CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Caltech Authors - Main

Complexity of Compact Proofreading for Self-Assembled Patterns

David Soloveichik and Erik Winfree

Department of CNS and CS, California Institute of Technology
{dsolov,winfree}@caltech.edu

Abstract. Fault-tolerance is a critical issue for biochemical computation. Recent theoretical work on algorithmic self-assembly has shown that error correcting tile sets are possible, and that they can achieve exponential decrease in error rates with a small increase in the number of tile types and the scale of the construction [24, 4]. Following [17], we consider the issue of applying similar schemes to achieve error correction without any increase in the scale of the assembled pattern. Using a new proofreading transformation, we show that compact proofreading can be performed for some patterns with a modest increase in the number of tile types. Other patterns appear to require an exponential number of tile types. A simple property of existing proofreading schemes – a strong kind of redundancy – is the culprit, suggesting that if general purpose compact proofreading schemes are to be found, this type of redundancy must be avoided.

1 Introduction

The Tile Assembly Model [22, 23] formalizes a generalized crystal growth process by which an organized structure can spontaneously form from simple parts. This model considers the growth of two dimensional "crystals" made out of square units called tiles. Typically, there are many types of tiles that must compete to bind to the crystal. A new tile can be added to a growing complex if it binds strongly enough. Each of the four sides of a tile has an associated bond type that interacts with matching sides of other tiles that have already been incorporated. The assembly starts from a specified seed assembly and proceeds by sequential addition of tiles. Tiles do not get used up since it is assumed there is an unbounded supply of tiles of each type. This model has been used to theoretically examine how to use self-assembly for massively parallel DNA computation [21, 26, 16, 13], for creating objects with programmable morphogenesis [10, 1, 2, 20], for patterning of components during nanofabrication of molecular electronic circuits [6], and for studying self-replication and Darwinian evolution of information-bearing crystals [18, 19]. Fig. 1 illustrates two different patterns and the corresponding tile systems that self-assemble into them. Both patterns are produced by similar tile systems using only two bond types, four tile types, simple boolean rules and similar seed assemblies (the L-shaped boundaries).

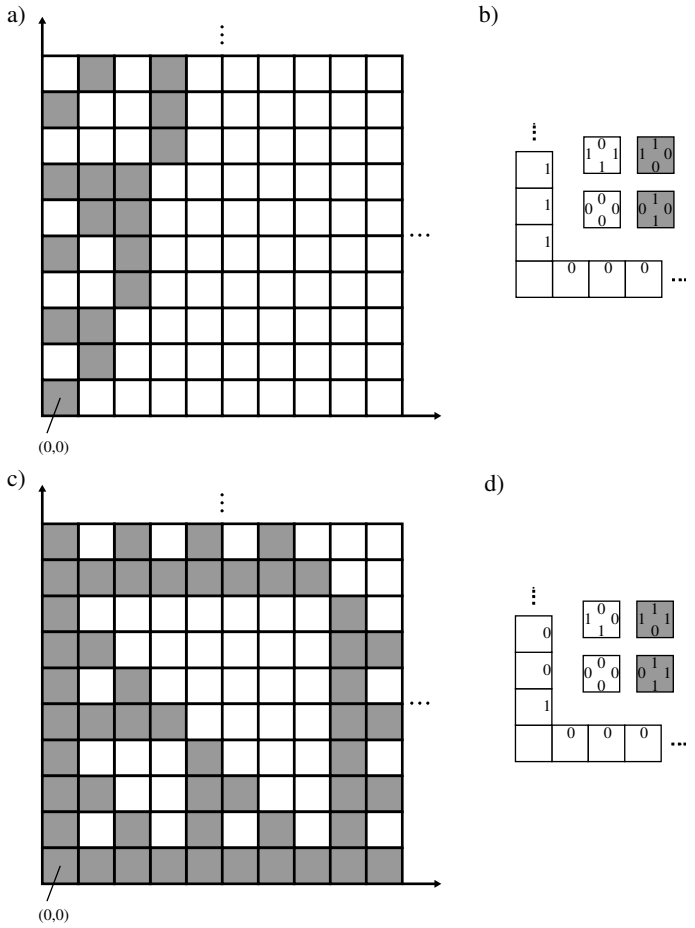

Fig. 1. (a) A binary counter pattern and (b) a tile system constructing it. (c) A Sierpinski pattern and (d) a tile system constructing it. In this formalism, identically-labeled sides match and tiles cannot be rotated. Tiles may attach to the growing assembly only if at least two sides match, i.e., if two bonds can form. Mismatches neither help nor hinder assembly. Note that the tile choice at each site is deterministic for these two tile sets.

Confirming the physical plausibility and relevance of the abstraction, several self-assembling systems have been demonstrated using DNA molecules as tiles, including both periodic [25, 15, 12] and algorithmic patterns [14, 9, 3]. A major stumbling block to making algorithmic self-assembly practical is the error rate inherent in any stochastic biochemical implementation. Current implementations seem to suffer error rates of 1% to 15% [9, 3]. This means that on average every eighth to hundredth tile that is incorporated does not correctly bond with its neighbors. Once such a mistake occurs, the erroneous information can be prop-

agated to tiles that are subsequently attached. Thus, a single mistake can result in a drastically different pattern being produced. With this error rate, structures of size larger than roughly 100 tiles cannot be assembled reliably.

There are generally two ways to improve the error-robustness of the assembly process. First, the physics of the process can be modified to achieve a lower probability of the incorporation of incorrect tiles into the growing complex. The second method, which we pursue here, is to use some logical properties of the tiles to perform error correction.

Proofreading tile sets for algorithmic self-assembly were introduced by Winfree and Bekbolatov [24]. The essential idea was to make use of a redundant encoding of information distributed across k tiles, making isolated errors impossible: to continue growth, errors must appear in multiples of k. Thanks to the reversible nature of crystallization, growth from erroneous tiles stalls and the erroneous tiles subsequently dissociate, allowing another chance for correct growth. Using this approach, a large class of tile sets can be transformed into more robust tile sets that assemble according to the same logic.

However (a) the proofreading tile sets produce assemblies k times larger than the original tile sets, involving k^2 times as many tiles; and (b) the improvement in error rates did not scale well with k in simulation. Chen and Goel [4] developed *snaked proofreading tile sets* that generalize the proofreading construction in a way that further inhibits growth on crystal facets. They were able to prove, with respect to a reversible model of algorithmic self-assembly, that error rates decrease exponentially with k, and thus to make an $N \times N$ pattern required only $k = \Omega(\log N)$. This provides a solution for (b), although the question of optimality remains open. Reif et al [17] raised the question of whether more compact proofreading schemes could be developed, and showed how to transform the two tiles sets shown in Fig. 1 to obtain lower error rates without any sacrifice in scale. However, Reif et al did not give a general construction that works for any original tile set, and did not analyze how the number of tile types would scale if the construction were to be generalized to obtain greater degrees of proofreading. Thus, question (a) concerning whether this can be improved in general and at what cost remained open.

The question of compactness is particularly important when self-assembly is used for molecular fabrication tasks, in which case the scale of the final pattern is of direct and critical importance. Furthermore, the question of scale is a fundamental issue for the theory of algorithmic self-assembly. In the error-free case, disregarding scale can drastically change the minimal number of tile types required to produce a given shape [20]; some shapes can be assembled from few tile types at a small scale, while other shapes can *only* be assembled from few tile types at a large scale. Examining whether proofreading can be performed without sacrificing scale is both of practical significance and could lead to important theoretical distinctions.

