Self-Assembling Tile Systems that Heal from Small Fragments*

Holin Chen
Stanford University
holin@stanford.edu

Ashish Goel
Stanford University
ashishg@stanford.edu

Chris Luhrs
Stanford University
cluhrs@stanford.edu

Erik Winfree
California Institute of Technology
winfree@dna.caltech.edu

May 4, 2007

Abstract

Tile systems have proved to be a useful model for understanding self-assembly at the nano scale. Self-healing tile systems, introduced by Winfree, have the property that the self-assembled shape can recover from the loss of arbitrarily many tiles, provided the seed tile of the assembly remains intact and the assembly remains connected. In this paper, we present improved self-healing tile systems for the self-assembly of several interesting classes of shapes, including counters, squares, and Turing-computable shapes. Our tile systems can recover from the loss of arbitrary many tiles, including the seed, provided that a large enough fragment (logarithmic in the size of the desired assembly for the case of counters and squares) is left intact.

*Research supported by grants from the National Science Foundation.

1 Introduction

Molecular self-assembly has emerged as an important tool for nano-scale fabrication, molecular computation, crystallography, and nano-machines. On the experimental side, the combinatorial nature of DNA molecules, their relatively simple geometry (compared to smaller molecules such as the protein), and the existence of well developed laboratory techniques for creating and manipulating DNA molecules have made DNA an important tool for exploring self-assembly at the molecular scale. DNA strands have been used to design small widgets or "tiles" [27], which have then been used to perform molecular computation [28, 3], make interesting and useful patterns [24], create simple nano-scale machines [16, 15, 22, 30, 17, 29, 20], and as scaffolds for hybrid (organic-inorganic) self-assembly.

Because of the combinatorial nature and simple geometry of the DNA molecule, DNA self-assembly has been amenable to algorithmic modeling and analysis. The tile model of self-assembly [23, 12, 24] has proved to be useful in designing laboratory experiments and in exploring the capabilities and the limitations of molecular self-assembly. In this model, a square tile is the basic unit of an assembly. Each tile has a glue on each side; each glue has a label and a strength (typically 1 or 2). A tile can attach to a position in an existing assembly if at all the edges where this tile "abuts" the assembly, the glues on the tile and the assembly are the same, and the total strength of these glues is at least equal to a system parameter called the temperature (typically 2). Assembly starts from a single seed crystal and proceeds by repeated accretion of single tiles. The speed of an addition (and hence the time for the entire process to complete) is determined by the concentrations of different tiles in the system. Details are in Section 2. Abstract tile systems have been designed that can count [12, 1, 9], efficiently produce complex shapes such as squares, fractals, and spirals [12, 1], do universal/NP-complete computation [24, 10], and produce arbitrary "computable" shapes. Stochastic analyses of these systems have also been performed [5, 4, 2].

In nature, self-assembling systems are often equipped with explicit mechanisms for both error prevention and error correction. For artificial self-assembly, these problems are even more acute since we are interested in assembling large systems with great precision. Experimentally, several effects are observed which lead to a loss of robustness compared to the model. The assembly tends to be reversible, i.e., tiles can fall away from an existing assembly. Also, incorrect tiles sometimes get incorporated and locked into a growing assembly, much like defects in a crystal. However, for sophisticated combinatorial assemblies (like counters, which form the basis for controlling the size of a structure), a single error can lead to assemblies drastically larger or smaller (or different in other ways) than the intended structure. Finally, the temperature of the system can be controlled only imperfectly. Experimental studies of algorithmic self-assembly have observed error rates of 1% to 10% [26]. Several error correction schemes have been proposed and analyzed [26, 7, 11, 18, 14], and preliminary experiments point to the promise of these approaches [8, 14]. While developed with DNA in mind, the tile model seems very natural, and we believe it likely that techniques based on this model should carry over to other self-assembly regimes.

In this paper, we address one specific type of error: tiles falling off from a self-assembled structure. We do not assume any stochastic distribution on how tiles fall off; our goal is to design tile systems which can recover from catastrophic events such as the loss or damage of chunks of arbitrary size and position in a self-assembled structure, provided a significant fragment of the structure is left undamaged. This problem was initially posed by Winfree [25]. Winfree defined a self-healing tile system as one where a self-assemble structure can recover (or "heal") from an arbitrary loss of tiles provided that the structure that is left behind is connected and contains the seed. To use a crude analogy, this is akin to a lizard growing a tail, provided its head and torso remain intact. Winfree also presented general transformations to make a tile system self-healing under this definition. We present tile systems for several important tasks (such as counting, assembling squares, and assembling arbitrary shapes) with a stronger guarantee – a self-assembled structure can self-heal from damage even if it loses the seed, provided that some "large enough" piece is left behind; for counters and squares, the size of the remaining piece needs to only be logarithmic in the height of the counter or the square.

