Algorithms for Symbolic Substitutions and Pisot Substitutions

Scott Balchin
Department of Mathematics
University of Leicester
University Road, Leicester, LE1 7RH
United Kingdom
slb85@le.ac.uk

Dan Rust
Department of Mathematics
Universität Bielefeld
Bielefeld, Universitätsstraße, D-33615
Germany
drust@math.uni-bielefeld.de

Abstract

We introduce a GUI fronted program that can compute combinatorial properties and topological invariants of recognisable and primitive symbolic substitutions on finite alphabets, their subshifts and their associated tiling spaces. We introduce theory from symbolic dynamics along with pseudocode highlighting the algorithms that we have implemented into the GUI. Grout is written using C++ and its standard library. We implement a check to verify that no counterexample to the strong coincidence conjecture exists for the first 12,573,955 irreducible Pisot substitutions on three letters and 15,001 Pisot substitutions on four letters.

Supplementary Resources

Grout is available to download for Windows and Mac OSX along with the supporting documentation at the following URL

www2.le.ac.uk/departments/mathematics/extranet/staff-material/staff-profiles/scott-balchin

1 Introduction

The study of aperiodic order has a rich history which emerged from the worlds of computer science and mathematical logic when Berger proved the undecidability of the domino prob-

lem in the 1960s [13]. It is now also a topic of general interest to the study of dynamical systems, topology, Diophantine approximation, ergodic theory, computer graphics, mathematical physics and even virology [17, 37, 15, 5, 32, 12, 42]. Once it was discovered that sets of tiles exist which can only tile the plane aperiodically, a flurry of advances quickly followed, culminating in the celebrated discovery of the famous *Penrose tilings* [35] and in the discovery of quasicrystals [38] for which Schectman was awarded the Nobel prize in Chemistry in 2011.

The seminal work of Anderson and Putnam [4] led to a new algebraic topological approach to the study of aperiodic tilings associated to a particular method of generating tilings of the plane known as the *substitution method*. It is this method, but restricted to a 1-dimensional analogue, which we address here. In this setting, 1-dimensional tilings are combinatorially equivalent to bi-infinite sequences over an alphabet whose letters label the tiles in order to distinguish them. There are still long standing conjectures in the 1-dimensional setting which need to be settled; principal among them being the *Pisot conjecture* [2] and related problems.

It is hoped that the use of this program will make testing conjectures in tiling theory and symbolic substitutional dynamics more efficient, as well as allowing for the confirmation of hand calculations and comparing different methods of calculation (especially methods of calculating cohomology). Analysis of large data sets which can be potentially generated by the Grout source code, and the recognition of underlying patterns in the data may also aid to further the theory. There are several hundred entries containing the term *morphism* in the Online Encyclopedia of Integer Sequences [40], where we use the term substitution in place of the term morphism, which occurs more frequently in the computer science literature. These sequences are of great interest and have been extensively studied.

The GUI front end for Grout is powered by Qt[1]. Grout has been designed with user experience in mind and includes many ease-of-use properties such as the ability to save and load examples, and convenient methods of sharing examples with other users via short strings that encode a substitution. There is also an option to export all of the data that has been calculated to a pre formatted LaTeX file including all the TikZ code for the considered complexes. This should be useful for those needing to typeset such diagrams in the future by fully automating the generation of diagrams in TikZ.

In Section 1.1 we introduce basic notions of subshifts (sequence spaces) and tiling spaces associated to substitutions on finite alphabets. In Section 2 we introduce the relevant theory from symbolic dynamics along with pseudocode for most of the non-trivial components of Grout that have been implemented. In this section we also include a new fully deterministic check for recognisability for primitive substitutions, for which the algorithm is simple enough to check by hand in a reasonable amount of time for small substitutions. Section 3 will cover specifically those methods implemented to compute cohomology for tiling spaces. Throughout, we give instances of these methods being applied to well-known example substitutions. In Section 4 we focus our efforts on applying the functions of Grout to the study of Pisot substitutions and a search for counterexamples to a conjecture related to them. In particular, we perform a search for counterexamples to the strong coincidence conjecture, a variant of

the Pisot conjecture [9]. No counterexample is found within the first 12,573,955 irreducible Pisot substitutions on three letters and the first 15,001 irreducible Pisot substitutions on four letters (with respect to a lexicographic order).

1.1 Background

Definition 1. Let $\mathcal{A} = \{a^1, \ldots, a^l\}$ be a finite alphabet on l symbols, and for all positive integers n let $\mathcal{A}^n$ be the set of length n words in $\mathcal{A}$. Denote the union of these as $\mathcal{A}^+ = \bigcup_{n\geq 1} \mathcal{A}^n$. If, further, the empty word ϵ is included, we denote the union $\mathcal{A}^+ \cup \{\epsilon\}$ by $\mathcal{A}^*$. A *substitution* ϕ on $\mathcal{A}$ is a function $\phi\colon \mathcal{A} \to \mathcal{A}^+$ which assigns to each letter a in $\mathcal{A}$ a non-empty word $\phi(a)$ whose letters are elements in $\mathcal{A}$. We extend ϕ to a function $\phi\colon \mathcal{A}^+ \to \mathcal{A}^+$ by concatenation; given a word $w = w_1 \ldots w_n \in \mathcal{A}^n$, we set $\phi(w) = \phi(w_1) \ldots \phi(w_n)$. In this way, we can consider finite iterates of the substitution ϕ^n acting on $\mathcal{A}^+$.

Example 2. Let $\mathcal{A} = \{0, 1\}$ and let $\phi\colon \mathcal{A} \to \mathcal{A}^+$ be given by $\phi\colon 0 \mapsto 01, 1 \mapsto 0$. By iterating ϕ on the initial seed letter 0 we get a nested sequence of words

$$0 \mapsto 01 \mapsto 010 \mapsto 01001 \mapsto 01001010 \mapsto 0100101001001 \mapsto \cdots$$

whose limit sequence is the *Fibonacci word* A003849,A096270

$$0100101001001010010100100101001001010010100100101001010\ldots$$

For our purposes, we do not combinatorially distinguish this sequence from the version on the alphabet $\{1, 2\}$ given by $1 \mapsto 12, 2 \mapsto 1$ A003842 or on the alphabet $\{a, b\}$ given by $a \mapsto ab, b \mapsto a$, etc.

Example 3. Let $\mathcal{A} = \{0, 1\}$ and let $\phi\colon \mathcal{A} \to \mathcal{A}^+$ be given by $\phi\colon 0 \mapsto 01, 1 \mapsto 10$. In this case, the limit sequence given by repeated substitution on the seed letter 0 is known as a *Thue-Morse sequence* A010060

$$0110100110010110100101100110100110010110011010010011\ldots$$

See also the version with seed letter 1 instead of 0 A010059 and the version on the alphabet $\{1, 2\}$ A001285

Note that, in general, a sequence of iterates on a letter does not necessarily form a nested sequence of superwords extending to the right, but if the substitution is suitably nice (soon to be introduced as primitive), then there exists some finite power of the substitution and some seed letter in the alphabet for which this property does hold. Infinite sequences which are limits of such collections of superwords are called *fixed points* of the substitution and bi-infinite analogues also exist (which we study here).

Definition 4. Let $\phi\colon \mathcal{A} \to \mathcal{A}^+$ be a substitution. We say a word $w \in \mathcal{A}^*$ is *admitted* by the substitution ϕ if there exists a letter $a \in \mathcal{A}$ and a natural number $n \geq 0$ such that w is a subword of $\phi^n(a)$ and denote by $\mathcal{L}^n \subset \mathcal{A}^n$ the set of all words of length n which are admitted by ϕ. Our convention is that the empty word ϵ is admitted by all substitutions. We form the *language* of ϕ by taking the set of all admitted words $\mathcal{L} = \bigcup_{n\geq 0} \mathcal{L}^n$.