If it is the case that some patterns can't be assembled with low error rates at the original scale using a concise tile set, while for other patterns compact proofreading can be done effectively, then we would be justified in calling the for-

mer intrinsically fragile, and the latter intrinsically robust. Any such distinctions should be independent of any particular proofreading scheme. Indeed, we here show that this is true (in a certain sense), and we give a combinatorial criterion that distinguishes fragile patterns from robust patterns. As examples, we show that the two patterns discussed in Reif et al's work on compact proofreading [17] and shown in Fig. 1 are fundamentally different, in that (within a wide class of potential proofreading schemes considered here) the cost of obtaining reliable assembly at the same scale becomes dramatically different as lower error rates are required.

1.1 The Abstract Tile Assembly Model

This section informally summarizes the abstract Tile Assembly Model (aTAM). See [8, 20] for a formal treatment. Self assembly occurs on a $\mathbb{Z} \times \mathbb{Z}$ grid of unit square locations, on which unit-square **tiles** may be placed under specific conditions. Each tile has **bond types** on its north, east, south and west sides. A finite set of **tile types** defines the set of possible tiles that can be placed on the grid. Tile types are oriented and therefore a rotated version of a tile type is considered to be a different tile type. A single tile type may be used an arbitrary number of times. A **configuration** is a set of tiles such that there is at most one tile in every location $(i, j) \in \mathbb{Z} \times \mathbb{Z}$. Two adjacent tiles **bond** if their abutting sides have matching bond types. Further, each bond type forms bonds of a specific strength, called its **interaction strength**. In this paper the three possible strengths of bonds are $\{0, 1, 2\}$. A new tile can be added to an empty spot in a configuration if and only if the sum of its interaction strengths with its neighbors reaches or exceeds some parameter τ. The tile systems shown in this paper use $\tau = 2$, i.e., at least a single strong (strength 2) or two weak (strength 1) bonds are needed to secure a tile in place.

For the purposes of this paper, a tile system consists of a finite set of tile types T with specific interaction strengths associated with each bond type, and a start configuration. Whereas a configuration can be any arrangement of tiles, we are interested in the subclass of configurations that can result from a self-assembly process. Thus, an **assembly** is a configuration that can result from the start configuration by a sequence of additions of tiles according to the above rules at $\tau = 1$ or $\tau = 2$ (i.e., it is connected). A **τ-stable** assembly is one that cannot be split into two parts without breaking bonds with a total strength of at least τ. **Deterministic** tile systems are those whose assemblies can incorporate at most 1 tile type at any location at any time.

1.2 The Kinetic Tile Assembly Model and Errors

The Kinetic Tile Assembly Model (kTAM) augments the abstract Tile Assembly Model with a stochastic model of self-assembly dynamics, allowing calculation of error rates and the duration of self-assembly. Following [23, 24] we make the following assumptions. First, the concentration of each tile type in solution is held constant throughout the self-assembly process, and the concentrations of

all tile types are equal. We assume that for every tile association reaction there is a corresponding dissociation reaction (and no others). We further assume that the rate of addition (**forward rate** f) of any tile type at any position of the perimeter of the growing assembly is the same. Specifically, $f = k_f e^{-G_{mc}}$ where k_f is a constant that sets the time scale, and G_{mc} is the logarithm of the concentration of each tile type in solution. The rate that a tile falls off the growing assembly (**reverse rate** r_b) depends exponentially on the number of bonds that must be broken. Specifically, $r_b = k_f e^{-bG_{se}}$ where b is the total interaction strength with which the tile is attached to the assembly, and G_{se} is the unit bond free energy, which may depend, for example, on temperature.

We assume the following concerning f and r_b. Following [23] we let $f \approx r_2$ for a $\tau = 2$ system since it provides the optimal operating environment [23]. Further, we assume f (and therefore r_2) can be arbitrarily chosen in our model by changing G_{mc} and G_{se}, for example by changing tile concentrations and temperature. (In practice, there are limits to how much these parameters can be changed.) However, k_f is assumed to be a physical constant not under our control.

In the kTAM, the $\tau = 2$ tile addition requirement imposed by the abstract Tile Assembly Model is satisfied only with a certain probability: assuming $f \approx r_2$ so $r_1 \gg f$, if a tile is added that bonds only with strength 1, it falls off very quickly as it should in the aTAM with $\tau = 2$. Tiles attached with strength 2 stick much longer, allowing an opportunity for other tiles to attach to them. Once a tile is bonded with total strength 3, it is very unlikely to dissociate (unless surrounding tiles fall off first).

Following [4], the fundamental kind of error we consider here is an **insufficient attachment**. At threshold $\tau = 2$, an insufficient attachment occurs when a tile attaches with strength 1, but before falling off, another tile attaches next to it, resulting in a 2-stable assembly. Since insufficient attachments are the only kind of error we analyze in this paper, we'll use "error" and "insufficient attachment" interchangeably.

Chen and Goel [4] make use of a simplification of the kTAM that captures the essential behavior while being more tractable for rigorous proofs. Under the conditions where $f = r_2$, the self-assembly process is dominated by tiles being added with exactly 2 bonds and tiles falling off via exactly 2 bonds. The **locking** kTAM model assumes that these are the only possible single-tile events. That is, $r_b = 0$ for $b \geq 3$, and tiles never attach via a single strength-1 bond. Additionally, insufficient attachments are modeled in the locking kTAM as atomic events, in which two tiles are added simultaneously at any position in which an insufficient attachment can occur. Specifically, any particular pair of tile types that can create an insufficient attachment in the kTAM is added at a rate $f_{err} = O(e^{-3G_{se}})$. (This is asymptotically the rate that insufficient attachments occur in kTAM [4].) Thus the total rate of insufficient attachments at a particular location is Qf_{err}, where Q is the number of different ways (with different tile types) that an insufficient attachment can occur there. We don't absorb Q into the $O(\cdot)$ notation because we will be considering tile sets with an increasing number of tile types

that can cause errors. Note that Q can be bounded by the square of the total number of tile types. These insufficient attachments are the sole cause of errors during growth.[1] Growth during which no insufficient attachments occur we call (reversible) $\tau = \mathbf{2}$ **growth**.

1.3 Quarter-Plane Patterns

The output of the self-assembly process is usually considered to be either the shape of the uniquely produced terminal assembly [10, 1, 2, 20] or the pattern produced if we focus on the locations of certain types of tiles [24, 4, 17, 6]. Here we will focus on self-assembling of **quarter-plane patterns**. A quarter-plane pattern (or just **pattern** for short) $\mathbf{P}$ is an assignment of symbols from a finite alphabet of "colors" to points on the quarter plane ($\mathbb{Z}^+ \times \mathbb{Z}^+$ by convention). A deterministic tile system can be thought to construct a pattern in the sense that there is some function (not necessarily a bijection) mapping tile types to colors such that tiles in any produced assembly correctly map to corresponding colors of the pattern. As the assembly grows, a larger and larger portion of the pattern gets filled. There are patterns that cannot be deterministically constructed by any tile system (e.g., uncomputable ones), but for the purposes of this paper we consider patterns constructible from deterministic tile systems where all bond strengths are 1 and the seed assembly (defining the boundary conditions) is an infinite L shape that is eventually periodic, with its corner on the origin. See Fig. 1 for two examples. Such tile system we'll call **quarter plane tile systems** and the patterns produced by them the **constructible quarter-plane patterns**. These systems include a wide variety of patterns, including the Sierpinski pattern, the binary counter pattern, the Hadamard pattern [6], and patterns containing the space-time history of arbitrary 1D block cellular automata and Turing machines. Note that by including the infinite seed assembly we are avoiding the issue of nucleation, which requires distinct error correcting techniques [18].

