each point $x \in X \setminus Y$ is contained in a subshift of X which contains only finitely many points of $X \setminus Y$. If $X \setminus Y$ were infinite, this would imply that X is not quasiminimal.

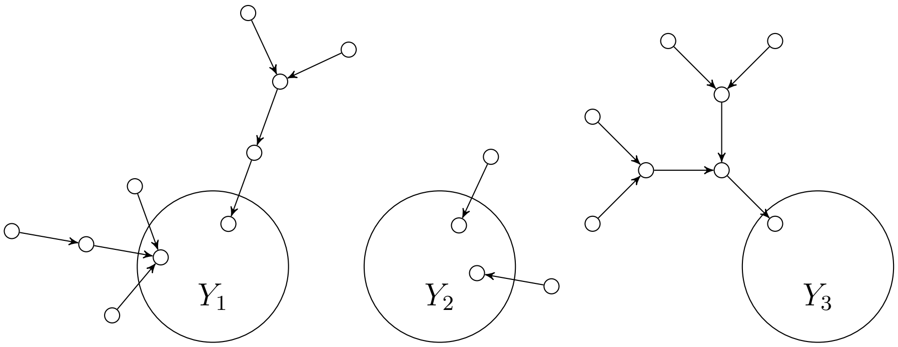

Figure 2: A typical quasiminimal $\mathbb{N}$-subshift. The Y_i are the minimal subshifts and the small circles are the points x_i and their images in $\bigcup_i Y_i$.

For $x, y \in X \setminus Y$, write $x \prec y$ if $\sigma^i(x) = y$ for some $i \in \mathbb{N}$. Since $\sigma^m(x) \in Y$, we cannot have $x \prec y \prec x$ unless $x = y$, so $\prec$ is a partial order on $X \setminus Y$. We obtain the points $x_1, \dots, x_\ell$ by extending it to a total order.

Let $Y_1, Y_2, \dots, Y_k$ be the minimal subsystems of X. The list is finite by quasiminimality. We now show that every point $x \in Y$ is uniformly recurrent, thus in $\bigcup_i Y_i$. Suppose this is not the case. Then there exists a word $w \sqsubset x$ which either occurs finitely many times in x, or occurs infinitely many times but with arbitrarily long gaps. In the first case, $y = \sigma^i(x)$ contains only one copy of w at $y_{[0,|w|-1]} = w$, and in the second, we find a limit point $y \in \overline{\mathcal{O}(x)}$ with this property. Since $y \in Y$, then by a compactness argument it has an infinite chain of preimages

$$\cdots \overset{\sigma}{\mapsto} y_4 \overset{\sigma}{\mapsto} y_3 \overset{\sigma}{\mapsto} y_2 \overset{\sigma}{\mapsto} y_1 = y$$

with $y_i \in Y$ for all i. Then $X_i = \overline{\mathcal{O}(y_i)}$ is a subshift of Y containing exactly i points where w occurs, and thus the X_i are an infinite family of subshifts of X, contradicting quasiminimality. □

We see that the case of $\mathbb{N}$-actions looks very different than that of $\mathbb{Z}$-actions.

Corollary 4. *Let $X \subset S^{\mathbb{N}}$ be recursive and quasiminimal. Then X is decidable.*

Proof. Lemma 8 of [3] states that a system whose limit set is a union of finitely many minimal systems is decidable. By the above theorem, it is clear that quasiminimal $\mathbb{N}$-subshifts have this property. □

4.3. Decidability in the countable case

We first show that, for pretty much trivial reasons, having only a single minimal subshift implies decidability in the countable case (even without assuming, a priori, that the subshift is Π^0_1). In particular, the subshift in Theorem 4 has the optimal number of subsystems.

Proposition 8. *Let $X \subset S^{\mathbb{Z}}$ be a countable subshift with $\mathrm{Q}(X) \leq 3$. Then $X = \overline{\mathcal{O}({}^{\infty}uvu^{\infty})}$ for some words u, v, and thus decidable. Its model-checking problem for context-free languages is also decidable.*

Proof. The case $\mathrm{Q}(X) < 3$ is trivial, so suppose $\mathrm{Q}(X) = 3$. The CB-derivative X' of X is a proper subshift of X. We must have $\mathrm{Q}(X') < \mathrm{Q}(X)$, and thus X' is a countable minimal subshift. This implies $X^{(2)} = \emptyset$, so the CB-rank of X is 2. Every countable subshift with CB-rank 2 is generated by a single point which is eventually periodic in both directions [12, 18], and since X has only one minimal subshift X', the repeating patterns in both tails must be equal. Both decidability of such a subshift, and the decidability of the model-checking problem of context-free languages then follow by observing that such X is sofic and its language is thus regular. In particular we note that the intersection of a context-free language with a regular language is context-free, and the emptiness problem is decidable for context-free languages. □

By Theorem 5, there exists a recursive countable quasiminimal subshift X for which the counting problem is Σ^0_1-complete. We needed 4 proper nontrivial subsystems for this, that is, $\mathrm{Q}(X) = 6$. We show this is optimal. We begin with the following recoding argument.