We say a bi-infinite sequence $s \in \mathcal{A}^{\mathbb{Z}}$ is *admitted* by ϕ if every subword of s is admitted by ϕ and denote by X_ϕ the set of all bi-infinite sequences admitted by ϕ. The symbol s_i denotes the label assigned to the ith component of the sequence s. The set X_ϕ has a natural (metric) topology inherited from the product topology on $\mathcal{A}^{\mathbb{Z}}$ and a natural shift map $\sigma\colon X_\phi \to X_\phi$ given by $\sigma(s)_i = s_{i+1}$. We call the pair (X_ϕ, σ) the *subshift* associated to ϕ and we will often abbreviate the pair to just X_ϕ when the context is clear.

We say ϕ is a *periodic* substitution if X_ϕ is finite, and say ϕ is aperiodic otherwise. If ϕ is *aperiodic* and has a property which will be introduced later known as *primitivity*, then X_ϕ is a Cantor set (in particular X_ϕ is non-empty) and σ is a minimal action on X_ϕ - that is, the only closed shift-invariant subsets of X_ϕ are the empty set $\emptyset$ and the subshift itself X_ϕ. Equivalently, the orbit of every point under σ is dense in X_ϕ.

Definition 5. Let ϕ be a substitution on the alphabet $\mathcal{A}$ with associated subshift X_ϕ. The *tiling space* associated to ϕ is the quotient space

$$\Omega_\phi = (X_\phi \times [0,1])/\sim$$

where $\sim$ is generated by the relation $(s, 0) \sim (\sigma(s), 1)$.

One should imagine the point $(s, t) \in \Omega_\phi$ as being a partition or tiling of the real line $\mathbb{R}$ into labelled unit-length intervals (called tiles), where the labels are determined by the letters appearing in s, as shown in Figure 1, and the origin of $\mathbb{R}$ is shifted a distance t to the right from the left of the tile labelled by the letter s_0. Two tilings T and T' in Ω_ϕ are considered ϵ-*close* in this topology if, after a translate by a distance at most ϵ, the tiles around the origin in $T' - \epsilon$ within a ball of radius $1/\epsilon$ lie exactly on top of the tiles around the origin in T within a ball of the same radius and share the same labels.

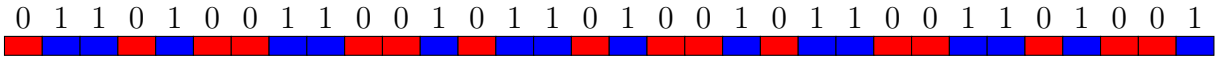

Figure 1: A portion of the tiling associated with the Thue-Morse sequence.

If ϕ is primitive and aperiodic, then Ω_ϕ is a compact connected metric space which fibers over the circle with Cantor set fibers. The natural translation $T \mapsto T + t$ for $t \in \mathbb{R}$ equips Ω_ϕ with a continuous $\mathbb{R}$ action which is minimal as long as ϕ is primitive. In this respect, tiling spaces are closely related to the more well-known spaces, the solenoids.

Associated to any topological space X is a collection of groups $\check{H}^*(X)$ called the *Čech cohomology* of X. There are many standard texts providing an introduction to Čech cohomology [16]. Čech cohomology is an important topological invariant of tiling spaces and

it is of general interest to be able to calculate and study these groups. Grout implements three different methods for calculating the cohomology of tiling spaces associated to symbolic substitutions on finite alphabets.

1. The method of Barge-Diamond complexes [10]
2. The method of Anderson-Putnam complexes [4]
3. The method of forming an equivalent *left proper* substitution [21]

All three outputs are algebraically equivalent—that is, they represent isomorphic groups—but it is not always obvious that this is the case given the presentations that each method produces. This disparity between presentations of results for the equivalent methods was one of the major motivating factors for developing Grout. These cohomologies are extremely laborious to calculate by hand for large alphabets unless special criteria are met.

2 Grout and its functions

2.1 Substitution Structure

We begin by outlining how we encode a substitution rule into Grout and how we implement the substitution rule. In general, we have done most of the implementation by string manipulation methods.

We use a class *sub* which has as its element a vector of strings. We always assume that our alphabet is ordered $a, b, c, \ldots$—for technical reasons, Grout takes as input symbols from the Latin alphabet, rather than numerals. The first entry of a *sub* class vector is $\phi(a)$, the second is $\phi(b)$ and so on. To validate the input we check that the number of unique characters appearing in all of the $\phi(x)$ is equal to the length of the alphabet, which is the length of the vector. The GUI also employs the use of regular expressions to prevent illegal characters from being entered.

Example 6 (Fibonacci Substitution)**.** The Fibonacci tiling is given by a substitution rule on the alphabet $\mathcal{A} = \{a, b\}$ and is defined as

$$\phi\colon \begin{cases} a \mapsto & b \\ b \mapsto & ba \end{cases}$$

The Fibonacci substitution is our main example used throughout the paper.

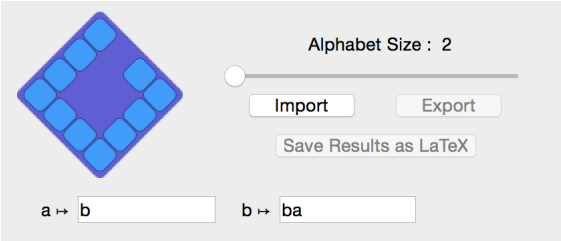

Figure 2: The Fibonacci substitution entered into Grout

Next we implement a way to perform an iteration of ϕ on a string. We do not include any checks to validate that the string can be iterated on, as all strings that pass to this function are created by the program itself, and therefore valid.

Algorithm 1 Substitution functions

```
1: function iterate(string rhs)
2:     result = empty string
3:     for each character x in rhs do
4:         append φ(x) to result
5:     output result
```

2.2 Substitution Matrices and their Properties

Definition 7. Let $\mathcal{A} = \{a^1, \ldots, a^l\}$ be an alphabet with a substitution $\phi\colon \mathcal{A} \to \mathcal{A}^+$, then ϕ has an associated *substitution matrix* M_ϕ of dimension $l \times l$ given by setting m_{ij} to be the number of times that the letter a^i appears in $\phi(a^j)$.

Example 8. Let $\mathcal{A} = \{0, 1\}$ and let $\phi\colon \mathcal{A} \to \mathcal{A}^+$ be given by $\phi\colon 0 \mapsto 01, 1 \mapsto 00$ whose limit sequence with seed 0 is the *period doubling sequence*A096268. A035263

$$0100010101000100010001010100010 1 \ldots$$

This substitution has associated substitution matrix $M_\phi = \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix}$.

We do not give the algorithm for constructing the substitution matrix; the definition can be taken as a pseudo-algorithm. We implement a square matrix class to work with the substitution matrix. The first property that we check for a substitution matrix is primitivity.

Definition 9. A substitution $\phi\colon \mathcal{A} \to \mathcal{A}^+$ is called *primitive* if there exists a positive natural number p such that the matrix M_ϕ^p has strictly positive entries. Such a matrix M is also called *primitive*. This condition is equivalent to asking that there is a positive natural number p such that for all $a, a' \in \mathcal{A}$ the letter a' appears in the word $\phi^p(a)$.

All examples which have been presented so are easily checked to be primitive.

Example 10. Let $\mathcal{A} = \{0, 1\}$ and let $\phi\colon \mathcal{A} \to \mathcal{A}^+$ be given by $\phi\colon 0 \mapsto 0010, 1 \mapsto 1$ whose limit sequence with seed 0 is the *non-primitive Chacon sequence* A049320, A049321

$$0010001010010001000101001010010001010010\ldots$$

As the name suggest, the non-primitive Chacon sequence fails to be primitive as its substitution matrix $M_\phi = \left[\begin{smallmatrix} 3 & 0 \\ 1 & 1 \end{smallmatrix}\right]$ is lower triangular and so has no power which is strictly positive in all entries.

We use the first definition to check the primitivity of our substitutions. To check this condition on M_ϕ, we count the number of zeros in the matrix and if the number of zeros reaches 0 then we conclude that the substitution is primitive. If not, then square the matrix and recount the number of zeros. If the number of zeros does not change then we can conclude that the substitution is not primitive. This means that this check always halts.