[1] Another error, with respect to the aTAM, that can occur in the original kTAM is when a tile attached by strength 3 (or more) falls off. Why do we feel comfortable neglecting this error in the locking kTAM, especially since as a function of G_{se}, r_3 and f_{err} are both $O(e^{-3G_{se}})$? One reason is that in practice the dissociation of tiles held to the assembly with strength 3 does not seem to cause the problems that insufficient attachments induce, in tile sets that we have simulated and examined: no incorrect tiles are immediately introduced, often the correct tile will quickly arrive to repair the hole, and if an incorrect tile fills the hole, further growth may be impossible, usually allowing time for the incorrect tile to fall off. A second reason is that as the number of tile types increases (i.e., with more complex patterns or more complex proofreading schemes), Qf_{err} becomes arbitrarily large, while r_3 stays constant. Nonetheless, a more satisfying treatment would not make these approximations and would address the original kTAM directly.

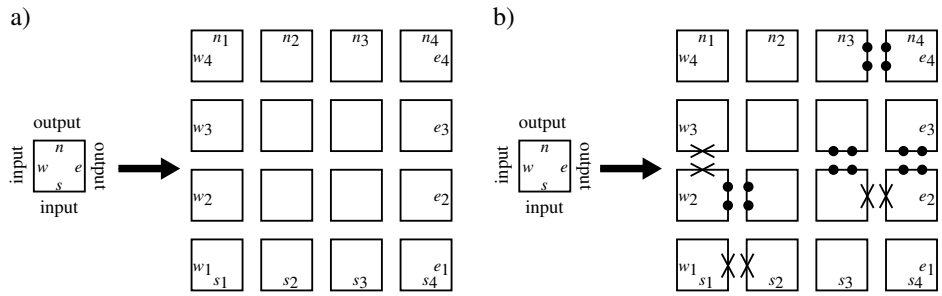

Fig. 2. Winfree and Bekbolatov (a) and Chen and Goel (b) proofreading transformations using 4×4 blocks. Each tile type is replaced with k^2 tile types that fit together to form the block as shown. Strength 2 bonds are indicated with 2 dots. Strength 0 bonds are indicated with a cross. All unlabeled (internal) bond types are unique (within the block and between blocks.) The placement of weak and strong bonds is dependent upon the orientation of growth, which in this case is to the north-east, since for quarter plane tile systems the input is always received from the west and south sides.

2 Making Self-Assembly Robust

The kinetic Tile Assembly Model predicts that for any quarter plane tile system, arbitrarily small error rates can be achieved by increasing G_{mc} and G_{se}, but at the cost of decreasing the overall rate of assembly. Specifically, the worst case analysis (which assumes that after any single error, assembly can be continued by valid $\tau = 2$ growth) predicts that the relationship between per tile error rate ε and the rate of assembly r (layers per second) approximately satisfies $r \propto \varepsilon^2$ [23]. This is rather unsatisfactory since, for example, decreasing the error rate by a factor of 10 necessitates slowing down self-assembly by a factor of 100.

Rather than talking about the relationship between the per tile error rate and the total rate of self-assembly, following [4] one can ask how long it takes to produce the correct $N \times N$ initial portion of the pattern with high probability. To produce this initial portion correctly with high probability, we need the per-tile error rate to be $\varepsilon = O(N^{-2})$ to ensure that no mistake occurs. This implies that $r = O(N^{-4})$ for worst case tile sets. This informal argument suggests that the time to produce the $N \times N$ square is $\Omega(N^4)$. This is unsatisfactory, because the same assembly can be grown in time $O(N)$ in the aTAM augmented with rates [1], and thus the cost of errors appears to be considerable.

Despite this pessimistic argument, certain kinds of tile systems can achieve better error rate/rate of assembly tradeoffs. Indeed, the reversibility of the self-assembly process can help. Some tile systems have the property that upon encountering an error, unless many more mistakes are made, the self-assembly process stalls. Stalling gives time for the incorrectly incorporated tiles to be eventually replaced by the correct ones in a random walk process, so long as not too many incorrect tiles have been added.

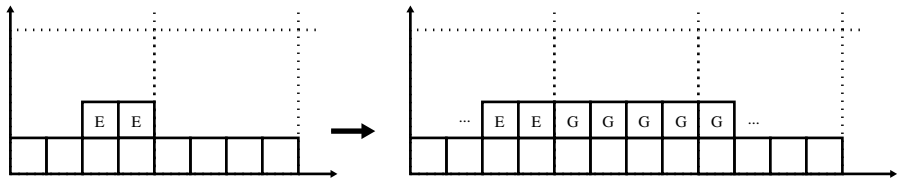

Fig. 3. The Winfree and Bekbolatov proofreading scheme is susceptible to single facet nucleation errors. If an insufficient attachment results in the two E tiles shown, then subsequent $\tau = 2$ growth (G) can continue indefinitely to the right. Thus many incorrect tiles can be added following a single facet nucleation error even if the block that E is in does not get completed. The dotted lines indicate block boundaries (for 4×4 blocks). Note that most of the incorrect tiles are attached with strength 3; therefore, they do not easily fall off, except at the left and the right sides.

Exploiting this observation, several schemes have been proposed for converting arbitrary quarter plane tile systems into tile systems producing a *scaled-up* version of the same pattern, resulting in better robustness to error. The initial proposal due to Winfree and Bekbolatov [24] suggests replacing each tile type of the original tile system with k^2 tile types, with unique internal strength-1 bonds (Fig. 2(a)). Such proofreading assemblies have the property that for a block corresponding to a single tile in the old system to get completed, either no mistakes, or at least k mistakes must occur. However, this scheme suffers from the problem that the self-assembly process after a single insufficient attachment can still result in a large number of incorrect tiles that must later be removed, spanning the length of the assembly. Consider the situation depicted in Fig. 3. If the insufficient attachment illustrated occurs (The first E is added with interaction strength 1, but before it dissociates, a tile attaches to it on the right with interaction strength 2), the incorrect information can be propagated indefinitely to the edge of the assembly by subsequent $\tau = 2$ tile additions.