Definition 14. *Suppose X is countable and quasiminimal. We say X is in* normal form *if the following holds. There exists a finite set of words $w_1, \ldots, w_n$, $w_{n+1}, \ldots, w_{n+m}$ such that*

- *the words w_i are over disjoint alphabets and $\mathcal{L}_1(w_i) = |w_i|$ for all w_i,*
- *every point $x \in X$ has a representation as a bi-infinite concatenation of the words w_i (which is automatically unique),*
- *for $j \in [1, n]$, the point $x_j = w_j^{\mathbb{Z}}$ is in X,*

- *for each $j \in [n+1, n+m]$, there is (up to shifting) a unique point $x_j \in X$ with $w_j \sqsubset x_j$ but $w_k \not\sqsubset x_j$ for $k > j$, and w_j occurs only once in x_j, and*

- *every point of X is in the orbit of one of the points x_j.*

Furthermore, each of the points x_j with $j \in [n+1, n+m]$ is of the following form: $x_j = x_j^L.w_j x_j^R$, where either $x_j^R = u_i^{\mathbb{N}}$ for some $i \in [1,k]$, or we have

$$x_j^R = (w_{h_1})^{\ell_1}\, w_{k_1}\, (w_{h_2})^{\ell_2}\, w_{k_2}\, (w_{h_3})^{\ell_3}\, w_{k_3}\, (w_{h_4})^{\ell_4} \ldots,$$

where for all i, $h_i \in [1,n]$, $k_i \in [n+1, n+j-1]$ and $\ell_i > 1$ and symmetric conditions hold for x_j^L.

It is easy to see that a subshift X in normal form with fixed n, m contains precisely n minimal subsystems.

Lemma 12. *Let Y be countable and quasiminimal. Then Y is conjugate to a subshift in normal form.*

Proof. We construct the conjugacy inductively on $\mathrm{Q}(Y)$, so that if $Z \subset Y$ and $Z \subset Y'$, then the conjugacies from Y and Y' to their recoded versions X and X' have the same restriction to Z. This is guaranteed if we always directly modify the conjugacy obtained for subsystems when building the conjugacy for the full system in the induction step.

If Y is countable and minimal, then it is clearly already of the form ${}^{\infty}w^{\infty}$ as required ($w_1 = w$, $n = 1$ and $m = 0$). If Y is the union of its proper subsystems, then we apply the inductive assumption to the subsystems, and observe that the conjugacy extends to the full system because the conjugacies agree in the intersections.[20]

Suppose then that Y is not minimal, and not a union of proper subsystems. Then it contains a generator w, which we may assume is an isolating pattern, because isolated points are dense in a countable subshift. Let y be the unique point containing w. Then w occurs only finitely many times in y, since Y is not minimal. Let ϕ be the conjugacy obtained for the union Z of proper subsystems of Y. Suppose ϕ has radius r, and suppose t is the maximum length of the words w_i obtained for Z. There exists $c > 2r + t$

[20] More precisely, this follows from the Pasting Lemma.

such that $y_{(-\infty,-c]}$ and $y_{[c,\infty)}$ both extend to points of the $(2r+t)$th order SFT approximation of Z.

This lets us apply ϕ in the tails of y: we define ψ by $\psi(x)_i = \phi(x)_i$ if $w \not\sqsubset x_{[i-c-|w|,i+c+|w|]}$, and $\psi(x)_i = a_j$ if $x_{[i+j,i+j+|w|-1]} = w$ for $j \in [-c-|w|, c+1]$, where the a_j are new symbols not used by ϕ. Then ψ is of the desired form, up to possibly extending the words surrounding the new word over the a_j to full ones (there are unique such extensions since the words w_j are over disjoint alphabets and $|\mathcal{L}_1(w_j)| = |w_j|$). We increase m by 1, let w_{n+m} be the new word over the symbols a_i, and $x_{n+m} = \psi(y)$. □