Algorithm 2 Primitivity check

```
1:  function primitive
2:      matrix = substitution matrix of φ
3:      while true do {
4:          a = number of zeros in matrix
5:          matrix = matrix × matrix
6:          b = number of zeros in matrix
7:          if a=b and a!=0
8:              output false
9:          if b=0
10:             output true
11:     }
```

We check primitivity for all substitutions before we do calculations on them as if the substitution is not primitive many of the subsequent methods do not work, or return false positive/negative results. Grout always displays whether a given substitution is primitive or not and also outputs the substitution matrix if asked to do so.

The next properties that can be calculated from a substitution matrix are the natural tile frequencies and tile lengths of the substitution. This requires us to compute the eigenvalues of the matrix. We have implemented the QR method for computing the eigenvalues (for example see [28]). This gives us approximations to the real eigenvalues, and for the complex ones we simply give the conjugate pairs by their absolute values, and we give the results to two decimal places. The eigenvalues of a substitution matrix may be printed out by ticking the eigenvalues box.

Proposition 11 (Perron-Frobenius)**.** *Let M be a primitive matrix.*

i *There is a positive real number λ_{PF}, called the* Perron-Frobenius eigenvalue*, such that λ_{PF} is a simple eigenvalue of M and any other eigenvalue λ is such that $|\lambda| < \lambda_{PF}$.*

ii *There exist left and right eigenvectors, called the* left and right Perron-Frobenius eigenvectors*, $\mathbf{l}_{PF}$ and $\mathbf{r}_{PF}$ associated to λ_{PF} whose entries are all positive and which are unique up to scaling.*

Given the above theorem, it is natural to ask what information is contained in the PF eigenvalue and eigenvectors of M_ϕ for a primitive substitution ϕ.

If we were to assign a length to the tiles labelled by each letter, then we would hope for such a length assignment to behave well with the given substitution. The left PF eigenvector offers a natural choice of length assignments. If we assign the length $(\mathbf{l}_{PF})_i$ to the letter a^i, the ith component of the left PF eigenvector, then we can replace our combinatorial substitution by a geometric substitution. This geometric substitution expands the tile with label a^i by a factor of λ_{PF} and then partitions this new interval into tiles of lengths and labels given according to the combinatorial substitution. In order to give a unique output, Grout normalises the left PF eigenvector so that the smallest entry is 1.

The information contained in the right PF eigenvector is also useful. The right PF eigenvector, once normalised so that the sum of the entries is 1, gives the relative frequencies of each of the letters appearing in any particular bi-infinite sequence which is admitted by ϕ. That is, if $|w|$ is the length of the word w, $|w|_i$ is the number of times the letter a^i appears in the word w, and letting $s_{[-k,k]} = s_{-k} \dots s_{-1} s_0 s_1 \dots s_k$, then

$$\lim_{k\to\infty} |s_{[-k,k]}|/|s_{[-k,k]}|_i = (\mathbf{r}_{PF})_i$$

for any $s \in X_\phi$.

Example 12. Let $\mathcal{A} = \{0, 1, 2, 3\}$ and let $\phi\colon \mathcal{A} \to \mathcal{A}^+$ be given by $\phi\colon 0 \mapsto 01, 1 \mapsto 02, 2 \mapsto 31, 3 \mapsto 32$ whose limit sequence with seed 0 is the fixed point of the *Rudin-Shapiro substitution* A100260, A073057, A020985

$$0102013101023202010201313231010 2\dots$$

which is related to the *paper-folding sequence* A014577, A014709, A014710,A106665 under a local block-replacement rule. The substitution matrix for ϕ has PF-eigenvalue $\lambda_{PF} = 2$ and PF-eigenvectors $\mathbf{l}_{PF} = (1, 1, 1, 1)$ and $\mathbf{r}_{PF} = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$. So from the left PF-eigenvector we assign unit length to all four tiles, and from the normalised right PF-eigenvector we can tell that each letter appears with equal frequency of $\frac{1}{4}$.

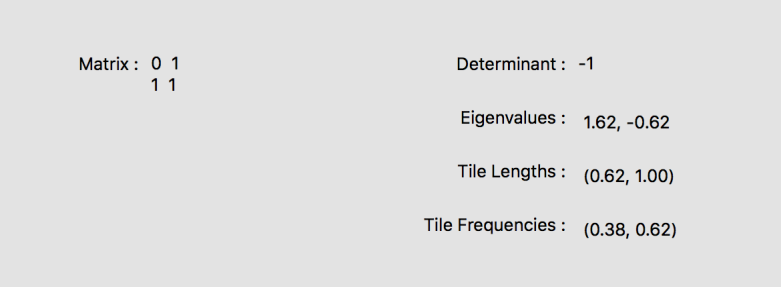

Figure 3: The results for the matrix calculations for the Fibonacci substitutions

2.3 Enumerating nLetter Words

Now that we have introduced the basic structure of the substitutions, and discussed the problem of primitivity and other matrix related calculations, we can discuss the first main function in Grout.

Definition 13. Given a substitution $\phi\colon \mathcal{A} \to \mathcal{A}^+$, we define the *complexity function* at n to be the number of unique n-letter words admitted by ϕ. We denote this function by $p_\phi(n)$, and so $p_\phi(n) = |\mathcal{L}_n|$.

Example 14. The period doubling sequence from Example 8 is found to have a complexity function A275202 whose first few terms are

$$p_\phi(n)\colon 2, 3, 5, 6, 8, 10, 11, 12, 14, 16, 18, 20, 21, 22, 23, 24, 26, 28, 30, 32, 34, 36, 38, 40, 41, 42\ldots$$

The complexity function of a sequence is a useful measure of disorder in a sequence whose asmyptotic growth rate gives topological and dynamical information about the subshift and tiling space of a sequence [31, 25]. One is usually interested in either a deterministic formula for p_ϕ or information about the asymptotic growth rate of p_ϕ such as polynomial degree; Grout can be used to at least give circumstantial evidence for these, though has no means of calculating either (this appears to be a very difficult problem in general). A useful survey article by Ferenczi collates many of the key results on complexity functions [26].

Of particular interest are the number of 2 and 3 letter words admitted by a substitution, as we use them later to compute cohomology. Our function not only enumerates the number of n-letter words, but prints out these words as required. The algorithm uses C++ sets as a data structure to store the n-letter words as it is automatically ordered and does now allow repetitions leading to fast computation. We start by generating a length-m admitted seed word w such that $m \geq n$, and count all unique n-letter words appearing as subwords of the seed. We then apply ϕ to the seed and add all new n-letter words to the result. At each stage we count the size before and after adding the new words. If the size does not change we can stop, as no new n-letter words are generated after a step without any new n-letter words. It follows that the value $p_\phi(n)$ is computable in finite time for any fixed $n \geq 1$.

Algorithm 3 Finding all n-letter words

```
1:  function nlw(int n)
2:      result = empty ordered set
3:      seed = 'a'
4:      while seed length < n do
5:          seed = iterate(seed)
6:
7:      difference = 1
8:      while difference != 0 do {
9:          a = cardinality of result
10:         seed = iterate(seed)
11:         for each n length word w in seed do
12:             append w to result
13:         b = cardinality of result
14:         difference = b-a
15:     }
16:     output result
```

Example 15. It is well known that the complexity for the Fibonacci substitution satisfies $p_\phi(n) = n + 1$, and we can verify this for any value of n by computing its complexity in Grout.

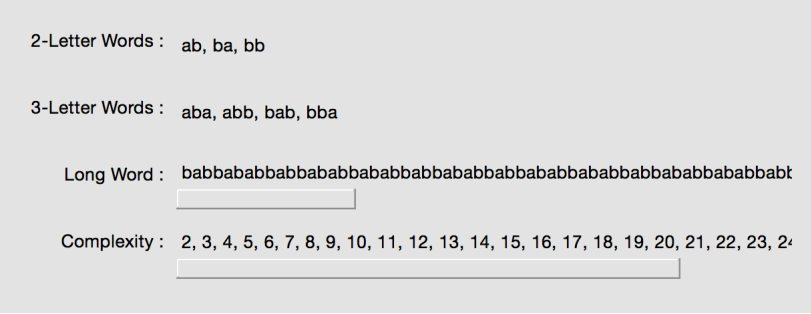

Figure 4: The words results for the Fibonacci substitution displayed in Grout

2.4 Barge-Diamond and Anderson-Putnam Complexes

Grout has the ability to output two simplicial complexes as PDFs (provided that the user has PDFLaTeX installed). The first of these is the *Barge-Diamond complex* [10], which is the key tool used in one of the methods of computing the Čech cohomology of the tiling space.