The quantitative improvement obtained by the new definition is clear – our tile systems can heal under a larger range of scenarios. However, we believe that this definition is also qualitatively stronger. Consider a finite binary counter (i.e. a $k \times 2^k$ sized rectangle) that starts from a single seed. If tiles are being removed from a completed assembly in an iid fashion with some small constant rate r, then the seed and the tiles surrounding the seed will get damaged at time $O(1/r)$, and hence, a construction that is self-healing in the sense of [25] will have a constant lifetime. In contrast, the finite binary counter that we present in this paper can recover as long as there is any connected path between the leftmost and the rightmost column, and hence will have a super-constant lifetime; in fact we conjecture that this time will be exponential in k, and that the infinite binary counter we present will have an infinite lifetime with positive probability in this model. More philosophically, the tile systems presented in this paper can be thought of as having a reproductive property – if a structure is broken into two parts, each part will have enough encoded information to be able to grow back into a complete structure of its own provided it is not too small. To continue the earlier crude analogy, this is akin to a complete starfish being regenerated from a very small fragment.

We formally define the self-healing property as well as the tile model of self-assembly in section 2. Informally, a tile system is said to be "self-healing" from a set of configurations (i.e. partial assemblies) S if it can recover from the loss of arbitrarily many tiles provided all the tiles in (at least) one of the configurations in the set S is left intact.

1.1 Our results

We first observe (section 2) that the self-healing property reduces to two simple and easily verified properties: immutability and progressiveness. Immutability corresponds to a wrong tile never being able to attach even if a correct tile falls off; progressiveness corresponds to the ability of the tile system to continue growing as long as one out of a special set of configurations is present. Where there is a unique terminal assembly, these properties are easy to check either manually by inspection, or computationally in time polynomial in the size of the terminal assembly and the number of special configurations from which the tile system is progressive.

We then study (section 3) a simple counter based on the Chinese remainder theorem; this counter is of width k and height $p_1p_2 \dots p_k$ where $p_1, p_2, \dots, p_k$ are distinct primes. We show that this counter is self-healing from all configurations which contain a path from the west-most column of the counter to the east-most. This construction can be extended to numbers which are not products of unique primes by using two counters which count in the opposite direction simultaneously, a technique that we call reverse augmentation.

Next, we present (section 4) a binary counter that uses only a constant number of tiles, "counts" from 0 to ∞, and is self-healing from all configurations which contain a path from a west-most tile to an east-most tile. The same ideas can be used to obtain a finite binary counter that is self-healing from the same set of configurations. It is interesting to note that binary counters that had been proposed earlier (eg. [9]) had the self-healing property; in particular they are not immutable. Since binary counters form the basis of many other constructions, we believe our self-healing tile sets will be a useful addition to the robust self-assembly tool kit.

Finally, we present (section 5.1) a self-healing tile system for assembling $n \times n$ squares. One possible approach would be to use two counters to bound the two sides of a square, and then use the general transformation of Winfree to assemble a self-healing square inside. However, such a square would only be self-healing from configurations that contain a complete counter row. We present a construction with a stronger guarantee: the entire square can self-assemble as long any $1 \times k$ or $k \times 1$ rectangle is left intact *anywhere* in the square, where $k = O(\log n)$. Thus, the information about the structure of the square is encoded uniformly throughout the square. Surprisingly, this construction uses only a constant number of tiles if we exclude the $O(\log n)$ tiles used to initiate the assembly. This is almost Kolmogorov-optimum [12]. Our construction involves a resolution loss of $O(\log n)$, i.e., we obtain a square of $n \times n$ blocks, where each block is of size $O(\log n) \times O(\log n)$ tiles. Each side of each block maintains two strings, one representing the i-th coordinate and the other representing the j-th coordinate. Thus, each side of each block encodes the position of the block. Our tile system "weaves" a horizontal and a vertical instance of our self-healing counter through these blocks; this allows us to conclude that the entire square would be self-healing if each block had the self-healing property. To complete the construction, we design a string-copy block that can heal from any of its four sides.

The same ideas can be used to obtain self-healing tile systems for assembling arbitrary Turing computable shapes; the resolution loss as well as the size of the configurations from which the assembly can recover depends on the Turing complexity of the shape and the running time of the corresponding Turing machine. A similar resolution loss is inherent in assembling Turing-computable shapes even without self-healing [19].

1.2 Simulations

We have tried to give intuition and draw diagrams where appropriate. However, self-assembly is a very visual and dynamic process and being able to see some of these constructions in simulation is an important aid to understanding. We have created an account titled "demoheal" on the computer rabi.stanford.edu, with a password "stoc2007"; readers and reviewers are welcome to ssh in and run the script "demoheal" from the home directory. A fast connection and an X server are required for this to work. The script will first run a demonstration of a self-healing Chinese remaindering counter. The second demonstration will be of a square that is bounded on two sides by self-healing Chinese remainder counters, and the square itself is self-healing once the counters are in place. This does not correspond to the strongest result we have obtained; however, in the words of another colleague, these tiles appear to almost possess "purpose and intention". Several tile sets are available in the same directory, and the simulator xgrow is available from Erik Winfree's web-page.