The conditions on the forms of points in Lemma 12 of course do not automatically guarantee that the subshift is quasiminimal – for this, one needs strong additional properties for the sequences h_i, ℓ_i, k_i in the left and right tails of the points x_j. The precise conditions for this are somewhat complicated. We note that in particular typically the gap sequence ℓ_i does not tend to infinity. This is because the tail of a point x_j may occasionally get close to another point x_k with $n < k < j$ (which happens if $\overline{\mathcal{O}(x_k)} \subset \overline{\mathcal{O}(x_j)}$). In such a case, one will see some short gaps ℓ_i, since there are typically short gaps in the representation of x_k. Conversely, when we see short gaps ℓ_i (far enough away from the central pattern u_j), this must mean some word u_k must be nearby, where $n < k < j$.

Theorem 12. *Suppose X is countable and $\mathrm{Q}(X) \leq 5$. Then the model-checking problem of star-free languages is decidable for X.*

Proof. We perform a small case analysis to reduce to the interesting case that the lattice of subshifts of X is isomorphic to $(\{0,1,2,3,4\}, <)$. As usual, in a model-checking problem we may assume X is not the union of its proper subshifts. Thus, there exists a generator $w \sqsubset x$, which we may take to be isolating. By forbidding w from X, we obtain a proper subshift Y with $\mathrm{Q}(Y) \leq 4$.

First, suppose Y is the union of proper subshifts, then we must have $Y = \mathcal{L}^{-1}(a^* + b^*)$ for some (possibly equal) words a, b. Namely, if Y contains at least two minimal subsystems $\mathcal{L}^{-1}(a^*)$ and $\mathcal{L}^{-1}(b^*)$, it contains their union (because $Y \subset \overline{\mathcal{O}(x)}$), which already gives 4 subsystems. It is impossible for Y to contain only one minimal subsystem $\{{}^\infty a^\infty\}$, as then Y would contain at least two distinct points left- and right-asymptotic to ${}^\infty a^\infty$, and thus their union, which gives 5 subsystems. If $Y = \mathcal{L}^{-1}(a^* + b^*)$, it is easy to see that

x is left asymptotic to ${}^\infty a^\infty$ and right asymptotic to ${}^\infty b^\infty$ or vice versa, and it follows that X is decidable.

Suppose then that Y is not a union of proper subshifts, and thus $Y = \overline{\mathcal{O}(y)}$. Since the maximal proper subshift Z of Y satisfies $\mathrm{Q}(Z) \leq 3$, we have $Z = \mathcal{L}^{-1}(u^*vu^*)$ for some words u, v. Thus, the subpattern poset of Z is isomorphic to $(\{0,1,2\}, <)$, that of Y is isomorphic to $(\{0,1,2,3\}, <)$, and that of X is isomorphic to $(\{0,1,2,3,4\}, <)$. In particular, X has only one minimal subsystem, so we may assume X is in normal form with $n = 1, m = 3$, since the hardness of model-checking problems is preserved under conjugacy.

Let L be a given star-free language. We may suppose $L = S^*LS^*$, as this does not change the answer to the model-checking problem, and S^*LS^* is star-free whenever L is. Using the algebraic characterization of star-free languages, let p be such that for all words u, $u^{p-1} \sim u^p$ in the syntactic monoid of L. We show that we can compute the central pattern and repeating patterns for an eventually periodic point x' such that some $u \in L$ occurs in x' if and only if some $v \in L$ occurs in x_n, which gives the answer since model-checking problems are easy to solve for eventually periodic points, and x_n is a weakly transitive point for X.

The important general observations here are the following: First, if $x = y_L u^{p+1} y_R$ for any word u and any tails y_L and y_R, then due to the assumption $L = S^*LS^*$ and the choice of p, the point $x' = y_L u^p y_R$ satisfies the model-checking problem for L if and only if x does. Furthermore, if we start with a point x and repeatedly contract a occurrences of $(p+1)$th powers u^{p+1} to u^p, then as long as all such contractions happen strictly to the right or strictly to the left of the origin, the infinite tails are always shifted towards the origin in such contractions, then we obtain a well-defined limit $y \in S^{\mathbb{Z}}$ (every cell is at a finite distance from the origin, and thus only finitely many contractions can change its value), and the language of y will satisfy the model-checking problem for L if and only if the original point does.

