The Tiling Problem

Nikhil Gopalkrishnan

December 08, 2008

1 Introduction

A *Wang tile* [12] is a unit square with each edge *colored* from a finite set of colors Σ. A set S of Wang tiles is said to *tile* a planar grid $\mathbb{Z}^2$ if copies of tiles from S can be placed, one at each grid position, such that abutting edges of adjacent tiles have the same color. Multiple copies of any tile may be used, with no restriction on the number. If we allow the tiles to be rotated or reflected, any single Wang tile can tile the plane by itself. The question of whether such a tiling exists for a given set of tiles is interesting only in the case where we do not allow rotation or reflection, thus holding tile orientation fixed. This decision problem, called the *tiling* or *domino* problem, was first posed in 1961 by Wang in a seminal paper. He also discussed the relation of this problem to the decision problem for certain classes of formulae of predicate calculus arising in automated theorem proving. Wang incorrectly conjectured that every tile set that tiles the plane permits a *periodic* tiling, that is, has a translational symmetry. Based on this assumption, he gave a general procedure for deciding the tiling problem. His assumption was disproved in 1966 by Berger [3] who constructed a tile set that allowed only an aperiodic tiling. He used this tile set to show that the tiling problem is undecidable. Berger's tile set was quite large, over twenty thousand tiles, and his proof quite involved. Robinson [10] reduced the number of tiles to just over fifty and gave a much simpler proof of undecidability. Previous to Berger's result, Wang himself showed a restricted version of the tiling problem, where only a certain tile was allowed at the origin, to be undecidable by reducing the halting problem to it. This and later work in tilings gave a method for simulating Turing machines using tiles, paving the way for thinking of tiles as a model of Turing universal computation.

Interest in tilings has renewed in recent years due to two unrelated developments. Firstly, rapid advancements in theoretical and experimental DNA self-assembly allow us to construct nanoscale physical approximations to Wang tiles that can be programmed to tile according to simple rules. This allows us to perform computation in the test-tube, with advantages like massive parallelism and energy efficiency over traditional circuit based silicon machines. Attempts to model and study self-assembly via tilings was introduced by Winfree [14, 13] who extended Wang tilings by adding a mechanism for modeling growth. Secondly, several authors [7, 8, 6] have recently been interested in providing simpler proofs for the undecidability of the tiling problem using a technique called two-by-two substitution systems. These proofs provide a deeper intuition into how aperiodic tile sets occur and how one might be designed.

In this paper we will give an overview of the tiling problem and its variants. We will look at the traditional approach to proving its undecidability as introduced by Wang, Berger, Robinson etc. We will also look at the newer approaches using two-by-two substitution systems. We also look at certain extensions to the tiling problem, particularly in self-assembly and give an overview of the results obtained.

2 Variants of the Tiling Problem

We saw that the tiling problem is interesting only in the case where rotation and reflection are disallowed. Consider the following closely related decision problem, the *complementary tiling problem*, where as before our task is to tile the plane with unit sized squares. The abutting edges of adjacent tiles must now have *complementary* colors, where the set of complementary colors is given. In the absence of rotation or reflection, this problem is equivalent to the conventional tiling problem (which we shall refer to as the matching tiling problem) and hence undecidable. Indeed, every question about complementary tilings can be converted to one about matching tilings by replacing each pair of complementary colors by a single new color. Also, every question about matching tilings is modified by replacing each color by a pair of new complementary colors. This equivalence doesn't hold when rotation is allowed. The complementary tiling problem is undecidable even in the presence of rotations, while the matching tiling problem is not, as shown earlier. We can reduce the matching tiling problem to the complementary tiling problem in the presence of rotations in the following manner. Since the matching problem in the absence of rotations doesn't have any interactions between an horizontal edge and a vertical edge, we can rename the vertical edges so that the horizontal and vertical edges have no color in common. Replacing each color by a pair of complementary colors gives us a question in complementary tilings with rotation allowed. However, due to our renaming, rotations do not give rise to any new tile interactions, and hence a complementary tiling exists iff a matching tiling exists.

The *notched tiling problem* is a close relative of the complementary tiling problem. The edges of our tiles are notched with dents and bumps that *fit* with each other. A set of tiles tile the plane if they can be arranged such that abutting notched edges of adjacent tiles fit together. In rotations are allowed but reflections are not, the notched tiling problem is equivalent to the complementary tiling problem, and hence undecidable. However, in the presence of reflections the two problems are not equivalent as asymmetric notches are a distinguishing feature. The notched tiling problem is undecidable even in the presence of rotations and reflections. The proof is a similar reduction as before, we introduce notches such that rotational and reflective symmetries are broken, thus rendering these operations useless. A similar slightly more involved

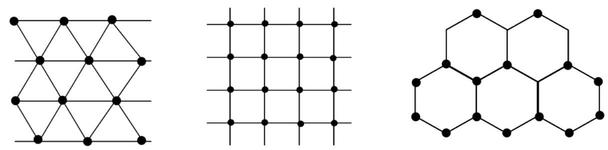

Figure 1: The three regular tilings

argument shows that the complementary tiling problem is also undecidable in the presence of reflections in addition to rotations.