Literature relevant to self-assembly has vastly grown over the last several years, and a comprehensive survey is beyond the scope of this paper. But it is important to note that there are many interesting alternatives to the tiling based model (eg. the recent single stranded viral assemblies [13]) which also deserve theoretical attention.

2 Definitions and Models

2.1 The Tile Assembly Model

In this section we introduce the combinatorial abstraction of tile assembly that we will use for the remainder of the paper. This model is essentially the one proposed by Rothemund and Winfree [12] with the addition of the possibility of tiles falling off to facilitate discussion of self-healing. Informally, a tile system consists of a set of squares with a glue on each side. Squares will want to attach to each other if their adjacent glues match. An assembly starts with a seed tile and proceeds to add tiles one by one using this rule.

Formally, let Σ be a set of glues containing a distinguished glue *null*. A tile t is defined by its north, east, south, and west glues, denoted $\sigma_N(t)$, $\sigma_E(t)$, $\sigma_S(t)$, and $\sigma_W(t)$ respectively, drawn from Σ. We define a *tile system* as a tuple $< T, s, g, \tau >$. Here, T is a set of tiles, s is a distinguished seed tile, g is the *glue function* from $\Sigma \times \Sigma$ to the non-negative integers, and τ is the *temperature*, a positive integer. We assume $g(x, y) = 0$ for $x \neq y$ (glues only attach to themselves), and that $g(null, null) = 0$ (*null* is inert).

A *configuration* for a tile system is a map from $\mathbb{Z} \times \mathbb{Z}$ to $T \cup \{empty\}$. Let C and D be two configurations such that C matches D except at (x, y) where C is *empty* and D is some tile t. t is *attachable* to C if the sum of its glue functions with the surrounding tiles is at least the temperature: $g(\sigma_N(C(x, y-1)), \sigma_S(t)) + g(\sigma_E(C(x-1, y)), \sigma_W(t)) + g(\sigma_S(C(x, y+1)), \sigma_N(t)) + g(\sigma_W(C(x+1, y)), \sigma_E(t)) \geq \tau$. If this is the case we write $C \rightarrow_T D$. Define a sequence (possibly infinite) of configurations $\{C_i\}$ to be an *assembly sequence* if $C_i \rightarrow_T C_{i+1}$. If $\{C_i\}$ is finite the limit of such a sequence is clearly just the last element of $\{C_i\}$. Suppose $\{C_i\}$ is infinite. Each point in $\mathbb{Z} \times \mathbb{Z}$ is either *empty* for all C_i or is set to some non-*empty* value and is fixed from that point on. Thus, the limit of a coordinate (x, y) in an assembly sequence is well-defined: *empty* if (x, y) is always *empty* or the first value it changes to otherwise. We can then define the limit of an infinite assembly sequence to be the pointwise limit for all (x, y). We say D is *derivable* from C (denoted $C \rightsquigarrow D$) if there is an assembly sequence beginning at C whose limit is D. The set of *reachable* configurations is the set of configurations derivable from the configuration that has s at $(0, 0)$ and *empty* elsewhere. A configuration is *terminal* if it is reachable and no other states are derivable from it. It will also be useful to define a *mismatch* in a tile configuration to be a pair of adjacent positions whose touching glues are different and both not *null* (e.g. if $\sigma_N C(x, y) \neq \sigma_S C(x, y+1)$).

With these definitions in place we can formally define self-healing.

2.2 Self-healing

In this section, we give a definition of self-healing, and provide a criterion that allows us to verify a system has that property. We say $C \preceq D$ if C is the same as D at all locations where C is non-empty. This property corresponds to C being producible from D as a result of a series of tile removals or to C being a sub-assembly of D. This notion is useful in describing what happens when tiles fall off from a self-healing system. Intuitively, if a self-healing tile system reaches the desired configuration it should get back to that configuration after tiles have fallen off. We might hope we could define a system to be self-healing if $C \preceq D$ implies $C \rightsquigarrow D$ for all reachable D. Such a definition is too strong, as it may be impossible to progress from C if too much has fallen off. In the extreme case, nothing could possibly attach if C were the empty configuration. A more reasonable definition of self-healing is that a configuration will eventually be recovered if some small configuration of tiles with enough information still remains. Our definition formalizes this intuition.

Definition 1. *A tile system is self-healing from a configuration B if for any reachable configuration D, $B \preceq C \preceq D$ implies (1) $C \rightsquigarrow D$ and (2) $C \rightsquigarrow E$ implies $D(x, y) = E(x, y)$ wherever D and E are both not empty.*

A system is self-healing from a set of configurations $\mathcal{B}$ if it is self-healing for all $B \in \mathcal{B}$. The definition from [25] is close to the one we present but with two differences. First, that definition is restricted to being self-healing from configurations containing the seed. Second, it requires that the configuration recovered from be connected.

Essentially, this definition promises two things. First, there is a way to recover any tiles that have fallen off. Second, no tiles different from those we already had will ever attach. Also, we know that adding a tile can only make a tile unattachable at a location by taking up that location in aTAM. Combining the above properties, we can say that this definition promises that all lost tiles will be recovered "eventually" in the sense that there will always be a derivable configuration that has them all. Thus, a self-healing system is one that will recover from any loss of tiles that leaves one of its special configurations.