Currently the only scheme that provably achieves a guaranteed level of proofreading is due to Chen and Goel [4] using the locking kTAM model. Their proofreading scheme, called *snaked proofreading*, is similar to the Winfree and Bekbolatov system, but additionally controls the order of self-assembly within each block by using strength-0 and strength-2 bonds, making sure that not too many incorrect tiles can be added by $\tau = 2$ growth after an insufficient attachment. In particular, the strength-0 bonds ensure that unless most of the block gets completed, self-assembly stalls. Fig. 2(b) shows their 4×4 construction; see their paper for the general construction for arbitrary block size.[2] They can

[2]Note that unlike the original proofreading transformation, the snake proofreading transformation does not result in a quarter plane tile system as it uses both strong and weak bonds.

attain a polynomial decrease in the error rate with only a logarithmic increase in k. Specifically the formal results they obtain are the following:[3,4]

Theorem 1 (theorem 4.2 of [4]). *For any constant $p < 1$, the $N \times N$ block initial portion of the pattern is produced correctly with probability at least p in time $O(Npoly(\log(N))$ by the $k \times k$ snaked proofreading tile system where $k = \theta(\log N)$, using the locking kTAM with appropriate G_{mc} and G_{se}.*

To obtain this result, assembly conditions (G_{mc} and G_{se}) need be adjusted only slightly as N increases.[5]

The above construction requires increasing the scale of the produced pattern, even if only logarithmically in the size of the total desired size of the self-assembled pattern. Reif et al [17] pointed this out as a potential problem and proposed schemes for decreasing the effective error rate while preserving the scale of the pattern. However, they rely on certain specific properties of the original tile system, and do not provide a general construction that can be extended to arbitrary levels of error correction. Further, their constructions suffer from the same problem as the original Winfree and Bekbolatov proofreading system. In the next section we argue that the snaked proofreading construction can be adopted to achieve same-scale proofreading for sufficiently "simple" patterns.

3 Compact Proofreading Schemes for Simple Patterns

In this section we argue that a wide variety of sufficiently "simple" patterns can be produced with arbitrarily small effective error rates without increasing the scale of self-assembly, at the cost of slightly increasing the number of tile types and the time of self-assembly. Based on Reif et al's nomenclature [17], we call these proofreading schemes *compact* to indicate that the scale of the pattern is not allowed to change.

The following definition illustrates our goal:

[3] [4] also guarantees that the assembly is stable for a long time after it is complete, a concern we ignore in this paper. For fixed k, they also provide theorem 4.1, which guarantees reliable assembly of an $N \times N$ square in time $O(N^{1+8/k})$.

[4] Chen and Goel only prove their result for the case when the initial L seed assembly has arms that span exactly N blocks. We need to cover the case when an infinite L seed assembly is used. See Appendix A for a proof that their results can be extended to an infinite seed assembly.

[5] It is hard to say whether the snaked proofreading construction is asymptotically optimal. While the best possible assembly time in a model where concentrations are held constant with changing N is linear in N, we assume that G_{mc} and G_{se} are free to change as long as the relationship $f = r_2$ is maintained. Of course while decreasing G_{mc} and G_{se} speeds up the assembly process, the rate of errors is increased; thus, the optimal tradeoff is not obvious.

Definition 1. *Let $p < 1$ be a constant (e.g., 0.99). A sequence of deterministic tile systems $\{\mathbf{T}_1, \mathbf{T}_2, \ldots\}$ is a* ***compact proofreading scheme*** *for pattern* $\mathbf{P}$ *if:*

(1: correctness) $\mathbf{T}_N$ *produces the full infinite pattern* $\mathbf{P}$ *under the aTAM.*
(2: conciseness) $\mathbf{T}_N$ *has* $poly(\log N)$ *tile types.*
(3: robustness) $\mathbf{T}_N$ *produces the correct* $N \times N$ *initial portion of pattern* $\mathbf{P}$ *(without scaling) with probability at least* p *in time* $O(Npoly(\log N))$ *in the locking kTAM for some* G_{se} *and* G_{mc}*.*

If you want to construct the initial $N \times N$ portion of pattern $\mathbf{P}$ with probability at least p in time $O(Npoly(\log N))$ you pick tile system $\mathbf{T}_N$ and the corresponding G_{se} and G_{mc}. The same tile system might be used for many N (i.e., the sequence of tile systems may have repetitions). The second condition indicates that we don't want this tile system to have too many tile types. For constructible quarter plane patterns, a constant number of tile types suffices to create the infinite pattern in the absence of errors. If the second condition is satisfied then the error correction itself is accomplished with a polylogarithmic number of additional tiles, which is comparable to the cost of error correction in other models studied in computer science. While one can imagine different versions of these conditions, the stated version gives the proofreading condition that can be obtained by adapting the snaked proofreading construction, as argued below. Finally, note that the tile systems $\{\mathbf{T}_1, \mathbf{T}_2, \ldots\}$ do not have to be quarter plane tile systems, and therefore our theorems will apply to a wide range of potential proofreading schemes.

For which patterns do there exist compact proofreading schemes? Given a pattern and a quarter plane tile system $\mathbf{T}$ producing it, consider any assembly of $\mathbf{T}$. For a given k, imagine splitting the assembly into $k \times k$ disjoint blocks starting at the origin. We'll use the term **block** to refer to aligned blocks, and **square** to refer to blocks without the restriction that they be aligned to integer multiples of k with respect to the origin. Each complete block contains k^2 tiles; two blocks at different locations are considered equivalent if they consist of the same arrangement of tile types. If there is some polynomial $Q(k)$ such that repeating this process for all assemblies and all k yields at most $Q(k)$ different (completed) block types, then we say that $\mathbf{T}$ **segments** into $poly(k)$ $k \times k$ block types.[6] Patterns produced by such tile systems are the "simple" patterns, for which, we will argue, there exist compact proofreading schemes; we term such patterns **robust** to indicate this.

On the other hand, there are patterns for which it is easy to see that no quarter tile system producing them segments into $poly(k)$ $k \times k$ block types.

[6] We use disjoint blocks aligned with the origin for simplicity in what follows. It is inessential that we define segmentation in terms of blocks rather than squares: A tile system segments into $poly(k)$ different $k \times k$ block types if and only if it produces assemblies that contain $poly(k)$ different types of non-aligned $k \times k$ squares. This is also true for other shapes than squares, as long as they have sufficient extent. See Appendix B for an example, the size-k diagonals.

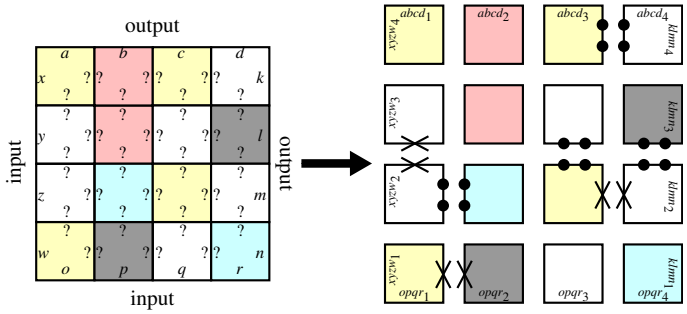

Fig. 4. Compact proofreading transformations using 4×4 blocks. Strength 2 bonds are indicated with 2 dots. Strength 0 bonds are indicated with a cross. Question marks indicate arbitrary bond types. All unlabeled (internal) bond types are unique (within the block and between blocks.) This construction is equivalent to "compressing" the $k \times k$ block on the left to a single tile and then applying the snaked proofreading construction, remembering to paint the resulting tiles with the original colors.

For example these include patterns which have $2^{\Omega(k)}$ different types of $k \times k$ squares of colors.[7] We'll prove negative results about such patterns, which we term **fragile** in the next section.[8]

Definition 2. *A pattern* $\mathbf{P}$ *is called* ***robust*** *if it is constructible by a quarter plane tile system* $\mathbf{T}$ *that segments into* $poly(k)$ *different* $k \times k$ *block types. A pattern* $\mathbf{P}$ *is called* ***fragile*** *if every quarter plane tile system segments into* $2^{\Omega(k)}$ *different* $k \times k$ *block types.*

The natural way to use Chen and Goel's construction to implement compact proofreading for robust patterns is as follows. For any k, for each of the $poly(k)$ $k \times k$ block types described above, create k^2 unique tile types with bond strengths according to the snaked proofreading blocks and colors according to the original pattern. The internal bond types are unique to each transformed $k \times k$ block type and do not depend upon the internal bond types in the original $k \times k$ block type. External bond types in the transformed block redundantly encode the full tuple of external bond types in the original block. (This transformation for a 4×4

[7] In what follows, we will consider both the number of blocks (or squares) in an assembly, in which case we mean blocks (or squares) of tile types, as well as the number of blocks (or squares) in a pattern, which which case we mean block (or squares) of colors. Since each tile type has a color, the latter is less than or equal to the former for patterns produced by quarter-plane tile systems.