3 The Three Regular Tilings

A regular tiling is a tiling of the plane by regular equal sized polygons. Wang tilings are an example of regular tilings where the regular polygons are squares. What other regular polygons permit tilings that completely cover the plane, leaving no holes? Not many, as we show below.

We will borrow Coxeter's [5] notation for representing regular tilings: $\{p, q\}$ is a regular tiling by regular p-gons with q of the p-gons meeting at each vertex. Thus, the internal angle of each p-gon is $(1-2/p)\pi$, and q of them meet at each vertex such that angles sum to 2π. This gives the equation $(p-2)(q-2) = 4$, whose solutions are $\{4, 4\}, \{3, 6\}$ and $\{6, 3\}$. $\{4, 4\}$ corresponds to the familiar tilings via squares. $\{3, 6\}$ is a tiling using equilateral triangles, 6 of which meet at every vertex while $\{6, 3\}$ is a tiling using regular hexagons, 3 meeting at each vertex (Fig. 1).

Tilings can also be thought of as infinite planar graphs embedded in the plane, with vertices and edges of the tiling as vertices and edges of the graph. We can take the dual of this graph. Clearly, $\{6, 3\}$ and $\{3, 6\}$ are duals of each other when the duals are drawn appropriately by taking vertices of the dual at the center of the faces of the tiling. $\{4, 4\}$ is its own dual, obtained by rotating each edge of the tiling in the plane orthogonally about its center. Also, if we think of $\{3, 6\}$ as a Delaunay triangulation, its dual $\{6, 3\}$ is the Voronoi diagram, as expected.

4 Simulating Turing Machines Using Tiles

Simulating Turing machines using tiles is an integral part of proofs of undecidability for various tiling problems. These simulations are also interesting in their own right, paving the way for tiles to be thought of as models of computation. We will simulate the working of any arbitrary Turing machine started on a blank tape and use it to show undecidability of a restricted version of the tiling problem, the origin-constrained tiling problem, where a particular tile is forced

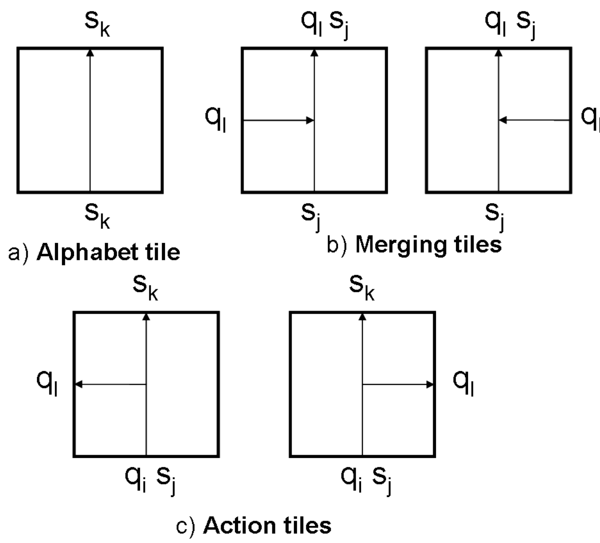

Figure 2: Simulating a Turing machine

to occur at the origin in any tiling of the plane. This was first done by Wang [12] and later improved by Berger [3] and Robinson [10]. We will also show how to simulate a universal Turing machine started on an arbitrary tape using only a fixed number of tiles. This will be used to show the undecidability of the completion problem: given a finite portion of the tiling can it be completed to cover the plane? This problem was posed and solved in [10].

4.1 Simulation of an Arbitrary Turing Machine Started on a Blank Tape

Given an arbitrary Turing maching T, let $q_0, q_1, \ldots$ be the finite number of states with q_0 as the start state and $s_0, s_1, \ldots$ be the finite number of tape symbols with s_0 as the blank. We will assume that T has only a single infinite tape with a head that can read exactly one cell of the tape at any given time. The action of T will be indicated by tuples of the form $q_i s_j s_k L q_l$ or $q_i s_j s_k R q_l$ indicating that the symbol s_j was overwritten by s_k as the machine transitioned from q_i to q_l moving either left (L) or right (R). The configuration of the machine will be encoded horizontally on the grid, corresponding to the horizontal edges of the tiles on a row. Successive configurations are encoded vertically one above the other, progressing above.