This notion of self-healing can be thought of as consisting of two parts; no incorrect tiles should be able to attach after tiles are lost and the assembly always has a way to make progress toward recovery. We call these two properties *immutability* and *progressiveness*.

Definition 2. *A tile system is immutable if for all reachable configurations D, $C \preceq D$ and $C \rightsquigarrow E$ imply $D(x, y) = E(x, y)$ when $D(x, y)$ and $E(x, y)$ are both not empty.*

Definition 3. *A tile system is progressive for a configuration B if for all reachable D, when $B \preceq C \preceq D$ there is E such that $C \rightsquigarrow E$ and E is not empty wherever D is not empty.*

The definition of immutability says that if a tile attaches at a position it is the only tile that can ever be attached at that position again. In the case where a tile system can produce a terminal configuration there is a criterion which is easy to check.

Lemma 1. *A tile system is immutable if for all pairs of tiles $s \neq t$ $\sum_{I \in \{N,S,E,W\}} g(\sigma_I(s), \sigma_I(t)) < \tau$, and the tile system has a terminal configuration with no mismatches.*

Proof. Assume $< T, s, g, \tau >$ meets the criteria described above but is not immutable. Let C be a terminal assembly with no mismatches. Consider any assembly sequence in our tile system starting with the seed. Let $D' \rightarrow_T D$ be the first transition such that $D(x, y) \neq C(x, y)$ and $D(x, y) \neq \mathit{empty}$ at some (x, y). All other tiles in D are the same as in C, so $C(x, y) \neq \mathit{empty}$ or C would not be terminal. $D(x, y)$ must have glue strength with adjacent tiles at least τ to attach. C has no mis-matches in it, so $C(x, y)$ must have the same glues as $D(x, y)$ on those sides. But then $C(x, y)$ and $D(x, y)$ have common glues whose strength add up to at least τ, which is a contradiction. □

In fact, suppose a self-healing system has a terminal configuration. There can be only one terminal configuration since otherwise we could form one terminal configuration, have everything but the seed fall off, and grow a different terminal configuration to contradict self-healing. Furthermore, if the terminal configuration is finite there can be no infinite chains of configurations since no tile can attach outside the terminal configuration using only tiles from inside the terminal configuration. Thus, in the case where a finite terminal configuration exists all growth chains lead to it, and the system is self-healing from the configuration consisting only of the seed.

3 A Self-healing Finite Counter

We now turn to our first self-healing system, a finite counter. Finite counters have been studied heavily [12, 1, 9, 3] and are useful in more complicated constructions. Thus, we begin by producing a self-healing system that grows to a fixed length (using some comparatively small width and number of tiles) and then stops growing. We will move to more complicated systems in the following sections. The most commonly described type of counters are binary counters, in which each row of the counter stores a binary number by labeling its tiles as ones and zeroes and incrementing this number for the next row. The binary counters normally described are not self-healing since they are not immutable: the wrong tile may attach if growth proceeds from a direction other than the normal one.

Instead of trying to modify a binary counter, we use a different type of counter that is already nearly self-healing, the Chinese remainder counter presented in [6]. The counter consists of k columns and operates at temperature 3. Column i is associated with an integer p_i such that p_i and p_j are relatively prime for $i \neq j$. For example, the p_i could be distinct primes. Column i has tiles labeled $0_i \ldots (p_i - 1)_i$. The south row of the counter is $0_1 \ldots 0_k$ from east to west. The glue on the north of x_i is the same as the glue on the south of $(x+1)_i$. These glues are all strength 3 except for glues between $(p_i - 1)_i$ and $(0_i)_i$ which have strength 2. The west glue of x_i matches the east glue of y_{i-1} for all x and y and has strength 1. Each column of the counter can grow by itself except from $(p_i - 1)$ to 0_i. The 0_i tile can be attached if either of the adjacent columns has reached that row though. Thus, new rows will continue to assemble until all columns are simultaneously at the $(p_i - 1)$ to 0 transition. By the Chinese remainder theorem, this happens at row $\prod p_i$. Figure 1 demonstrates the Chinese remainder counter for $k = 2$, $p_1 = 2$ and $p_2 = 3$.

It is easy to see the basic Chinese remainder counter is already immutable. It has a reachable terminal structure with no mismatches where each column is 0 in the bottom row and $p - 1$ in the top row. Each tile has a unique glue on the north and south, and all the glues on the sides are single strength. Thus, distinct tiles can share at most strength 2. The system is also progressive from $\mathcal{B}$, the set of all configurations in which all occupied positions have the correct tile and there is a connected path from the east to the west of the counter. To see this, observe that a tile can always be attached whenever there is a tile either north or south of it and to the east or west. Thus, a sequence of additions can derive a configuration where all rows that have a tile are completely full in the connected region around the path. Then normal assembly can proceed up and down from completed rows as long as all columns at the top do not have the $p_i - 1$ simultaneously and all tiles at the bottom do not have the 0 tile. Thus, if we finished constructing a Chinese remainder counter it would be able to reconstruct itself from any damage that preserved a path across.