We explain how to perform the contraction process to the right tail of x_4 (in the notation of Definition 14), so that we obtain an eventually periodic point in the limit. The left tail is handled symmetrically, to obtain the eventually periodic point for which we then solve the model-checking problem of L. Let j be maximal such that w_j occurs in x_4 infinitely many times to the right. Our contraction process contracts the right tail of x_4 to $w_4(uw_jv)^\infty$ for suitable words u, v.

If $j = 1$, then x_4 is already eventually periodic to the right, and we are done. Suppose then that $j > 1$. Since w_4 is an isolating pattern, we have $j < 4$. We may assume no w_r with $r > j$ appears to the right of the central pattern w_4 of x_4 by modifying a finite part of the conjugacy (or otherwise restricting our attention to a suitable tail).

Consider the case $j = 2$. Then, letting h_i, ℓ_i, k_i be as in Definition 14 for the point x_4 (whence $h_i = 1$ for all i), in fact $k_i = 2$ for all i and $\ell_i \to \infty$ as $i \to \infty$. In this case, we can compute s such that $w_2 w_1^{p-i'} w_2$ for $i' > 0$ cannot occur in the right tail of x_4 beyond the coordinate s. Such s exists, because otherwise there is a right limit point of x_4 where w_2 occurs twice, but w_3 and w_4 do not occur, and X, being in normal form, does not contain such a point. We can compute s as in the proof of Theorem 10 using Lemma 2. Now, we simply contract, one by one, each ℓ_i with large enough i (say, $i > s$) to p. The new right tail (to the right of w_4) is then of the form

$$x_4^R = w_1^{\ell_1} w_2 w_1^{\ell_2} w_2 w_1^{\ell_3} w_2 \cdots w_1^{\ell_s} w_2 w_1^p w_2 w_1^p w_2 w_1^p \ldots,$$

and we can compute such a new tail from the description of the language L.

The case that is left is $j = 3$. We apply the reasoning of the previous paragraph to x_3. We observe that in x_3, either the right tail is periodic with repeating pattern w_1, or we have $\ell'_i \to \infty$ where ℓ'_i is the sequence of gaps in the representation of x_3, and the corresponding claim is true also on the left. In particular, for some h and h', further than h away from w_3 in x_3, no pattern $w_2 w_1^{p-i'} w_2$ for $i' > 0$ occurs, and in either $(x_3)_{[-h',-h]}$ or $(x_3)_{[h,h']}$, the word w_2 occurs at least p times. The existence of h follows as in the previous paragraph, and h' exists because $\overline{\mathcal{O}(x_2)} \subset \overline{\mathcal{O}(x_3)}$.

We can now compute s such that for each $i \geq s$, either $\ell_i, \ell_{i+1} > p$ and $k_i = 2$, or $k_{i'} = 3$ for some i' with $|i' - i| < h$. Take s further large enough that the distance between two occurrences of w_3 is at least $2h' + 2h$. Now, by the assumption on h and h', between any two occurrences of w_3, there are at least p occurrences of w_2 separated by $w_1^{p+i'}$ for some $i' \geq 0$. We contract each $w_1^{p+i'}$ to w_1^p, and then the maximal pattern $(w_2 w_1^p)^t$ between the w_3 to $(w_2 w_1^p)^p$. Clearly, we obtain the same distance and intermediate word between any two occurrences of w_3, and we can compute this repeating pattern. □

5. Some open problems

We showed in Section 3.1 that every substitution generates a subshift which has a decidable model-checking problem for regular languages. It seems likely that one can prove the decidability of model-checking for ω- and ζ-regular languages similarly. While this would be an interesting result, we feel the following is a more interesting generalization:

Question 2. *Is the model-checking problem for context-free languages decidable for subshifts generated by substitutions?*

One can naturally extend context-free languages to infinite words, just like regular languages. As with regular languages, there are multiple ways to do this, and it is an interesting question whether their model-checking problems are always decidable for substitutive subshifts. We give a rough description of the approach of [22], where the *Muller context-free languages* or MCFL, are defined. We take the family of countable finitely branching trees where each node carries a label from a finite set, the possible sets of children of each node are determined by a local rule, and for each infinite childward path, the set of labels occurring infinitely many times is an element of a prescribed set (the Muller condition). The language associated to it is the subset of countable totally ordered sets labeled by S (written $S^{\#}$), defined by taking the induced ordering and labeling of the frontier, and restricting to words over terminal symbols.