[8] Analogous to the uncomputability of topological entropy for cellular automata [11], it is in general undecidable whether a tile set produces a robust or fragile pattern, due to the undecidability of the Halting Problem: a tile system that simulates a universal Turing machine may either produce a pattern that is eventually periodic (if the Turing machine halts), or else it may continue to produce ever more complicated subpatterns. The former patterns (that are eventually periodic) are formally robust, although only for very large k does this become apparent, while the latter patterns are fragile.

block is illustrated in Fig. 4.) The L-shaped seed assembly must also be revised to use the new compound bond types. The above set of tile types together with this seed assembly yields a new tile system $\mathbf{T}(k)$.

It is easy to check that under aTAM $\mathbf{T}(k)$ correctly produces the pattern. At the corner between two existing blocks, only a tile that matches all the border tiles of both blocks, can attach. Any other internal tile must bind correctly since at least one side must match a bond type unique to the block. Since the original block assembled deterministically from its west and south sides, the transformed block also grows deterministically in the same direction. In fact, $\mathbf{T}(k)$ is locally deterministic [20], which makes a formal proof easy. Furthermore, for any particular choice of k, Chen and Goel's theorem 4.1 applies directly to our compact proofreading tile sets, but with multiplicative constants that increase with k. But we also claim the following, where $M = \lceil N/k \rceil$ is the size of our target assembly in units of blocks:

Lemma 1. *If a pattern* $\mathbf{P}$ *is robust then: For any constant* $p < 1$*, the* $M \times M$ *block initial portion of the pattern is produced correctly with probability at least* p *in time* $O(M poly(\log M))$ *by some* $\mathbf{T}(k)$ *(as defined above) where* $k = \theta(\log M)$*, using the locking kTAM with appropriate* G_{mc} *and* G_{se}*.*

Proof. Recall, as long as a particular location remains susceptible, insufficient attachments at that location constitute a Poisson process with rate $QO(e^{-3G_{se}})$. Here Q can be upper bounded by the total number of different blocks since that is the maximum number of different tile types that can be added as an insufficient attachment at any location. Thus, the maximum rate of insufficient attachments at any location is $q(G_{se}) = Q(k)O(e^{-3G_{se}})$, where $Q(k) = poly(k)$ since the pattern is robust.

The difference between the proof of Chen and Goel [4] and what we need is that Chen and Goel assumed that $Q(k)$ was a constant. Thus, whereas they were able to increase k without increasing the rate of insufficient attachments, q, we are not so fortunate. To remedy this situation, we must slow down growth slightly in order to sufficiently decrease the rate of insufficient attachments, but not so fast as to change the asymptotic form of the results.

Informally, note that Chen and Goel's bound on the probability of successfully completing the square within a certain time (scaled relative to f) depends only on the ratio q/f; the absolute time scale does not matter, nor does it matter whether q is the result of many or a few possible erroneous block types. Thus, we can slow down f by a polynomial in k without affecting the completion time asymptotics of $O(M poly(\log M)$, since $k = \Theta(\log M)$. Does q decrease enough? So long as it decreases faster relative to f, we can compensate for the polynomial increase in insufficient attachments. We will see that a factor of $Q(k)^2$ is sufficient.

Formally, assuming the maximum rate of insufficient attachments is any $\tilde{q}(G_{se}) = O(e^{-3G_{se}})$ independent of k, and the forward (=reverse) rate is any $\tilde{f}(G_{se}) = \Omega(e^{-2G_{se}})$, for any M, Chen and Goel give a value $\tilde{k}$ for k and $\tilde{G}_{se}$ for G_{se} such that with high probability the assembly completes correctly

in time $t = O(Mpoly(\log M))$. We, of course, have $q(G_{se}) = O(Q(k)e^{-3G_{se}})$ and $f(G_{se}) = \Omega(e^{-2G_{se}})$. Now let us define $\tilde{q}(G_{se}) = q(G_{se} + \ln Q(k)) \cdot Q(k)^2$ and $\tilde{f}(G_{se}) = f(G_{se} + \ln Q(k)) \cdot Q(k)^2$. Observe that $\tilde{q}(G_{se}) = O(e^{-3G_{se}})$ and $\tilde{f}(G_{se}) = O(e^{-2G_{se}})$. This means that if the maximum rate of insufficient attachments and the forward rate were these $\tilde{q}$ and $\tilde{f}$, then Chen and Goel's proof gives values $\tilde{k}$ and $\tilde{G}_{se}$ such that with high probability the assembly completes correctly in time $t = O(Mpoly(\log M))$. But now note that if we set $G_{se} = \tilde{G}_{se} + \ln Q(\tilde{k})$ then the actual maximum rate of insufficient attachments and the forward rate are both exactly a factor of $Q(\tilde{k})^2$ slower than $\tilde{q}$ and $\tilde{f}$. Thus our system is simply overall slower by a factor of $Q(\tilde{k})^2$. This means that our system would finish correctly with the same high probability as achieved by Chen and Goel by time $O(tQ(\tilde{k})^2)$. But this is still $O(Mpoly(\log M))$ since $\tilde{k} = \theta(\log M)$ and $Q(\tilde{k}) = poly(\tilde{k})$.

□

Theorem 2. *If a pattern* $\mathbf{P}$ *is robust then there exists a compact proofreading scheme for* $\mathbf{P}$.

Proof. Let us use the sequence $\{\mathbf{T}_N = \mathbf{T}(k)\}_N$ where k for each N is from lemma 1. Each of these tile systems can produce the whole pattern correctly under aTAM so the correctness condition of definition 1 is satisfied. Since $O(M\ poly(\log M)) = O(Npoly(\log N))$, lemma 1 implies that the sequence satisfies the robustness condition. Further, because $\mathbf{T}$ segments into $poly(k)$ different $k \times k$ block types and $k = \theta(\log M)$ implies $k = O(\log N)$, $\mathbf{T}_N = \mathbf{T}(k)$ has only $poly(k)k^2 = poly(\log N)$ tile types, satisfying the conciseness condition. □

For some patterns, Chen and Goel's theorem can be applied directly (without requiring lemma 1). These include patterns whose quarter plane tile systems segment into a constant number of $k \times k$ block types. Furthermore, consider the Sierpinski pattern (Fig. 1(c)). The Sierpinski pattern is a fractal that has the following property: split the pattern into blocks of size $k \times k$ for any k that is a power of 2, starting at the origin. For any such k there are exactly 2 different types of blocks in the pattern. If you consider the assembly produced by the Sierpinski tile system in Fig. 1(d), there are exactly 4 different $k \times k$ blocks of tiles (the difference is due to the fact there are now two types of black and two types of white tiles.) We can let the sequence of tile systems for the compact proofreading scheme for the Sierpinski pattern consist only of $\mathbf{T}(k)$ for k that are a power of 2. Note that because of the restriction on k, we may have to use a block size larger than that which results from Chen and Goel's theorem. But since it does not have to be more than twice as large, definition 1 is still satisfied.