Definition 16. Let $\mathcal{A} = \{a^1, \ldots, a^l\}$ be an alphabet with a substitution $\phi\colon \mathcal{A} \to \mathcal{A}^+$, then we construct the *Barge-Diamond complex* of ϕ as follows. We have two vertices for each a^i,

an *in* node v_i^+ and an *out* node v_i^-. We draw an edge from v_i^+ to v_i^- for all i. Then for all two letter words $a^i a^j \in \mathcal{L}^2$ admitted by ϕ, we draw an edge from v_i^- to v_j^+.

Example 17. As we have seen previously, the only two letter words admitted by the Fibonacci substitution are ab, ba and bb, this gives us the following Barge-Diamond complex output in Grout.

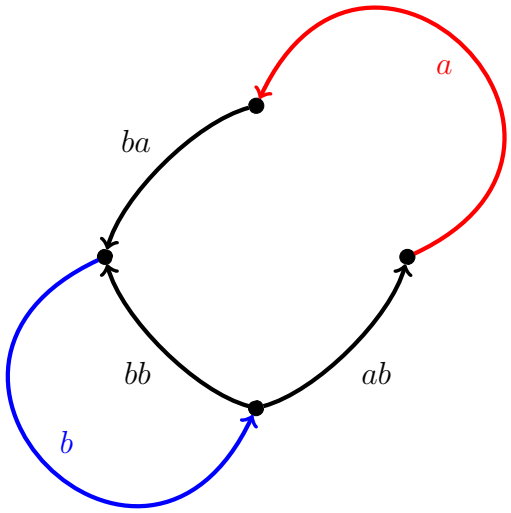

Figure 5: The Barge-Diamond complex for the Fibonacci substitution.

The other complex that we consider for a substitution is (a variant of) the *collared Anderson-Putnam complex* [4] which is again the key tool used in one of the methods of computing Čech cohomology of the tiling space. For brevity we will often shorten this to the *AP complex*. This particular definition is based on what Gähler and Maloney call the *Modified Anderson-Putnam complex* [27]. The AP complex is constructed in a similar fashion to the Barge-Diamond complex, but makes use of both the two and three letter words.

Definition 18. Let $\mathcal{A} = \{a^1, \ldots, a^l\}$ be an alphabet with a substitution $\phi\colon \mathcal{A} \to \mathcal{A}^+$, then we construct the *Anderson-Putnam complex* of ϕ as follows. We have a vertex v_{ij} for each two letter word $a^i a^j \in \mathcal{L}^2$ admitted by ϕ. We draw an edge from v_{ij} to v_{jk} if and only if the three letter word $a^i a^j a^k$ is admitted by ϕ.

Remark 19. One should note that this modified AP complex is slightly different to the definition originally introduced by Anderson and Putnam. In particular, the original definition distinguishes between different occurrences of a two letter word $a^i a^j$ if the occurrences of three letter words containing as a subword $a^i a^j$ do not overlap on some admitted four letter word. For example, if the language of a substitution included the two letter word ab, the three letter words xab, yab, abw, abz, and the four letter words $xabw, yabz$ but the words

$xabz$ and $yabw$ did not belong to $\mathcal{L}$, then the original definition of the AP complex would have two instances of vertices with the label ab, say $(ab)_1$ and $(ab)_2$. In our definition, these vertices are identified, so that $(ab)_1 \sim (ab)_2 \sim ab$. An example of such a substitution is given by $\phi\colon a \mapsto bc$, $b \mapsto baab$, $c \mapsto caac$ where we label exactly one vertex with the label aa, but the original definition would require we include two distinct vertices labelled $(aa)_1$ and $(aa)_2$.

In our discussion of cohomology calculated via AP complexes in Section 3.2, we use this version of the AP complex to describe the performed calculations, and Grout implements this particular method. It would have been possible to use the original definition, or one of the many variant AP complexes that have been defined in the literature. There are at least three such variants discussed in [27], of varying complexities and situations in which they can be used.

Example 20.

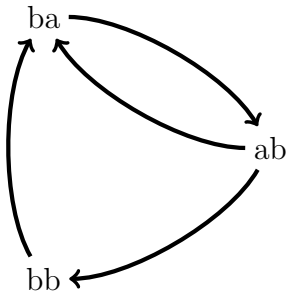

Figure 6: The Modified Anderson-Putnam complex for the Fibonacci substitution

2.5 Recognisability

Definition 21. Let ϕ be a substitution on the alphabet $\mathcal{A}$. We say ϕ is *recognisable* if for every bi-inifnite sequence $s \in X_\phi$ admitted by ϕ there is a unique way of decomposing s into words of the form $\phi(a)$ for $a \in \mathcal{A}$. That is, up to a finite shift, there exists a unique bi-infinite sequence $\ldots a_{-1}a_0a_1 \ldots$ such that $s = \ldots \phi(a_{-1})\phi(a_0)\phi(a_1)\ldots$ and so we can *recognise* which substituted letter each letter in s has come from.

Equivalently, we say ϕ is recognisable if there exists a natural number $K \geq 1$ such that for all admitted words $w \in \mathcal{L}$ with $|w| > 2K$, there exist unique words x, y of length $|x|, |y| \leq K$ and a unique admitted word $u \in \mathcal{L}$ such that $w = x\phi(u)y$.

Recognisability is an important property of a tiling as many of the tools used to study the topology of the associated tiling spaces rely on recognisability as a hypothesis, much like primitivity. Recognisability of a primitive substitution is equivalent to aperiodicity of the subshift X_ϕ [34], and we make use of this result to decide recognisability. The algorithm designed to determine if a given substitution is recognisable relies on finding a fixed letter and return words to that fixed letter.

Definition 22. Given a substitution ϕ on an alphabet $\mathcal{A}$, the letter a is said to be *fixed* (on the left) of *order* k if there exists some integer k such that $\phi^k(a) = au$ for some word u. Every substitution has at least one fixed letter and the value of k for such a letter is bounded by the size of the alphabet.

Definition 23. Given a fixed letter a, a *return word* to a is a word v such that $v = au$ for some (possibly empty) word $u \in (\mathcal{A} \setminus \{a\})^*$, and va is an admitted word of the substitution.

If ϕ is primitive then, due to the primitivity of the substitution, the set of return words to any letter is finite. One can see this directly from the definition of primitivity, or as a consequence of the *linear recurrence* of the subshift X_ϕ, shown by Damanik and Lenz [20]. We omit a definition of linear recurrence here.

Algorithm 4 Finding all return words to fixed letter f

```
1:  function returnwords(character f)
2:      result = empty ordered set
3:      length = 2
4:      while new return words are being added do {
5:          nwords = nlw(length)
6:          for all words w in nwords do {
7:              if last character of w = first character of w= f and w has no other f appearing
    do
8:                  append w with the last f removed to result
9:          }
10:         length = length + 1
11:     }
12:     output result
```

We use these return words to determine whether a substitution is recognisable or not. The following proposition appears in a paper of Harju & Linna [29].

Proposition 24. *Let ϕ be a primitive substitution on $\mathcal{A}$ and let a be a fixed letter. Let $\mathcal{R}$ be the set of all return words to a. So $\mathcal{R} = \{v \mid v = au, aua \in \mathcal{L}, u \in (\mathcal{A} \setminus \{a\})^*\}$. The substitution ϕ is not recognisable if and only if, for all $v, v' \in \mathcal{R}$, there exists a $p \geq 1$ such that $\phi^p(vv') = \phi^p(v'v)$.*

As $\mathcal{R}$ is finite, and together with the next proposition which appears in work of Culik [19] and also Ehrenfeucht & Rozenberg [24], this gives us a finite deterministic check for recognisability.