The problem is the Chinese remainder counter as we described above is not well-specified. We have not specified a seed tile, and in fact an initial row of tiles will not assemble by itself from any single tile. Unless we are willing to just assume the 0 row is given to us preformed(which may be reasonable in some cases) we need to introduce more tiles to get the system started. Our approach is to include a $(k+1) \times 2$ L-shape of unique tiles. Tiles 1 to k of the bottom row have the strength 2 $(p_i - 1)_i$ to 0_i glue on the north of them and unique strength three glues between adjacent tiles. Their south glues are *null*. The corner tile has a strength 3 glue to the k-th tile of the row and a strength 3 glue to the tile of the short leg. The short leg tile has a strength 2 glue which will match 0_k's east glue. Thus, the L will form by itself uniquely from any of its tiles, and once it does a row of 0's will form on top of it. It is easy to see the new system is still immutable, but it is no longer progressive. Once the L is removed it will not be able to attach again. We can fix this problem as well though by creating another Chinese remainder counter that counts in the opposite direction next to the first one(see figure). This way if both counters have formed the middle tile common to the two L's can attach and the two seed L's can reassemble. If we define $\mathcal{B}_2$ to be the set of configurations that have a path from west to east in both counters our system is self-healing from $\mathcal{B}_2$. Immutability is easy to check, as the two counters and the L's each use different glues internally. Progressiveness follows from the fact that both the full counters can form from an element of $\mathcal{B}_2$, and the two L's can form once the two counters are completed. Figure 2 gives a schematic for the full system.

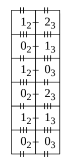


Figure 1: The full assembly of the Chinese remainder counter for 2 and 3.

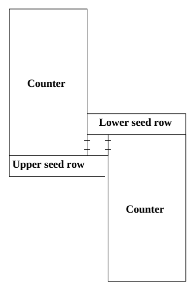


Figure 2: How the seed interacts with the two Chinese remainder counters.

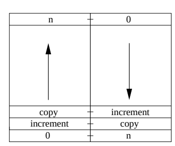


Figure 3: The CRT counter modified to produce an arbitrary length.

The counter described above would only be efficient for lengths that are products of several small relatively prime factors(i.e. smooth numbers [21]). We will now improve on the counter to get around this restriction, using a method we call reverse augmentation. First, we modify the original Chinese remainder counter. After each increment, we add a copy row with strength 2 glues from below and strength one glues between columns. This copy row will form completely given one element and the row below it but cannot start without help at temperature 3. Suppose we had another Chinese remainder counter on the right of our current one counting south instead of north with strength one glues on the left of its increment rows. Now the copy row of the original counter can form provided the new counter has formed to that row. This method gives us a second way to stop the counter.

To produce a counter that grows to n using any set of primes whose product exceeds n we can proceed as follows. Place two Chinese remainder counters next to each other with the modifications described above. Seed the first row of the left counter with zeros and have it count up. Seed the right counter with $n \mod p_i$ and have it count down. The right counter will not be able to grow below the seed row since the left counter will not grow beyond that point. Similarly, the left counter will not grow higher than the nth row because at that point the right counter will reach 0 and stop. Thus, the length of this counter will be exactly n. The final result is a self-healing counter of any length. Figure 3 presents a schematic for this counter. Placing it in the construction of figure 2 will produce a self-healing counter of any length.

The above system operates at temperature 3 for ease of discussion. It is easy to modify the system to operate at temperature 2 using the method described in [6].

4 The self-healing infinite counter

In this section, we will present a tile system that simulates a self-healing binary counter. The counter can count from 1 to infinity and is progressive from any connected piece of the assembly that touches both sides of the counter.

We will start by presenting a temperature two tile system that simulates an infinite binary counter that can count in both directions but is not self-healing. The structure of the final assembly is shown in figure 4(a). The counter counts from south to north and the least significant bit is on the west. A list of every tile in this tile system is in figure 4(b). The binary number stored on the tiles increases/decreases by one for every two rows. For the increment operation, tiles on row A attach from west to east. The glues on the north sides of these tiles code the same number with the glues on the south sides. The only difference is that the least significant zero is marked by a special glue. Tiles on Row B attach from east to west, copying the bits encoded on the glues from the south sides to the north sides before the tile with the special glue attaches and flip every bit afterwards. The decrement operation is similar. Tiles on row B attach from west to east, search for the least significant 1, output a special glue on the south side, and flip all the less significant bits. Row A copies all the bits from the glues on the north sides to the glues on the south sides. When the number on row A is $111\ldots1$, the system tries to search for the least significant 0 but fails. In this case, a special tile will be able to attach next to the most significant bit, increasing the number of bits by one. Since there are no strength zero glues in the interior of this counter, from any connected piece that touches both the east and west end of the counter, there is at least one row in which all tiles can attach. Then all the tiles will be able to attach on normal counter operations.