For the model-checking problem, one should further intersect with $S^{\mathbb{Z}}$. Thus, we are interested in the model-checking problem for the MCFL languages contained in $S^{\mathbb{Z}}$. These languages are essentially bi-infinite concatenations of context-free languages governed by a suitable finite-state machine.

It is known to be decidable whether a subshift generated by a substitution is finite. Recently, this was shown for the more general class of substitutive images of subshifts generated by substitutions [23, 24], called HD0L. The image of a quasiminimal subshift in a non-erasing substitution is quasiminimal, and thus these subshifts fit into our framework (at least in the case of syndetic long symbols). Which model-checking problems are decidable for this class?

In the topic of countable quasiminimal subshifts, it seems that the proof of Theorem 12 extends to show that in general, when the lattice of subshifts of X is totally ordered and X is countable and recursive, then its model-checking problem for star-free languages is decidable. However, the contraction process becomes harder to describe.

The condition $\mathrm{Q}(X) = 3$ automatically means X is at least weakly transitive and its maximal proper subshift is minimal. Further, subshifts with $\mathrm{Q}(X) = 3$ can be seen to split into the following 4 types:

1. X is countable,
2. X is uncountable but contains an isolated point,
3. X has no isolated point but its maximal proper subshift is finite, or
4. X has no isolated point and its maximal proper subshift is infinite.

As Proposition 8 shows, decidability questions are trivial for subshifts of type 1, and the proof of Theorem 3 shows that almost nothing is decidable for subshifts of type 4.

Problem 1. *Which model-checking problems are decidable for subshifts of type 2 or 3?*

We conjecture that in the countable case, the model-checking problem is decidable for local languages in general.

Conjecture 2. *Let X be a countable quasiminimal recursive subshift. Then the model-checking problem of X for local languages is decidable.*

Another interesting direction is the study of decidability when restricting to minimal systems, but considering the model-checking problem of harder languages. We gave an example of a minimal subshift whose model-checking problem is Σ^0_1-complete for context-free languages accepted by a deterministic pushdown automaton. For this, we used a subshift of the well-known Dyck shift. The high-level idea giving nonregularity in this example is the balancing of brackets, which inherently requires the use of a full stack. We do not know whether there are examples where the stack contains only unary symbols. A slightly more formal way to ask this question is the following.

Question 3. *Is the model-checking problem of languages accepted by one-counter machines decidable for every minimal Π^0_1 subshift?*

The precise definition of one-counter machines used in the question is left implicit – it is part of the question, as we do not know what implications such choices have. The question is also interesting for *strictly ergodic subshifts* – minimal subshifts where words occur with well-defined frequencies. Here, one might (perhaps mistakenly) expect that on long enough words, the counter is decremented or incremented at a roughly constant rate. All we know about one-counter machines is that a variant of the language $a^n b^n$ is unlikely to yield a hard model-checking instance, since for any word w, w^n can occur in an infinite minimal subshift for only finitely many n.

Of course, there are many natural classes in both formal language theory and complexity theory which fall between regular and context-free languages. Further decidability and undecidability results for the model-checking problem for minimal systems for any such class might be an interesting research direction.

Acknowledgements

The author would like to thank Ilkka Törmä for detailed discussions on the topic, and the referees for their comments, in particular the suggestion of adding Table 1. The author was partially supported by the Academy of Finland Grant 131558, CONICYT Proyecto Anillo ACT 1103, by Basal project CMM, Universidad de Chile and the FONDECYT project 3150552.

[1] J. Auslander, Minimal Flows and Their Extensions, North-Holland Mathematics Studies, Elsevier Science, 1988.
URL `http://books.google.cl/books?id=e3wFvPvpWvwC`

[2] P. Kůrka, Topological and symbolic dynamics, Vol. 11 of Cours Spécialisés [Specialized Courses], Société Mathématique de France, Paris, 2003.

[3] J.-C. Delvenne, P. Kůrka, V. Blondel, Decidability and universality in symbolic dynamical systems, Fund. Inform. 74 (4) (2006) 463–490.

[4] V. Salo, Quasiminimality, compactness and derivatives of subshifts, work in progress.

[5] D. Lind, B. Marcus, An introduction to symbolic dynamics and coding, Cambridge University Press, Cambridge, 1995. doi:10.1017/CBO9780511626302.
URL `http://dx.doi.org/10.1017/CBO9780511626302`

[6] B. P. Kitchens, Expansive dynamics on zero-dimensional groups., Ergodic Theory Dyn. Syst. 7 (1987) 249–261. doi:10.1017/S0143385700003989.