Proposition 25. *Let ϕ be a substitution on $\mathcal{A}$ and let $|\mathcal{A}| = n$. For words $u, w \in \mathcal{A}^+$, there exists a $p \geq 1$ such that $\phi^p(u) = \phi^p(w)$ if and only if $\phi^n(u) = \phi^n(w)$.*

That is, if some iterated substitution of u and v are ever equal, then their iterates must become equal at least by the nth iteration of the substitution, where n is the size of the alphabet. In the algorithm, k is taken to be the k from the definition of the fixed letter, and n is the size of the alphabet.

Algorithm 5 Recognisability check

```
1: function recognisable
2:     rwords = returnwords(f)
3:     for each word w in rwords do
4:         for each word v ≠ w in rwords do
5:             if(φ^{k×n}(w + v) = φ^{k×n}(v + w))
6:                 output true
7:     output false
```

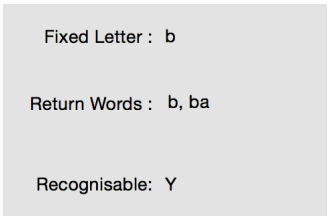

Figure 7: The recognisability results for the Fibonacci substitution

3 Cohomolgy of Tiling Spaces in Grout

3.1 Via Barge-Diamond

Let ϕ be a primitive, recognisable substitution on the alphabet $\mathcal{A}$. Let G be the Barge-Diamond complex of ϕ and let S be the subcomplex of G formed by all those edges labelled with two letter admitted words $a^i a^j$.

Let $\tilde{\phi}$: $S \to S$ be a graph morphism defined in the following way on vertices. Let $l(i)$ and $r(i)$ be such that $\phi(a^i) = a^{l(i)} u a^{r(i)}$ and define $\tilde{\phi}(v_i^+) = v_{l(i)}^+$ and $\tilde{\phi}(v_i^-) = v_{r(i)}^-$. Note that if $a^i a^j$ is admitted by ϕ, then $a^{r(i)} a^{l(j)}$ is also admitted by ϕ, and so $\tilde{\phi}$ is a well defined graph morphism on S. As $\tilde{\phi}(S) \subset S$, we can define the eventual range $ER = \bigcap_{m \geq 0} \tilde{\phi}^m(S)$ (which stabilises after finitely many substitutions).

For this method of computation we make use of the following result attributed to Barge and Diamond [10].

Proposition 26. *There is a short exact sequence*

$$0 \to \check{H}^0(ER) \to \varinjlim M_\phi^T \to \check{H}^1(\Omega_\phi) \to H^1(ER) \to 0.$$

The eventual range ER is a (possibly disconnected) graph, and so $\tilde{H}^0(ER)$ and $H^1(ER)$ are finitely generated free abelian groups of some ranks k and l respectively. Hence we have

Corollary 27.

$$\check{H}^1(\Omega_\phi) \cong \varinjlim M_\phi^T/\mathbb{Z}^k \oplus \mathbb{Z}^l.$$

Grout displays the Čech cohomology using the Barge-Diamond method in the above form.

These results fail in general if ϕ is not primitive or recognisable, and so Grout only performs this calculation after checking these two conditions.

The only involved part of this calculation is finding the eventual range of the Barge-Diamond complex. After we have this we can simply find the number of connected components and use the Euler characteristic to find the reduced cohomology in rank zero and one. Therefore, we now give the algorithm that we use to find the eventual range. We denote by $w[0]$ the first letter of the word ab and $w[1]$ the second letter of ab.

Algorithm 6 Eventual range of the Barge-Diamond complex

```
1:  function eventual range
2:      twoletter = nlw(2)
3:      difference = 1
4:      while difference != 0 do {
5:          temp = empty ordered set
6:          for each word w in twoletter do
7:              append last character of iterate(w[0]) + first character of iterate(w[1]) to temp
8:          difference = cardinality of twoletter - cardinality of temp
9:          twoletter = temp
10:     }
11:     output twoletter
```

3.2 Via Anderson-Putnam

Let ϕ be a primitive, recognisable substitution on the alphabet $\mathcal{A}$. Let K be the Anderson-Putnam complex of ϕ. Anderson and Putnam showed in [4] that the Čech cohomology of Ω_ϕ is determined by the direct limit of an induced map acting on the cohomology of a CW complex. Using the modified AP complex K, Gähler and Maloney showed that this complex, as defined in Section 2.4, can be used in place of the originally defined AP complex. We define the map acting on the AP complex K in the following way.

Again let $l(i)$ and $r(i)$ be such that $\phi(a^i) = a^{l(i)}ua^{r(i)}$. Let E be an edge with label $a^ia^ja^k$ and define $L = |\phi(a^j)|$. Suppose $\phi(a^j) = a_1a_2\dots a_L$. Define a continuous map called the *collared substitution* $\tilde{\phi}\colon K \to K$ by mapping the edge E to the ordered collection of edges with labels

$$[a^{r(i)}a_1a_2][a_1a_2a_3]\cdots[a_{L-2}a_{L-1}a_L][a_{L-1}a_La^{l(k)}]$$

in an orientation preserving way and at normalised speed. This map is well defined and continuous, hence an induced map $\tilde{\phi}^*\colon H^1(K) \to H^1(K)$ on cohomology exists. We use the following result [27] (Or [4] if K is replaced with the original definition of the AP complex).

Proposition 28.

$$\check{H}^1(\Omega_\phi) \cong \varinjlim(H^1(K), \tilde{\phi}^*)$$

Let $R = \mathrm{rk}H^1(K)$, the rank of the cohomology of K. Grout finds an explicit generating set of cocycles for the cohomology of K and then outputs the induced map $\tilde{\phi}^*$ as the associated $R \times R$-matrix M_{AP}, which should be interpreted as acting on $\mathbb{Z}^R$ with respect to this generating set. So $\check{H}^1(\Omega_\phi) \cong \varinjlim M_{AP}$.

The algorithm for this computation begins by constructing the boundary matrix for the AP-complex. To do this we take all of the admitted three letter words abc and we use the convention that the boundary of this edge is $bc - ab$. Using this, we construct the associated $m \times n$ boundary matrix B where m is the number of two letter words and n is the number of three letter words. We then use standard methods from linear algebra to find a maximal set of linearly independent n dimensional vectors g such that $Bg = 0$, searching over the set of all vectors of 0s and 1s. By construction, this set generates the kernel of the boundary map inside the simplicial 1-chain group of K. This gives us a generating set of cycle vectors for the first homology of K.

We then apply the collared substitution to each of these generating vectors, giving us a new set of image vectors. Using Gaussian elimination, we find the coordinates of these image vectors in terms of the generating vectors. This induced map on homology can be represented as a square matrix. The transpose of this matrix M_{AP} then represents the induced map on cohomology, and M_{AP} is the output for the cohomology calculation via the Anderson-Putnam method. It should be noted that this algorithm is not efficient in the case where the substitution has many three letter words, as the dimension m of the 1-chain complex is the dominant limiting factor when finding linearly independent generating cycles. The time complexity increases exponentially with respect to m.

3.3 Via Properisation

For this method of computation we make use of a technique involving return words, as outlined in work of Durand, Host & Skau [21], for replacing a primitive substitution with an equivalent *pre-left proper* primitive substitution. One may then use the fact that if ϕ is a recognisable pre-left proper primitive substitution, then $\check{H}^1(\Omega_\phi) \cong \varinjlim M_\phi^T$ [37].

We begin by defining what it means to be proper.

Definition 29.

A substitution is *left proper* if there exists a letter $a \in \mathcal{A}$ such that the leftmost letter of $\phi(b)$ is a for all $b \in \mathcal{A}$. That is, $\phi(b) = aw_b$ for some $w_b \in \mathcal{A}^+$.

A substitution is *right proper* if there exists a letter $a \in \mathcal{A}$ such that the rightmost letter of $\phi(b)$ is a for all $b \in \mathcal{A}$. That is, $\phi(b) = w_b a$ for some $w_b \in \mathcal{A}^+$.

A substitution is *fully proper*[1] if it is both left and right proper.

A substitution is *pre-left proper* if some power of the substitution is left proper. Similarly for *pre-right proper* and *pre-fully proper*.

The following algorithm produces what we call the *pre-left properisation* of a substitution, so-called because there exists a finite power of the new substitution which is left proper. As usual, k is the one from the definition of the fixed letter f.