The system described above is not immutable because some of the tiles share the same glues on opposite sides. For example, the second and third tile for row A share the two strength 1 glues indicated by "copy". But this system has two nice properties, so we can convert it to an immutable system: *Property 1:* Each tile has at most two sets of input edges during the normal counter operations. When the assembly grows from south to north (increment operation), a tile in row A needs to attach only when the tiles to its west and south are in place; a tile in row B needs to attach only when the tiles to its east and south are in place. Similar properties hold when the assembly grows from north to south. *Property 2:* For every tile in row A, there is no other tile with the same west and south glues or with the same east and north glues; similar things hold for row B.

Based on the above two properties, we can convert this system into an immutable system by replacing each tile in the original system by a 2×2 block of tiles. The idea of this conversion is similar to the self-healing conversion described in [25]. As shown in figure 4(c) for the 2×2 blocks on row A and row B, all the interior glues have strength two and are unique to this block. The original glues on the east-west direction are replaced by two glues of the same strength; the original glues on the north-south sides are replaced by one glue of the same strength and one inert edge. In row A, if both the blocks to its west and south are both in place, then the tile on the southwest corner can attach and the whole block can attach according to that tile after this change. Similar properties holds for all normal counter operations. Hence, property 1 described above ensures that the correct tiles can always attach in the new system, while the 2×2 blocks cannot attach from any other direction.

We now check the immutability and progressiveness of this new system. To check whether the new system is immutable, first observe that there are no mismatches in the assembly. Hence, we only need to show no two tiles share glues with total strength two. We do not need to worry about the glues interior to each block since they are unique to that block and only appear on the opposite sides of two tiles in the whole system. So we only need to consider the tiles on the southwest and northeast corners of blocks in row A and tiles on the northwest and southeast corners of blocks in row B. By property 2 described above, the system must be immutable. Progressiveness is easy to check since the correct operation ensures that the counter can grow both towards north and south.

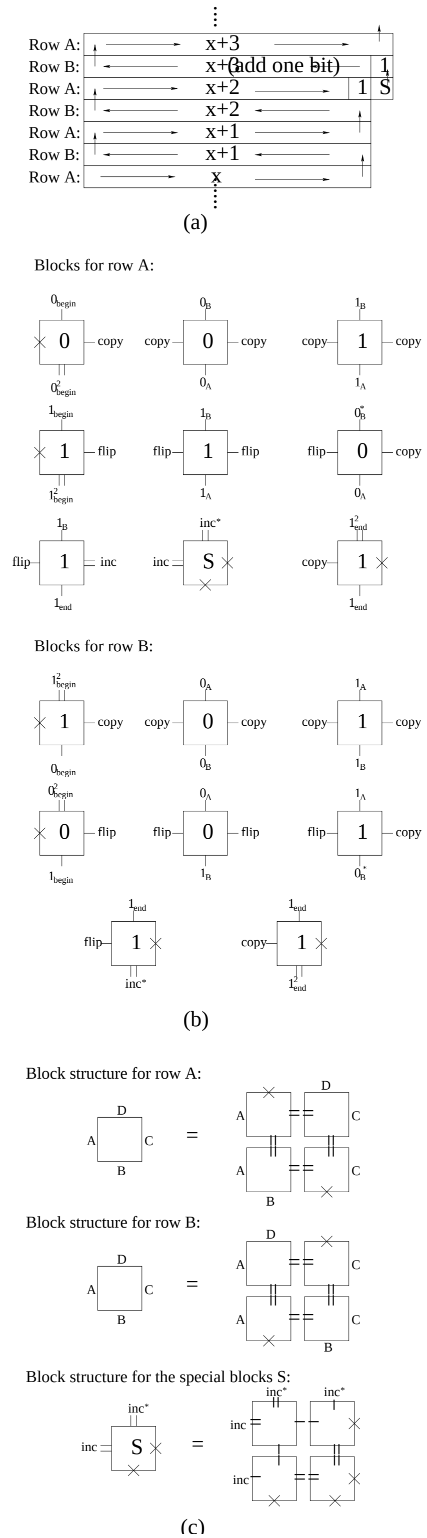


Figure 4: (a) System illustration. (b) A list of all tiles in this system. (c) Every tile in the system is replaced by a two by two block to achieve immutability.

Furthermore, if we remove the special block marked by "S" in figure 4 and use the L-shaped seed structure described in the Chinese remainder counter, the system becomes a binary counter that counts from 0 to 2^n. We can obtain a self-healing binary counter that counts from 0 to n for arbitrary n using the reverse augmentation method used in the Chinese remainder counter section.

5 Self-Healing Square System

In this section, we will show a self-healing system that construct a square with $n \times n$ blocks. For ease of exposition, we assume that n is a power of 2. A square with an arbitrary number of blocks can be constructed using the reverse augmentation method used in the Chinese remainder counter section. Each block consists of $(8 \log n + 16) \times (8 \log n + 16)$ tiles. We refer to this size as the resolution lost, r. The square is progressive from any $1 \times 2r$ or $2r \times 1$ rectangle. Furthermore, besides the $O(\log n)$ tiles that encodes the size of the square and are used as the "seed", the system only uses a constant number of tiles. We also generalize this system to produce any given finite connected shape with a constant number of tiles plus $O(K + \log n)$ tiles to initialize, where K is the size of the description of a Turing machine that determines whether a specific location (i, j) is empty and n is the length of the shape. However, there will be a resolution loss of $O(T)$, where T is the time required for a universal Turing machine to simulate the given Turing machine.