The colors on the edges will be indicated by a combination of arrows and labels making the simulation transparent to the reader. Tile edges match only if the labels match and the arrows align head to tail. The various kind of tiles used in the simulation are indicated in Fig 2. The alphabet tile simply copies a

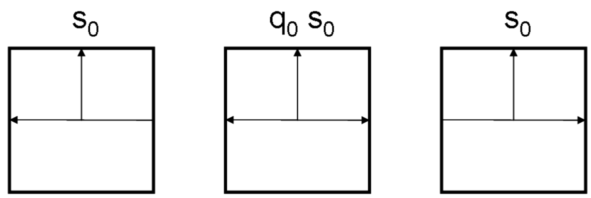

Figure 3: Starting tiles for a blank tape

symbol to the next step. The merging tiles combine a state and a symbol and are used to track the movement of the tape head. The action tiles are used to overwrite a symbol and to indicate movement of the tape head. Given a row of tiles whose upper edges encode the current configuration of the computation, exactly one edge will have an up arrow with label $q_i s_j$ and the others will have labels of the type s_k. An action tile will attach at $q_i s_j$ and overwrite s_j and pass the new state to a merging tile which will place the tape head at the correct position. The tiling at the next step will be possible iff there is a valid transition for the Turing machine. The initial configuration is represented by the tiles in Fig. 3. The tile with label $q_0 s_0$ on its top edge is placed at the origin which forces the other two tiles to repeat infinitely on either side. A dummy tile is used to completely tile the plane below this seed row of tiles.

The above construction is a reduction of the problem of deciding if an arbitrary Turing machine halts when started on a blank state to the origin constrained tiling problem. The tile set covers the plan iff the Turing machine does not halt, which is an undecidable problem. Thus, we have proved the origin constrained tiling problem undecidable.

4.2 Simulation of a Universal Turing Machine Started on an Arbitrary Tape

Given a universal Turing machine U, there is some symbol s_h such that when the machine is started on some arbitrary finite string containing s_h with its head on s_h there is no decision procedure that tells us if the machine halts. We shall simulate the run of U starting with some such string on its tape using a constant number of tiles.

We will use tiles of the form described in Fig. 2 for our universal machine U. In addition we will use the five tiles illustrated in Fig. 4. Corresponding to any initial finite string on the tape, we start with a sequence of tiles of the same length. The cell on which the head restes is represented by the center tile in Fig. 4. All other symbols on the tape are represented by their corresponding alphabet tiles from Fig. 2. The second tile in Fig. 4 is placed to the left of the leftmost alphabet tile which forces the first tile in Fig. 4 to repeat infinitely to the left. The fourth tile in Fig. 4 is placed to the right of the rightmost alphabet tile which forces the last tile to repeat infinitely to the right. A dummy tile is

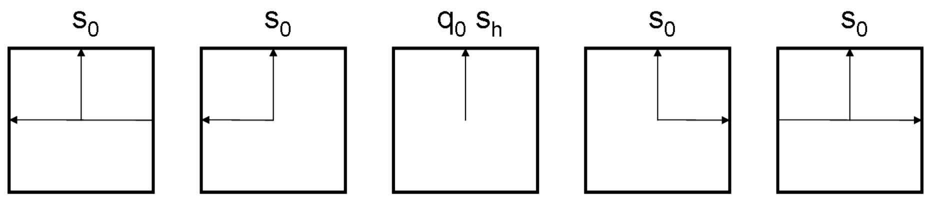

Figure 4: General starting tiles

used to completely tile the plane below this seed row of tiles. The rest of the simulation proceeds exactly like in the previous section.

The set of tiles cover the plane iff the machine U started on this string does not halt. The above construction is a reduction of a version of the halting problem for universal machines to the problem of deciding if a tiling exists starting from a finite set of tiles in a row. Thus, we have shown the tiling completion problem to be undecidable.

5 Substitution Systems

Given a set of tiles S, a tiling of the plane is a function from $\mathbb{Z}^2$ to S. Given S, the set of all tilings of the plane form a discrete topological space and $\mathbb{Z}^2$ acts on the space via translations. This is a two-dimensional symbolic dynamical system of finite type. Periodic tilings correspond to finite orbits in this system. A substitution system is a set of derivation rules on an alphabet. Mozes [7] defined two-dimensional substitution systems as generalizations of one-dimensional substitution systems. He studied the dynamical systems that arise from these substitution systems. For a certain class of two-dimensional substitution systems he showed how one might construct tiling systems that have the same dynamical system (upto isomorphism) as the one arising from the substitution system. This paved the way for showing a dynamical system that does not have any finite orbits. Thus, an aperiodic tile set is constructed, giving a new proof of the tiling problem.