Note that if v is a return word to the fixed letter a, then $va \in \mathcal{L}$ and so $\phi^k(va)$ must also be admitted by ϕ. But $\phi^k(va) = \phi^k(v)\phi^k(a)$, and both of $\phi^k(v)$ and $\phi^k(a)$ begin with the fixed letter a, hence $\phi^k(v)$ is an exact composition of return words to a. So, if we apply ϕ^k to a return word, then the result is a composition of return words. We denote by ψ this newly constructed substitution rule on the new alphabet $\mathcal{R}$ of return words.

Algorithm 7 Pre-left properisation

```
1: function preprop
2:     rwords = returnwords(f)
3:     ψ = empty substitution with alphabet size being the cardinality of rwords
4:     for all words w in rwords do {
5:         temp = φ^k(w)
6:         decomposition = decompose temp into return words w_{i_1} ... w_{i_m}
7:         ψ(w) = decomposition
8:     }
9:     output ψ
```

This algorithm gives us a new substitution on possibly more letters than with what we began, and we may take a power of this substitution to get a left proper one. That such a power exists is clear. Indeed, every return word $v \in \mathcal{R}$ begins with the fixed letter a and, by primitivity, $\phi^i(a)$ contains at least two copies of the letter a for large enough i, so $\phi^i(a) = v_0 a u$ for some return word $v_0 \in \mathcal{R}$ and some other word u. But then $\psi^i(v)$ must begin with v_0. It follows that ψ^i is a left-proper substitution with leftmost letter v_0.

We may also form an equivalent fully proper substitution on $\mathcal{R}$ by composing ψ^i with its *right conjugate*. The right conjugate $\phi^{(R)}$ of a left proper substitution ϕ is given by setting $\phi^{(R)}(b) = w_b a$ where a is the fixed letter such that $\phi(b) = a w_b$ for all $b \in \mathcal{A}$. The right conjugate is a right proper substitution, and the composition of a left proper and right proper substitution is both left and right proper, hence fully proper. It is easy to show [23] that $X_{\phi\circ\phi^{(R)}}$ and X_ϕ are topologically conjugate subshifts. In fact, a word is admitted by $\phi \circ \phi^{(R)}$ if and only if it is admitted by ϕ, so $X_{\phi\circ\phi^{(R)}}$ and X_ϕ are equal. Hence, $\check{H}^1(\Omega_{\phi\circ\phi^{(R)}}) \cong \check{H}^1(\Omega_\phi)$.

We make use of the following which has been paraphrased from results appearing in the work of Durand, Host and Skau [21].

[1] We will often abbreviate this to just *proper*.

Proposition 30. *Let ϕ be a primitive substitution on $\mathcal{A}$ and let ψ be the pre-left properisation of ψ. The tiling space Ω_ψ is homeomorphic to Ω_ϕ.*

Hence we get the corollary

Corollary 31.

$$\check{H}^1(\Omega_\psi) \cong \check{H}^1(\Omega_\phi)$$

As ψ^i is left proper, $\check{H}^1(\Omega_{\psi^i}) \cong \varinjlim M^T_{\psi^i} \cong \varinjlim (M^i_\psi)^T \cong \varinjlim M^T_\psi$. Hence $\check{H}^1(\Omega_\phi) \cong \varinjlim M^T_\psi$.

Grout outputs the pre-left properisation ψ, the left properisation ψ^i, the full properisation $\psi^i \circ (\psi^i)^{(R)}$, and owing to the above, Grout also outputs the matrix M^T_ψ in the cohomology section.

Pre Left Properisation :	Left Properisation :	Full Properisation :
a ↦ b b ↦ ba	a ↦ b b ↦ ba	a ↦ ba b ↦ bba

Figure 8: The properisation results of the Fibonacci substitution

Remark 32. If the substitution is already proper, the properisation algorithm may return a different proper version of this substitution. This may seem like it is a feature which has no use, but by iterating this process, we find that the sequence of substitutions is eventually periodic, first proved by Durand [22]. It was therefore decided to leave this feature intact, in order to study such sequences of properisations.

Barge-Diamond :	Properisation :	Anderson-Putnam :
lim M^T	0 1 1 1	0 1 1 1

Cohomology Rank : 2

Figure 9: The cohomology results of the Fibonacci substitution

4 Pisot substitutions

In this section, we direct our attention to a special class of substitutions known as *Pisot substitutions*. This class of substitutions has received considerable attention in the literature. For an overview of the various flavours of Pisot conjectures and a mostly up-to-date exposition of current knowledge, we refer the reader to the review article of Akiyama, Barge, Berthé & Lee [2].

Definition 33. Let ϕ be a primitive aperiodic substitution on an alphabet $\mathcal{A}$ with l letters and let M_ϕ be the associated substitution matrix with characteristic polynomial p_ϕ and PF-eigenvector λ_{PF}. We say that ϕ is a *Pisot substitution* if for every root $\lambda \neq \lambda_{PF}$ of the minimal polynomial of λ_{PF}, we have $|\lambda| < 1$. If we further have that p_ϕ is an irreducible polynomial over the rationals (equivalently over the integers) then we say that ϕ is an *irreducible Pisot substitution*. Equivalently, ϕ is irreducible Pisot if every eigenvalue $\lambda \neq \lambda_{PF}$ of M_ϕ satisfies $0 < |\lambda| < 1$.

Note that irreducible Pisot substitutions are necessarily primitive and recognisable.

Example 34. Some of the most well-known substitutions are irreducible Pisot:

- The Fibonacci substitution from Example 2 has eigenvalues $\lambda_{PF} = \tau$ and $\lambda_1 = \tau - 1$ where $\tau = (1 + \sqrt{5})/2$ is the golden mean
- The *silver mean substitution* $\phi\colon 0 \mapsto 001, 1 \mapsto 0$ with fixed point A171588 A049472

  $$0010010001001000010010010001\ldots$$

  has eigenvalues $\lambda_{PF} = 1 + \sqrt{2}$ and $\lambda_1 = 1 - \sqrt{2}$
- [36] The *Tribonacci substitution* $\phi\colon 0 \mapsto 01, 1 \mapsto 02, 2 \mapsto 0$ with fixed point A092782

  $$0102010010201010201001020102\ldots$$

  has eigenvalues $\lambda_{PF} \simeq 1.83929, \lambda_{1,2} \simeq -0.419643 \pm 0.606291i$
- [39] The *flipped Tribonacci substitution* $\phi\colon 1 \mapsto 12, 2 \mapsto 31, 3 \mapsto 1$ with fixed point A100619

  $$1231112121231123112311121 23\ldots$$

  has the same eigenvalues as the Tribonacci substitution as they have equal substitution matrices.

Example 35. The Thue-Morse substitution from Example 3 is a Pisot substitution as it has PF-eigenvalue $\lambda_{PF} = 2$ whose minimal polynomial is just $\lambda - 2$ and so is trivially Pisot. However, this is not an irreducible Pisot substitution because $p_\phi = \lambda(\lambda - 2)$ is reducible. Equivalently, because 1 is also an eigenvalue of M_ϕ, the substitution cannot be irreducible Pisot.

Example 36. The substitution $\phi\colon 0 \mapsto 001111, 1 \mapsto 001$ is not Pisot as the characteristic polynomial of M_ϕ is $p_\phi = \lambda^2 - 3\lambda - 6$ which is irreducible with both roots of modulus greater than 1.

The Tribonacci substitution and the flipped Tribonacci substitution have very different dynamical properties, even though they appear extremely similar in their presentations and share eigenvalues [39].

An important tool for studying primitive aperiodic symbolic substitutions is the associated dynamical spectrum of both its subshift and tiling space. A good reference text for such notions is the book by Baake and Grimm [5]. The notion of the dynamical spectrum for a compact measure dynamical system is beyond the scope of this work, but there is a simple equivalent combinatorial definition for irreducible Pisot substitutions [11].

Remark 37. We note that, due to results of Clark-Sadun [18], the subshift of a Pisot substitution has pure discrete spectrum if and only if its corresponding tiling space (with possibly non-unital tile-lengths) has pure discrete spectrum. With this in mind, we only mention the dynamical spectrum of the subshift, confident that everything also applies to the tiling space.