5.1 High Level Design

To make the self-healing square, we designed a special structure that we call "string copy" blocks. String copy blocks encode the same information on all four sides and are progressive from any connected piece from north to south or from east to west. The structure of the string copy block is described in section 5.2, where we prove immutability and progressiveness of the block. The structure of the self healing square consists of $n \times n$ string copy blocks of size $(8 \log n + 12) \times (8 \log n + 12)$. The string copy block at location (x, y) has the information (x, y) encoded on its four sides. The string copy blocks are connected by fixed width binary counters described in section 4. Figure 5 illustrates how these two different systems connect. Each counter just increments/decrements the corresponding coordinate. On the boundary of the square, the counter reaches 0 or n and stops growing. We use the same L-shaped seed described in the Chinese remainder counter section to initiate the counter at $(0, 0)$.

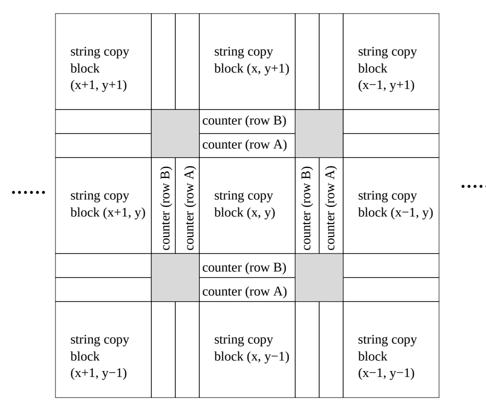


Figure 5: The structure near the (i, j)-th string copy block of the self-healing square. All the dotted regions are filled with a single type of filling tile that has glues of strength one on all four sides are matches with the corresponding sides of the binary counter.

Theoremc 1. *The square described above is self-healing from any $2r \times 1$ or $1 \times 2r$ rectangle.*

Proof. First we show the square is progressive from any $2r \times 1$ rectangle: any such rectangle inside the square must contain a path from east to west in one string copy block or a complete row of a binary counter.

By the progressiveness of string copy blocks (as proved later in lemma 3) and binary counters, at least one string copy block will be able to attach. Afterwards, the four binary counters adjacent to this block will be able to attach, and this enables the attachment of the four string copy blocks they connect to, and so on. Eventually, the whole square will be able to attach except the $O(\log n)$ L-shaped initial tiles. Those initial tiles will be lost, but we can use the same technique described in the infinite counter section to make the initial tiles recoverable.

Further, a tile in the string copy block a any tile in the binary counter do not share glues on the same side. The immutability of the string copy block is proved in lemma 2, and the immutability of the binary counter is proved in section 4. Hence, no two tiles share glues of total strength two. From lemma 1, immutability of the square follows, since the square is a terminal shape with no mismatches inside. ☐

5.2 The "String Copy" Assembly

In this subsection, we will describe the temperature 2 "string copy" block used in section 5.1. The string copy system maintains a square with four identical strings on its four sides. This square is progressive from any path that goes from north to south or from east to west. The strings are of the form $s_1 s_2 \dots s_k$, where symbol s_1 is a special starting symbol, symbol s_k is a special ending symbol and all the other s_is are either 0, 1, or a special separator symbol. The special symbol is used to separate coordinates i and j when this block is used in the high level design.

As shown in figure 6(a), the string copy block consists of $k \times k$ gadgets, each of size 4×4. The basic idea is to let gadget $(i,\ j)$ encode the information of symbol s_i and s_j. Hence, if you look at any side of this block, each of the k gadgets will correspond to one symbol in the string. For each gadget (i, i), the structure of the gadget ensures that the symbol encoded on all four sides must be the same, so the string encoded on all four sides are identical. The structure of the gadget at location $(i, j), 1 < i, j < k$ is described in figure 6(b). Each white tile has nine duplicates, one for each combination of symbols s_i and s_j. (Recall that s_i and s_j could be 0, 1, or the special symbol.) For striped tiles, there are three duplicates, one for each value of s_i if $i > j$ and one for each value of s_j if $i \leq j$. Every glue between the white tiles (i.e. marked by "B_*") also has nine duplicates corresponding to each white tile. Every glue marked by "I_*" has three duplicates corresponding to different values of s_i and every glue marked by "J_*" has three duplicates corresponding to different values of s_j. Gadgets on the boundary of the string copy block is illustrated in fig 7. They have the same structure as the gadgets in the interior of the block except the glues on the boundary are modified to match with the glues for the binary counter.