Alternative proofs using substitution systems were given more recently in [6, 8]. The basic framework is the same as before. Define certain class of substitution systems, show that their dynamical systems correspond to tilings. Find interesting dynamical systems through these substitution methods. In particular, look for dynamical systems that have no finite orbit. This gives an aperiodic tile set which is used to show the undecidability of the tiling problem. This is a more structured and algebraic approach and provides a certain intuition of where aperiodic tilings come from, which is not apparent from classical work of Robinson, Berger etc. It also shows that even aperiodic tilings are not completely random. They are actually quite regular and avoid translational symmetry in a very regular and deliberate manner. There has also been some work related to higher dimensional substitution systems [9] which look at tilings in $\mathbb{Z}^n$. The classical methods for producing aperiodic tilings fail here because the geometry cannot be visualized anymore. However, substitution systems work via ideas analogous to ones in lower dimensional susbtitution systems.

6 Self-assembly and Tilings

Winfree [13] used tilings to model certain simple self-assembly processes. Wang tilings by themselves are inadequate to capture the complexity of self-assembly processes as the lack a growth mechanism. Winfree introduced the idea of growth and co-ordinated binding to give a Tile Assembly Model that attempts to model simple self-assembly processes. He asked what simple shapes can be formed in such a mode. In particular, he tried to minimize the number of different tile types required to form a square of size $N \times N$. The minimum number of tiles to form a shape corresponds to the program size complexity of Turing machines. Winfree and Rothemund [11] showed how to form a $N \times N$ size square using only $\Theta(\log N)$ type of tiles. They proved that the minimum number of tile types required to form a square is almost always $\Omega(\frac{\log N}{\log \log N})$. Adleman et al. [1] achieved this lower bound in a subsequent paper. There have also been combinatorial questions asked [2] about shapes and patterns and the smallest number of tiles requred to form these.

Recently, Reif et al. [4] extended the Tile Assembly Model by introducing randomization. They construct linear assemblies, a finite row of tiles, of expected length N using $\Theta(\log N)$ tiles as against the lower bound of N in the conventional Tile Assembly Model. They also prove that this is the optimal construction by proving a lower bound of $\Omega(\log N)$ for any tile system that gives a linear assembly of length N in expectation.

References

[1] Leonard Adleman, Qi Cheng, Ashish Goel, and Ming-Deh Huang. Running Time and Program Size for Self-Assembled Squares. In *STOC*, pages 740–748, 2001.

[2] Leonard Adleman, Qi Cheng, Ashish Goel, Ming-Deh Huang, David Kempe, Pablo Moisset de Espanés, and Paul Rothemund. Combinatorial Optimization Problems in Self-Assembly. In *STOC*, pages 23–32, 2002.

[3] Robert Berger. The Undecidability of the Domino Problem. volume 66, pages 1–72, 1966.

[4] Harish Chandran, Nikhil Gopalkrishnan, and John Reif. The tile complexity of linear assemblies. Unpublished, submitted to STOC 2009.

[5] H. S. M. Coxeter. *Regular polytopes*. Dover Publications Inc., New York, 1973.

[6] Bruno Durand, Leonid A. Levin, and Alexander Shen. Complex tilings. In *STOC*, pages 732–739, 2001.

[7] Shahar Mozes. Tilings, substitution systems and dynamical systems generated by them. *Journal d'Analyse Mathmatique*, 53(1):139–186, December 1989.

[8] Nicolas Ollinger. Two-by-two substitution systems and the undecidability of the domino problem. In *CiE*, pages 476–485, 2008.

[9] Charles Radin. Space tilings and substitutions. *Geometriae Dedicata*, 55(3):257–264, May 1995.

[10] Raphael Robinson. Undecidability and Nonperiodicity for Tilings of the Plane. In *Inventiones Mathematicae*, volume 12, pages 177–209, 1971.

[11] Paul Rothemund and Erik Winfree. The Program-Size Complexity of Self-Assembled Squares. In *STOC*, pages 459–468, 2000.

[12] Hao Wang. Proving Theorems by Pattern Recognition II. 1961.

[13] Erik Winfree. *Algorithmic Self-Assembly of DNA*. PhD thesis, Caltech, 1998.

[14] Erik Winfree, Furong Liu, Lisa Wenzler, and Nadrian Seeman. Design and Self-Assembly of Two-Dimensional DNA Crystals. *Nature*, 394:539–544, 1999.