It would be interesting to identify constructible quarter plane patterns that have at least k^d different $k \times k$ block types for all k and for some constant $d \geq 1$.

4 A Lower Bound

In this section we will show that we cannot make compact proofreading schemes for fragile patterns using known methods.

First of all, note that although the definition of fragile patterns quantifies over all quarter plane tile systems, it can be very easy to prove that a pattern is fragile using the following lemma.

Lemma 2. *If a pattern* $\mathbf{P}$ *has* $2^{\Omega(k)}$ *different types of* $k \times k$ *squares of colors then it is fragile.*

Proof. If a pattern contains $2^{\Omega(k)}$ different types of $k \times k$ squares of colors, then any tile system producing it contains at least $2^{\Omega(k)}$ different types of $k \times k$ squares, and therefore comparably many block types. □

The scheme described in the previous section does not work for quarter plane tile systems that segment into $2^{\Omega(k)}$ $k \times k$ block types (i.e., fragile patterns). This is because for $k = \theta(\log N)$, $\mathbf{T}(k)$ would then have $poly(N)$ tile types, violating the second condition (conciseness) of compact proofreading schemes (Definition 1).[9] However, it is unclear whether other methods exist to make compact proofreading schemes for patterns produced by such tile systems. While we cannot eliminate this possibility entirely, we can show that a variety of schemes will not work.

Existing attempts at making self-assembly robust through combinatorial means ([24, 4, 17]) are based on creating redundancy in the produced assembly. Specifically, knowing only a few tiles allows one to figure out a lot more of the surrounding tiles. Intuitively, this redundancy allows the tile system to "detect" when an incorrect tile has been incorporated and stall. We will argue that if a pattern is sufficiently complex, then only if there are many possible tile types can a few tiles uniquely determine a large portion of the pattern. Since the definition of compact proofreading schemes (Definition 1) limits the number of tile types, we will be able to argue that for complex patterns there do not exist compact proofreading schemes that rely on this type of redundancy.

Definition 3. *An assembly* A *is* (k, d)***-redundant*** *if there exists a decision procedure that, for any* $k \times k$ *(completed) square of tiles in* A*, querying at most* d *relative locations in the assembly for its tile type, can determine the types of all tiles in that square.*

The proofreading schemes of [24] and [4], using a block size $k \times k$, are $(k, 3)$-redundant: even if the square is not aligned with the blocks, it is enough to ask for the types of the tiles in the upper-left, lower-left, and lower-right corners of the square. Because all tiles in a block are unique, and because the tile system is deterministic, these three tiles allow you to figure out all four blocks that the square may intersect. A proofreading construction that generalizes Reif et al's [17] 2-way and 3-way overlay tile sets to k-way overlays is shown in Appendix B to be $(k, 3)$-redundant as well. This construction is not based on block transformations; the fact that its power is nonetheless limited by Theorem 3, below, illustrates the strength of our lower bound.

[9] Further, we believe Lemma 1 does not hold if the number of block types increases exponentially, rather than polynomially in k. This is an open question.

Lemma 3. *If a tile system* $\mathbf{T}$ *produces* (k, d)*-redundant assemblies in which more than* 2^{ck} *different types of (completed)* $k \times k$ *squares appear, then it must have at least* $2^{ck/d}$ *tile types.*

Proof. Let m be the number of tile types of $\mathbf{T}$. If an assembly produced by $\mathbf{T}$ is (k, d)-redundant, then it has no more than m^d types of squares of size $k \times k$ because the decision procedure's decision tree is of depth at most d and of fan-out at most m. But we assumed that $\mathbf{T}$ makes assemblies that have 2^{ck} different types of $k \times k$ squares. Thus, $m^d \geq 2^{ck}$, which can only happen if $m \geq 2^{ck/d}$. □

Lemma 4 lets us limit the types of compact proofreading schemes that such complex patterns may have.

Theorem 3. *If a pattern is fragile then there does not exist a compact proofreading scheme* $\{\mathbf{T}_1, \mathbf{T}_2, \ldots\}$ *such that* $\mathbf{T}_N$ *produces assemblies that are* $(\Omega(\log N), d)$*-redundant (for any constant* d*).*

Proof. Any tile system producing this pattern makes $2^{\Omega(k)}$ different types of $k \times k$ (completed) squares of tiles. Suppose $\mathbf{T}_N$ produces assemblies which are $(c' \log N, d)$-redundant, for constants c', d. Take $k = c' \log N$ and note that for large k, $\mathbf{T}_N$ makes at least 2^{ck} $k \times k$ squares for some constant c. Apply Lemma 3 to conclude that $\mathbf{T}_N$ has at least $2^{ck/d} = N^{cc'/d}$ tile types, which violates the second condition of Definition 1. □

Even though both the Sierpinski pattern and the counter pattern (Fig. 1) are infinite binary patterns that can be constructed by very similar tile systems, they are very different with respect to error correction. We saw that the Sierpinski pattern has compact proofreading schemes. However, because the counter must count through every binary number, for any k there are 2^k rows that have different initial patterns of black and white squares. This implies that there are exponentially many (in k) different squares. By Theorem 3 this implies that the counter pattern does not have compact proofreading schemes that use $(\Omega(\log N), d)$-redundant assemblies. That is, no existing proofreading scheme can be adapted for making compact binary counters arbitrarily reliable.

This theorem suggests that in order to find universal compact proofreading schemes we must find a method of making self-assembly more error-robust without making it too redundant. However, we conjecture that there are inherent tradeoffs between robustness and conciseness (small number of tile types) raising the possibility that there do not exist compact proofreading schemes for patterns having an exponential number of $k \times k$ squares.

Acknowledgments: We thank Ho-Lin Chen, Ashish Goel, Paul Rothemund, Matthew Cook, and Nataša Jonoska for discussions that greatly contributed to this work. This work was supported by NSF NANO Grant No. 0432193.