[7] J. E. Hopcroft, R. Motwani, J. D. Ullman, Introduction to Automata Theory, Languages, and Computation (3rd Edition), Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006.

[8] J.-C. Delvenne, P. Kůrka, V. D. Blondel, Computational universality in symbolic dynamical systems, in: Machines, computations, and universality, Vol. 3354 of Lecture Notes in Comput. Sci., Springer, Berlin, 2005, pp. 104–115. doi:10.1007/978-3-540-31834-7_8.
URL http://dx.doi.org/10.1007/978-3-540-31834-7_8

[9] D. Perrin, J. Pin, Infinite Words: Automata, Semigroups, Logic and Games, Pure and Applied Mathematics, Elsevier Science, 2004.
URL https://books.google.fi/books?id=S7hHhJc4iNgC

[10] A. Ballier, B. Durand, E. Jeandel, Structural aspects of tilings, in: P. W. Susanne Albers (Ed.), Proceedings of the 25th Annual Symposium on the Theoretical Aspects of Computer Science, IBFI Schloss Dagstuhl, Bordeaux, France, 2008, pp. 61–72, 11 pages.
URL http://hal.archives-ouvertes.fr/hal-00145800

[11] G. Sacks, Higher recursion theory, Perspectives in mathematical logic, Springer-Verlag, 1990.
URL http://books.google.fi/books?id=aOvuAAAAMAAJ

[12] D. Cenzer, A. Dashti, F. Toska, S. Wyman, Computability of countable subshifts in one dimension, Theory of Computing Systems 51 (3) (2012) 352–371. doi:10.1007/s00224-011-9358-z.
URL http://dx.doi.org/10.1007/s00224-011-9358-z

[13] The On-Line Encyclopedia of Integer Sequences, published electronically at http://oeis.org, 2014, Sequence A007814.

[14] W. Thomas, Automata on infinite objects, in: Handbook of theoretical computer science, Vol. B, Elsevier, Amsterdam, 1990, pp. 133–191.

[15] P. Kůrka, On topological dynamics of Turing machines, Theoret. Comput. Sci. 174 (1-2) (1997) 203–216. doi:10.1016/S0304-3975(96)00025-4.
URL http://dx.doi.org/10.1016/S0304-3975(96)00025-4

[16] R. McNaughton, S. Papert, Counter-free automata. With an appendix by William Henneman., Research Monograph No.65. Cambridge, Massachusetts, and London, England: The M. I. T. Press. XIX, 163 p. $ 12.50 (1971). (1971).

[17] S. Bezuglyi, J. Kwiatkowski, K. Medynets, Aperiodic substitution systems and their bratteli diagrams, Ergod. Th. & Dynam. Sys.

[18] V. Salo, I. Törmä, Complexity of Conjugacy, Factoring and Embedding for Countable Sofic Shifts of Rank 2, ArXiv e-printsarXiv:1408.6695.

[19] R. H. Herman, I. F. Putnam, C. F. Skau, Ordered bratteli diagrams, dimension groups and topological dynamics, International Journal of Mathematics 03 (06) (1992) 827–864. arXiv:http://www.worldscientific.com/doi/pdf/10.1142/S0129167X92000382, doi:10.1142/S0129167X92000382.
URL http://www.worldscientific.com/doi/abs/10.1142/S0129167X92000382

[20] M. Hochman, On the dynamics and recursive properties of multidimensional symbolic systems, Invent. Math. 176 (1) (2009) 131–167. doi:10.1007/s00222-008-0161-7.
URL http://dx.doi.org/10.1007/s00222-008-0161-7

[21] A. Ballier, E. Jeandel, Computing (or not) Quasi-Periodicity Functions of Tilings, ArXiv e-printsarXiv:1012.1222.

[22] Z. Ésik, S. Iván, On Müller context-free grammars, in: Y. Gao, H. Lu, S. Seki, S. Yu (Eds.), Developments in Language Theory, Vol. 6224 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2010, pp. 173–184. doi:10.1007/$978 - 3 - 642 - 14455 - 4_17$.
URL $http://dx.doi.org/10.1007/978-3-642-14455-4_17$

[23] I. Mitrofanov, A proof for the decidability of HD0L ultimate periodicity, ArXiv e-printsarXiv:1110.4780.

[24] F. Durand, Decidability of the HD0L ultimate periodicity problem., RAIRO - Theor. Inf. and Applic. 47 (2) (2013) 201–214.
URL http://dblp.uni-trier.de/db/journals/ita/ita47.html#Durand13