Let $u^{ab} \in \mathbb{Z}^l$ denote the abelianisation of the word u. That is, the ith coordinate of the vector u^{ab} is the number of times the letter a_i appears in the word u. We say that the pair of words (u, u') are a *balanced pair* if $u^{ab} = u'^{ab}$. If u and u' cannot be factored as proper subwords $u = u_1u_2$ and $u' = u'_1u'_2$ such that (u_1, u'_1) and (u_2, u'_2) are balanced pairs, then we say that (u, u') is an irreducible balanced pair. For every $a \in \mathcal{A}$, we call the balanced pair (a, a) a *coincidence*. For pairs (u_1, u'_1) and (u_2, u'_2) we write $(u_1, u'_1)(u_2, u'_2) = (u_1u_2, u'_1u'_2)$ and call this *concatenation of pairs*. We say that (u_1, u'_1) and (u_2, u'_2) are *factors* of $(u_1u_2, u'_1u'_2)$. Clearly, every balanced pair can be factored uniquely into irreducible balanced pairs and if (u, u') is a balanced pair, then $(\phi(u), \phi(u'))$ is a balanced pair.

For a balanced pair (u, u'), let

$$I(u, u') = \{(u_i, u'_i) \mid \exists n \geq 0, \text{ such that } (u_i, u'_i) \text{ appears as an irreducible balanced pair factor of } (\phi^n(u), \phi^n(u'))\}$$

Definition 38. Let ϕ be a primitive aperiodic substitution on an alphabet $\mathcal{A}$ with l letters. If $I(u, u')$ is finite, we say that the balanced pair algorithm for ϕ *terminates* on (u, u'). If $I(u, u')$ contains a coincidence, then we say that the balanced pair algorithm for ϕ *terminates with coincidence* on (u, u').

The following theorem [11] gives a simple criterion in terms of the balanced pair algorithm for determining if the subshift of an irreducible Pisot substitution has pure discrete spectrum.

Theorem 39. *If ϕ is an irreducible Pisot substitution, then for any distinct pair of letters $a, a' \in \mathcal{A}$ the subshift X_ϕ for ϕ has pure discrete spectrum if and only if ϕ terminates with coincidence on the balanced pair $(aa', a'a)$.*

The following notion of strong coincidence is different from the balanced pair algorithm terminating with coincidence, though they are intimately related (and possibly equivalent) [9].

Definition 40. Let ϕ be a primitive aperiodic substitution on an alphabet $\mathcal{A}$ with l letters. For a distinct pair of letters $a, a' \in \mathcal{A}$, if there exists an $n \geq 1$ and a letter $b \in \mathcal{A}$ such that $\phi^n(a) = ubv$, $\phi^n(a') = u'bv'$ and $u^{ab} = u'^{ab}$ then we say that ϕ has a *coincidence* for the pair a, a'. If ϕ has a coincidence for every pair of letters $a, a' \in \mathcal{A}$ then we say that ϕ is *strongly coincident*.

Solomyak showed that if the subshift of a primitive substitution has pure discrete spectrum, then the substitution must be Pisot [41]. We are concerned with the converse of this statement and a close variant concerning strong coincidence. Although the first conjecture in the following is associated with the name *Pisot*, the conjecture is often referred to as the *Pisot substitution conjecture* to distinguish it from conjectures in other areas adopting the same name.

Conjecture 41 (Pisot conjecture)**.** *The subshift of every irreducible Pisot substitution has pure discrete spectrum.*

Conjecture 42 (Strong coincidence conjecture)**.** *Every irreducible Pisot substitution is strongly coincident.*

It is not known in general if the subshift of every strongly coincident substitution has pure discrete spectrum. If this is the case, then the Pisot conjecture is equivalent to the strong coincidence conjecture.

Other than in specific cases of single substitutions, there are relatively few families of substitutions for which we know that the Pisot conjecture holds: Hollander and Solomyak showed that for two-letter Pisot substitutions, the strong coincidence conjecture is equivalent to the Pisot conjecture [30]; Barge and Diamond had previously shown that the strong coincidence conjecture was true for two-letter substitutions and hence that the Pisot conjecture holds for all substitutions on two-letters [9]; Barge showed that the Pisot conjecture holds for substitutions of β-type [6]; Akiyama, Gähler and Lee showed via an exhaustive search that the Pisot conjecture holds for all three-letter substitutions whose incidence matrix has trace at most 2 [3]; Berthé, Jolivet, and Siegel showed that the Pisot conjecture holds for substitutions of Arnoux-Rauzy type [14] (See also [11]); Barge showed that the Pisot conjecture holds for all substitutions that are left proper and right bijective (or vice-versa) [7]; Barge, Bruin, Jones and Sadun showed that a naïve homological analogue of the Pisot conjecture admits counterexamples for all algebraic degrees, but that the coincidence rank conjecture (a variant of the homological Pisot conjecture relating the norm of the PF-eigenvalue and the *coincidence rank* of the substitution) holds for any substitution whose PF-eigenvalue has algebraic degree 1 [8].

4.1 Computational Search for Counterexamples of Conjecture 42

Perhaps the largest computer search for counterexamples to either conjecture in the literature appears in work of Akiyama-Gähler-Lee [3] where all 446,683 irreducible Pisot substitutions

on 3 letters whose incidence matrix has trace at most 2 were checked for counterexamples to the Pisot conjecture. We instead choose to focus our search for counterexamples to the strong coincidence conjecture. One can use the algorithms discussed in the previous sections of this paper to do a large scale search for counterexamples to the strong coincidence conjecture. The only methods which we have not discussed here is how to determine if the substitution is irreducible, but the concept of checking irreducibility over $\mathbb{Z}$ is a well studied one (see for example [33]). For our search we limit to substitutions over 3 or 4 letters, where irreducibility can be checked in a combinatorial fashion instead of full implementation of the irreducibility algorithms.

A subtlety encountered while implementing the check involved a deterministic check for when eigenvalues which have modulus exactly 1 or are instead very close but strictly within the unit disc. This problem was overcome by implementing an algorithm that was brought to our attention on Stack Exchange [2].

Algorithm 8 Set-up for Checking for Counterexamples to the Strong Coincidence Conjecture

```
S = generated primitive substitution
C = characteristic polynomial of substitution matrix of S
if C is irreducible over ℤ do {
    Evals = Eigenvalues of the substitution matrix of S
    Evals = sort(absolute value of Evals)
    if Evals[1] < 1 do {
        Check strong coincidence of S
        if strong coincidence algorithm does not halt
            Counterexample found
    }
}
```

We place an ordering on substitutions over the same alphabet (modulo the trivial equivalences given by reversing and letter permutations) via the lexicographical ordering of the words $(\phi(a^1), \phi(a^2), \ldots, \phi(a^n))$. For example, $(a, a, a) \prec (a, a, b) \prec (a, c, a) \prec (aa, a, a)$. We then methodically generate the substitutions in this list.

Results 43 (3 letters)**.** *The first 321,425,442 three letter substitutions were checked, of which 12,573,955 were irreducible Pisot. All of these substitutions passed the strong coincidence check, therefore no counterexample to the strong coincidence conjecture was found. During these checks, the number of iterations of the substitution that it took for a strong coincidence to be found was recorded. This is summarised in Table 1.*

The substitutions that took 10 iterations to form their first coincidence for all pairs of

[2] How to test if a primitive matrix has an eigenvalue of unit modulus—http://math.stackexchange.com/q/1440992/29059

letter were

$$\begin{array}{llllll} a \mapsto & cabb & b \mapsto & ac & c \mapsto & a \\ a \mapsto & cbcbab & b \mapsto & a & c \mapsto & b. \end{array}$$

Table 1: Number of iterations to produce a strong coincidence

Number of Iterations	**Number of Substitutions**
1	1,921,144
2	5,778,331
3	3,777,324
4	984,669
5	105,325
6	6,608
7	512
8	37
9	3
10	2
11+	0

Results 44 (4 letters)**.** *The first 200,399 four letter substitutions were checked, of which 15,001 were irreducible Pisot. All of these substitutions passed the strong coincidence check, therefore no counterexample to the strong coincidence conjecture was found.*

5 Acknowledgements

The authors would like to thank Fabien Durand for his helpful advice in piecing together a robust recognisability algorithm, and Etienne Pillin for helpful comments with regards to the coding. We would also like to thank Alex Clark, Greg Maloney and Jamie Walton for helpful suggestions during the development of Grout and for testing early versions of the program and Marcy Barge for suggestions and discussions relating to the section on Pisot substitutions.