References

1. L. M. Adleman, Q. Cheng, A. Goel, and M.-D. A. Huang. Running time and program size for self-assembled squares. In *ACM Symposium on Theory of Computing (STOC)*, pages 740–748, 2001.
2. G. Aggarwal, M. Goldwasser, M. Kao, and R. T. Schweller. Complexities for generalized models of self-assembly. In *Symposium on Discrete Algorithms (SODA)*, pages 880–889, 2004.
3. R. D. Barish, P. W. K. Rothemund, and E. Winfree. Two computational primitives for algorithmic self-assembly: Copying and counting. *NanoLetters*, to appear.
4. H.-L. Chen and A. Goel. Error free self-assembly using error prone tiles. In Ferretti et al. [7], pages 62–75.
5. J. Chen and J. Reif, editors. *DNA Computing 9*, volume LNCS 2943, Berlin Heidelberg, 2004. Springer-Verlag.
6. M. Cook, P. W. K. Rothemund, and E. Winfree. Self-assembled circuit patterns. In Chen and Reif [5], pages 91–107.
7. C. Ferretti, G. Mauri, and C. Zandron, editors. *DNA Computing 10*, volume LNCS 3384, Berlin Heidelberg, 2005. Springer-Verlag.
8. P. W. K. Rothemund. *Theory and Experiments in Algorithmic Self-Assembly*. PhD thesis, University of Southern California, Los Angeles, 2001.
9. P. W. K. Rothemund, N. Papakakis, and E. Winfree. Algorithmic self-assembly of DNA Sierpinski triangles. *PLoS Biology*, 2:e424, 2004.
10. P. W. K. Rothemund and E. Winfree. The program-size complexity of self-assembled squares. In *ACM Symposium on Theory of Computing (STOC)*, pages 459–468, 2000.
11. L. Hurd, J. Kari, and K. Culik. The topological entropy of cellular automata is uncomputable. *Ergodic Theory Dynamical Systems*, 12:255–265, 1992.
12. T. H. LaBean, H. Yan, J. Kopatsch, F. Liu, E. Winfree, J. H. Reif, and N. C. Seeman. Construction, analysis, ligation, and self-assembly of DNA triple crossover complexes. *Journal of the Americal Chemical Society*, 122:1848–1860, 2000.
13. M. G. Lagoudakis and T. H. LaBean. 2-D DNA self-assembly for satisfiability. In E. Winfree and D. K. Gifford, editors, *DNA Based Computers V*, volume 54 of *DIMACS*, pages 141–154, Providence, RI, 2000. American Mathematical Society.
14. C. Mao, T. H. LaBean, J. H. Reif, and N. C. Seeman. Logical computation using algorithmic self-assembly of DNA triple-crossover molecules. *Nature*, 407:493–496, 2000.
15. C. Mao, W. Sun, and N. C. Seeman. Designed two-dimensional DNA holliday junction arrays visualized by atomic force microscopy. *Journal of the Americal Chemical Society*, 121:5437–5443, 1999.
16. J. Reif. Local parallel biomolecular computing. In H. Rubin and D. H. Wood, editors, *DNA Based Computers III*, volume 48 of *DIMACS*, pages 217–254, Providence, RI, 1999. American Mathematical Society.
17. J. H. Reif, S. Sahu, and P. Yin. Compact error-resilient computational DNA tiling assemblies. In Ferretti et al. [7], pages 293–307.
18. R. Schulman and E. Winfree. Programmable control of nucleation for algorithmic self-assembly. In Ferretti et al. [7], pages 319–328.
19. R. Schulman and E. Winfree. Self-replication and evolution of DNA crystals. In M. S. Capcarrere, A. A. Freitas, P. J. Bentley, C. G. Johnson, and J. Timmis, editors, *Advances in Artificial Life: 8th European Conference (ECAL)*, volume LNCS 3630, pages 734–743. Springer-Verlag, 2005.

20. D. Soloveichik and E. Winfree. Complexity of self-assembled shapes, 2005. Extended abstract; preprint of the full paper is cs.CC/0412096 on arXiv.org.
21. E. Winfree. On the computational power of DNA annealing and ligation. In R. J. Lipton and E. B. Baum, editors, *DNA Based Computers*, volume 27 of *DIMACS*, pages 199–221, Providence, RI, 1996. American Mathematical Society.
22. E. Winfree. *Algorithmic Self-Assembly of DNA*. PhD thesis, California Institute of Technology, Pasadena, 1998.
23. E. Winfree. Simulations of computing by self-assembly. Technical Report CS-TR:1998.22, Caltech, 1998.
24. E. Winfree and R. Bekbolatov. Proofreading tile sets: Error-correction for algorithmic self-assembly. In Chen and Reif [5], pages 126–144.
25. E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman. Design and self-assembly of two dimensional DNA crystals. *Nature*, 394:539–544, 1998.
26. E. Winfree, X. Yang, and N. C. Seeman. Universal computation via self-assembly of DNA: Some theory and experiments. In L. F. Landweber and E. B. Baum, editors, *DNA Based Computers II*, volume 44 of *DIMACS*, pages 191–213, Providence, RI, 1998. American Mathematical Society.

A Extension of Chen and Goel's theorem to infinite seed boundary assemblies

The following argument uses terms and concepts from [4].

First, suppose we desire to build an $(N + N^2) \times (N + N^2)$ block initial portion of the pattern starting with the L seed assembly having arms that are $N + N^2$ blocks long. The extra N^2 blocks will serve as a buffer region. Chen and Goel's [4] theorem 4.2 then gives us a $k = \theta(\log(N + N^2)) = \theta(\log N)$ and G_{se} s.t. with high probability no block error occurs in the $(N+N^2) \times (N+N^2)$ block region in time $O(N^2 poly(\log N))$ that it takes to finish it. Further, with high probability the initial $N \times N$ block portion of the pattern is completed in time $t_N = O(N poly(\log N))$.

Now, let's suppose we use this k and G_{se} with an infinite L seed assembly, and we'll be interested in just the $N \times N$ block initial portion of the pattern. The only way the infinite seed assembly can affect us is if a block error outside the $(N+N^2) \times (N+N^2)$ block region propagates to the $N \times N$ initial region before it completes. For this to occur, at least N^2 tiles must be added sequentially, at least one per block through the buffer region, to propagate the error. The expected time for this to happen is N^2/f with standard deviation N/f (i.e., it is a gamma distribution with shape parameter N^2 and rate parameter f). However, the propagated error can only cause a problem if it reaches the $N \times N$ rectangle before time t_N. Since $t_N = O(N poly(\log N))$, this becomes less and less likely as N increases by Chebyshev's inequality. Small N are handled by increasing k and G_{se} appropriately, which does not affect the asymptotic results. Thus we have a $k = \theta(\log N)$ and G_{se} such that with high probability (i.e., $\geq p$) the initial $N \times N$ block portion of the pattern is completed correctly in time $O(N poly(\log N))$, even if we use an infinite L seed assembly.

B An Overlay Proofreading Scheme

In this appendix we give an example showing that our lower bound on the complexity of same-scale proofreading schemes also applies to proofreading schemes that are not based on block transformations. Here, we consider a k-way overlay scheme (suggested by Paul Rothemund and Matt Cook) that generalizes the 2-way and 3-way overlay schemes introduced by Reif et al [17]. The construction is shown in Fig. 5.

Consider the assembly grown using some original tile set, as in Fig. 5a. When the shaded tile x was added, it attached to the tiles a and b to its west and to its south. Since we consider only deterministic quarter-plane tiles sets, the tile type at a particular location is a function of the tile types to its south and to its west, e.g., $x = f(a, b) = f_{ab}$. Therefore, it is possible to reconstruct the same pattern without keeping track of bond types, explicitly transmitting only information about tile types.

The 1-overlay tile set, derived from the original tile set, is a deterministic tile set for doing exactly that. As shown in Fig. 5b, for each triple of neighboring tiles a,b, and x that appears in the assembly produced by the original tile set (in the relative positions shown in (a)), create a new tile (x, x, b, a), colored the same as x, that "inputs" the original tile types of its west and its south neighbors, and "outputs" tile type x to both its north and its east neighbor. With an appropriately re-coded L-shaped boundary, the new tile set will produce exactly the same pattern as the original tile set: the output of the tile at location $\langle i, j\rangle$ in the 1-overlay assembly is the tile type at $\langle i, j\rangle$ in the original assembly. Supposing the original tile set T has $|T|$ tile types, the new tile set contains at most $|T|^2$ tile types, and possibly fewer if not all pairs of inputs a, b appear in the pattern.

Redundancy is achieved in a k-way overlay tile set by encoding not just one original tile, but k adjacent tiles along the diagonal growth front. Specifically, each tile in the k-way overlay assembly will output the k-tuple of original tile types that appear in the same location in the original assembly and locations to the east and south. For example, in Fig. 5c, the output of the tile at $\langle i, j\rangle$ in the 4-overlay assembly is the 4-tuple *abcd* containing the tile types at locations $\langle i, j\rangle$, $\langle i + 1, j - 1\rangle$, $\langle i + 2, j - 2\rangle$, and $\langle i + 3, j - 3\rangle$. Each new tile is colored according to the first tile type in its output tuple. The new tile set consists of all such tiles that appear in the k-overlay assembly[10,11]. The new tile set contains at most $|T|^{k+1}$ tiles, since there are at most $|T|^k$ input k-tuples, and the two

[10] In addition, the L-shaped boundary must be properly re-coded to carry the boundary information in the form the new tiles require. This is easy to do if the pattern is consistent with a larger hypothetical assembly that extends k tiles beyond the quarter plane region, since then tuples on the boundary encode for tile types in this buffer zone. Otherwise a few extra tile types will be necessary, but as this does not change the nature of our arguments, we ignore this detail here.

[11] Note that the exact (minimal) set of such tiles is in general uncomputable, since the original tile set could be Turing-universal, and thus predicting whether a particular original tile appears in the assembly is equivalent to the Halting Problem. However, the new tile set is well-defined and in many cases can be easily computed.

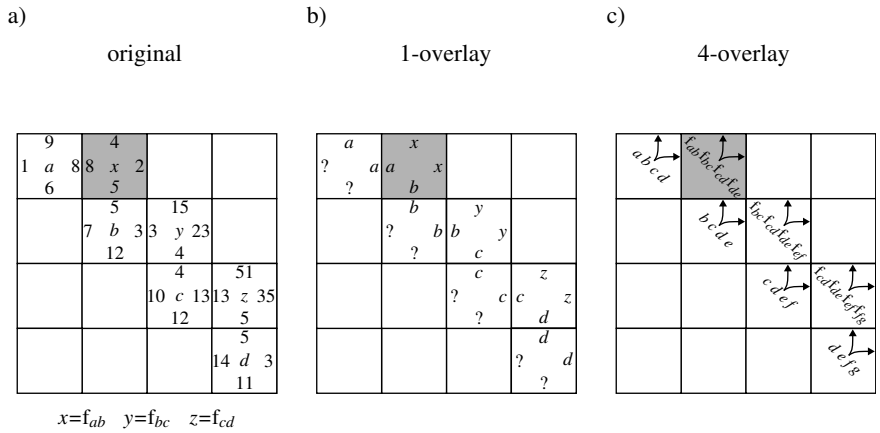

Fig. 5. The construction for k-way overlay proofreading tile sets. **(a)** An original quarter-plane tile set T, containing $|T|$ tile types. Numbers indicate bond types. Letters name the tile types. For example, the tile $x = (4, 8, 5, 2)$. **(b)** The 1-overlay transformation of the original tile set. The question marks indicate that there may be several different new tile types that output a or b; **(c)** The 4-overlay transformation of the original tile set.

inputs to a given tile will always agree at $k-1$ indices. This is exponential in k, but for some patterns – e.g., robust patterns, as we will see – only a polynomial number of tile types will be necessary. Note that growth with the new tile set is still deterministic, since the tuple output by a tile is a function of the two input tuples.

In what sense is the k-overlay tile set guaranteed to be proofreading? Consider a growth site where a tile is about to be added. Unless the two input k-tuples agree at all $k-1$ overlapping positions, there will be no tile that matches both inputs. Thus, every time that a tile is added without a mismatch, it provides a guarantee that $k-1$ parallel computations are carrying the same information, locally. Note that the fact that site $\langle i, j \rangle$ in the original assembly contains tile type t is encoded in k locations in the k-overlay assembly. It is reasonable to conjecture that it is impossible for all k locations to have incorrect information, unless at least k insufficient attachments have occurred.

Unfortunately, like the original proofreading tile sets of [24] and the 2-way and 3-way overlay tile sets described in [17], the k-way overlay tile sets do not protect against facet nucleation errors, and therefore we do not expect error rates to decrease substantially with k. We do not see an obvious way to correct this deficiency.

Nonetheless, as a demonstration of the general applicability of our lower bound, we will show that *even if* the k-way overlay tile sets reduced errors sufficiently, for fragile patterns the k-way overlay tile sets will contain an exponential number of tile types and are thus infeasible, whereas for robust patterns the k-

way overlay tile sets will contain a polynomial number of tile types and are thus feasible.

First we show that all k-overlay tile sets are $(k, 3)$-redundant, regardless of the original tile set. To determine all tile types in the $k \times k$ square with lower left coordinate $\langle i, j \rangle$, we need only know the tiles at $\langle i, j-1 \rangle$, $\langle i-k, j+k-1 \rangle$, and $\langle i+k-1, j-k \rangle$. The outputs of these tiles encodes for the entire diagonal from $\langle i-k, j+k-1 \rangle$ to $\langle i+2k-2, j-2k+1 \rangle$ in the original assembly. Deterministic growth from this diagonal results in a triangle of tiles with upper right corner at $\langle i+2k-1, j+k-1 \rangle$, in the original assembly. Thus all tile types are known for the input and output k-tuples of overlay tiles in the $k \times k$ square of interest.

Theorem 3 tells us that fragile patterns cannot have compact proofreading schemes that are $(\Omega(\log N), d)$-redundant for any constant d. Therefore, k-overlay tile sets can't work as compact proofreading schemes for fragile patterns; they must have an exponential number of tile types. This is what we wanted to show.

Alternatively, we could have directly bounded the number of tile types in k-overlay tile sets for fragile and robust patterns. For robust patterns, with $poly(k)$ $k \times k$ squares of tile types, clearly there are also $poly(k)$ size-k diagonals. Since each tile in the k-overlay tile set contains two inputs encoding size-k diagonals, there can be at most $poly(k)^2 = poly(k)$ tile types altogether. Thus, (although probably not satisfying the robustness criterion of Definition 1) k-overlay tile sets are at least concise for robust patterns. Conversely, concise k-overlay tile sets, having $poly(k)$ tile types by construction, have a comparable number of size-k diagonals in the original assembly. Consider now the original assembly. Since growth is deterministic, the diagonal determines the upper right half of a $k \times k$ square, and thus there are $poly(k)$ tops and $poly(k)$ sides; taking these as inputs to other squares, we see that there are $poly(k)^2 = poly(k)$ $k \times k$ squares. In this loose sense, k-overlay tile sets are neither more nor less concise than $k \times k$ snaked proofreading, for robust patterns.

On the other hand, for a fragile pattern, requiring $2^{\Omega(k)}$ $k \times k$ squares of tiles in any tile system that produces it, we can see that there will also be at least $2^{\Omega(k)}$ size-k diagonals of tiles. Specifically, if $S(k)$ is the number of such squares, and $D(k)$ is the number of such diagonals, then $S(k) \leq D(2k)$ because deterministic growth from a size-$2k$ diagonal results in the completion of a triangular region containing a $k \times k$ square. $S(k)$ being at least exponential therefore implies the same for $D(k)$. Conversely, a pattern generated by a tile system with $2^{\Omega(k)}$ size-k diagonals obviously also has at least that many $k \times k$ squares as well. Thus, our notions of fragile and robust patterns appears to be sufficiently general.