References

[1] Qt version 5.4.1, 2015. http://www.qt.io.

[2] S. Akiyama, M. Barge, V. Berthé, J.-Y. Lee, and A. Siegel. On the Pisot Substitution Conjecture. In *Mathematics of aperiodic order*, volume 309 of *Progr. Math.*, pages 33–72. Birkhäuser/Springer, Basel, 2015.

[3] Shigeki Akiyama, Franz Gähler, and Jeong-Yup Lee. Determining pure discrete spectrum for some self-affine tilings. *Discrete Math. Theor. Comput. Sci.*, 16(3):305–316, 2014.

[4] Jared E. Anderson and Ian F. Putnam. Topological invariants for substitution tilings and their associated C^*-algebras. *Ergodic Theory Dynam. Systems*, 18(3):509–537, 1998.

[5] Michael Baake and Uwe Grimm. *Aperiodic order. Vol. 1*, volume 149 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2013. A mathematical invitation, With a foreword by Roger Penrose.

[6] M. Barge. The Pisot Conjecture for β-substitutions. *Preprint*, arXiv:1505.05508, 2015.

[7] Marcy Barge. Pure discrete spectrum for a class of one-dimensional substitution tiling systems. *Discrete Contin. Dyn. Syst.*, 36(3):1159–1173, 2016.

[8] Marcy Barge, Henk Bruin, Leslie Jones, and Lorenzo Sadun. Homological Pisot substitutions and exact regularity. *Israel J. Math.*, 188:281–300, 2012.

[9] Marcy Barge and Beverly Diamond. Coincidence for substitutions of pisot type. *Bulletin de la Société mathématique de France*, 130(4):619–626, 2002.

[10] Marcy Barge and Beverly Diamond. Cohomology in one-dimensional substitution tiling spaces. *Proc. Amer. Math. Soc.*, 136(6):2183–2191, 2008.

[11] Marcy Barge, Sonja Štimac, and R. F. Williams. Pure discrete spectrum in substitution tiling spaces. *Discrete Contin. Dyn. Syst.*, 33(2):579–597, 2013.

[12] J. Bellissard, D. J. L. Herrmann, and M. Zarrouati. Hulls of aperiodic solids and gap labeling theorems. In *Directions in mathematical quasicrystals*, volume 13 of *CRM Monogr. Ser.*, pages 207–258. Amer. Math. Soc., Providence, RI, 2000.

[13] R. Berger. *The Undecidability of the Domino Problem*. Memoirs ; No 1/66. American Mathematical Society, 1966.

[14] Valérie Berthé, Timo Jolivet, and Anne Siegel. Substitutive Arnoux-Rauzy sequences have pure discrete spectrum. *Unif. Distrib. Theory*, 7(1):173–197, 2012.

[15] Valérie Berthé and Anne Siegel. Tilings associated with beta-numeration and substitutions. *Integers*, 5(3):A2, 46, 2005.

[16] R. Bott and L.W. Tu. *Differential Forms in Algebraic Topology*. Graduate Texts in Mathematics. Springer New York, 1982.

[17] Alex Clark and John Hunton. Tiling spaces, codimension one attractors and shape. *New York J. Math.*, 18:765–796, 2012.

[18] Alex Clark and Lorenzo Sadun. When shape matters: deformations of tiling spaces. *Ergodic Theory Dynam. Systems*, 26(1):69–86, 2006.

[19] Karel Culik, II. The decidability of v-local catenativity and of other properties of D0L systems. *Information Processing Lett.*, 7(1):33–35, 1978.

[20] David Damanik and Daniel Lenz. Substitution dynamical systems: characterization of linear repetitivity and applications. *J. Math. Anal. Appl.*, 321(2):766–780, 2006.

[21] F. Durand, B. Host, and C. Skau. Substitutional dynamical systems, bratteli diagrams and dimension groups. *Ergodic Theory and Dynamical Systems*, 19:953–993, 8 1999.

[22] Fabien Durand. A characterization of substitutive sequences using return words. *Discrete Math.*, 179(1-3):89–101, 1998.

[23] Fabien Durand and Julien Leroy. S-adic conjecture and bratteli diagrams. *Comptes Rendus Mathematique*, 350(21-22):979 – 983, 2012.

[24] A. Ehrenfeucht and G. Rozenberg. On simplifications of PDOL systems. In *Proceedings of a Conference on Theoretical Computer Science (Univ. Waterloo, Waterloo, Ont., 1977)*, pages 81–87. Comput. Sci. Dept., Univ. Waterloo, Waterloo, Ont., 1978.

[25] Sébastien Ferenczi. Rank and symbolic complexity. *Ergodic Theory Dynam. Systems*, 16(4):663–682, 1996.

[26] Sébastien Ferenczi. Complexity of sequences and dynamical systems. *Discrete Math.*, 206(1-3):145–154, 1999. Combinatorics and number theory (Tiruchirappalli, 1996).

[27] Franz Gähler and Gregory R. Maloney. Cohomology of one-dimensional mixed substitution tiling spaces. *Topology Appl.*, 160(5):703–719, 2013.

[28] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 1996.

[29] Tero Harju and Matti Linna. On the periodicity of morphisms on free monoids. *Informatique théorique et applications*, 20(1):47–54, 1986.

[30] Michael Hollander and Boris Solomyak. Two-symbol Pisot substitutions have pure discrete spectrum. *Ergodic Theory Dynam. Systems*, 23(2):533–540, 2003.

[31] A. Julien. Complexity as a homeomorphism invariant for tiling spaces. *Preprint*, arXiv:1212.1320, 2012.

[32] A. Lagae. *Wang Tiles in Computer Graphics*. Synthesis Lectures on Computer Graphics and Animation. Morgan & Claypool Publishers, 2009.

[33] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534.

[34] Brigitte Mossé. Puissances de mots et reconnaissabilité des points fixes d'une substitution. *Theoret. Comput. Sci.*, 99(2):327–334, 1992.

[35] R. Penrose. Pentaplexity a class of non-periodic tilings of the plane. *The Mathematical Intelligencer*, 2(1):32–37, 1979.

[36] G. Rauzy. Nombres algébriques et substitutions. *Bull. Soc. Math. France*, 110(2):147–178, 1982.

[37] Lorenzo Sadun. *Topology of tiling spaces*, volume 46 of *University Lecture Series*. American Mathematical Society, Providence, RI, 2008.

[38] D. Shechtman, I. Blech, D. Gratias, and J. W. Cahn. Metallic phase with long-range orientational order and no translational symmetry. *Phys. Rev. Lett.*, 53:1951–1953, Nov 1984.

[39] V. F. Sirvent. Semigroups and the self-similar structure of the flipped Tribonacci substitution. *Appl. Math. Lett.*, 12(1):25–29, 1999.

[40] N. J. A Sloane. The on-line encyclopedia of integer sequences. http://oeis.org.

[41] Boris Solomyak. Dynamics of self-similar tilings. *Ergodic Theory Dynam. Systems*, 17(3):695–738, 1997.

[42] R. Twarock. A tiling approach to virus capsid assembly explaining a structural puzzle in virology. *Journal of Theoretical Biology*, 226(4):477–482, 2004.

2010 *Mathematics Subject Classification*: Primary 37B10, 68R15; Secondary 52C23, 54H20, 55N05.
Keywords: Tiling spaces, Symbolic dynamics, Čech cohomology, Pisot, Substitutions.

(Concerned with sequences A001285, A003842, A003849, A010059, A010060, A014577, A014709, A014710, A020985, A035263, A049320, A049321, A049472, A073057, A092782, A096268, A096270, A100260, A100619, A106665, A171588, and A275202.)