Lemma 2. *The string copy block is immutable.*

Proof. First, it is easy to verify that any two different tiles in figure 6(b) only share at most one glue of strength one no matter which duplicate we choose. Second, if two duplicates of a tile share some glues with total strength at least two, then they must either share one glue marked by "B_*" or two glues marked by "I_*" and "J_*" respectively. Hence these two tiles must correspond to the same set of s_i and s_j and must be the same tile. Hence, we know that no two tiles share glues with total strength two in the string copy system. Since there is no mismatch in the string copy block, it is immutable. ☐

Lemma 3. *The string copy block is progressive from any connected piece from north to south or from east to west.*

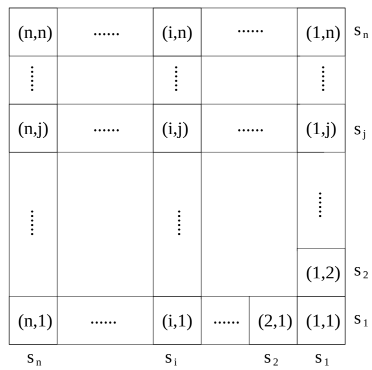
Proof. We first show that the string copy block can assemble as long as we have at least one side of the block remaining. If the east side of square remains, then the whole $(1, 1)$ gadget will attach followed by $(1, 2)$ gadget, $(1, 3)$ gadget, and so on. Also, after gadget $(1, 2)$ attaches, all tiles in gadget $(2, 2)$ will be able to attach, which will let gadget $(2, i)$ attach for every i. The remainder of the block attach in the same way. A similar situation happens when any one of the other sides of the string copy square remains.

This shows that if we cut the string copy square into two pieces with a path from north to south or from east to west, the square can assemble from any of those two pieces. Therefore the square can assemble if any path from north to south or from east to west remains. ☐

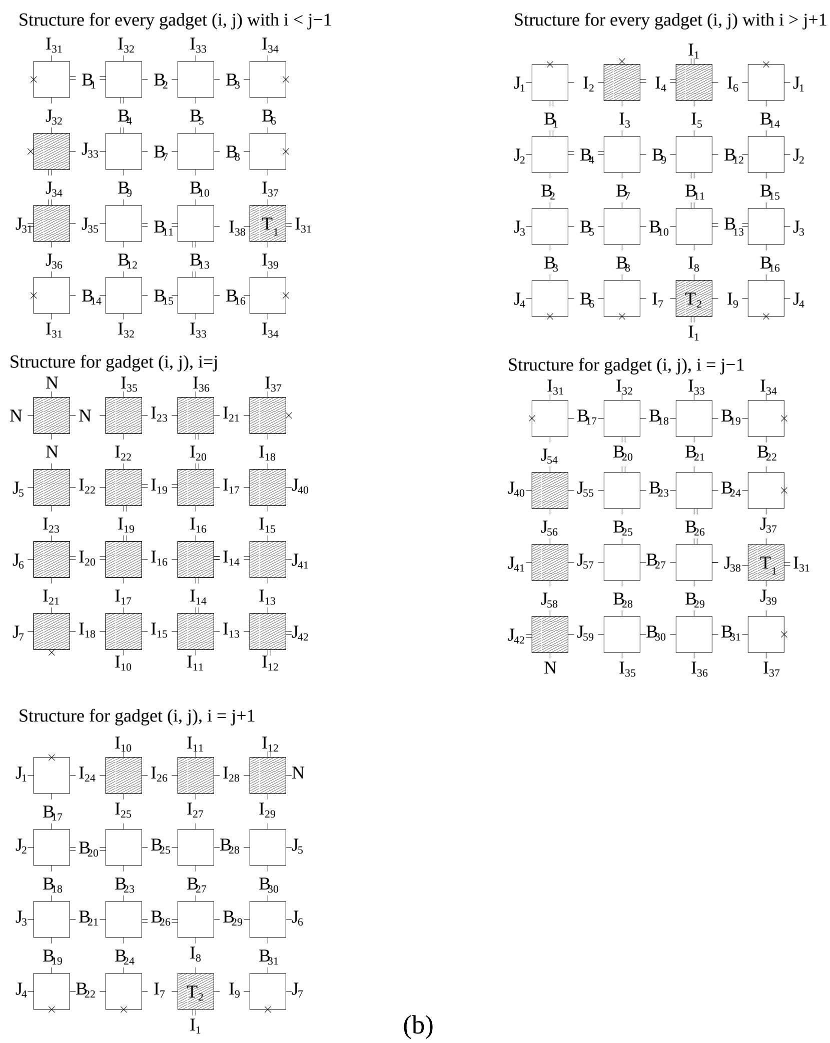
5.3 Generalization

In this section, we sketch how to make an arbitrary connected shape self-healing. Before describing the complete system, we need to describe two essential components. First, we make a "double string copy block" by combining four string copy blocks into a 2×2 square. From the property of the string copy block, we know that this block encodes two copies of the same symbols $s_1, s_2, \ldots, s_k$ on its four sides and can assemble as long as one half of one side of the block is available. Second, we combine the Turing machine designed in [19] and the self-healing scheme in [25] to get a "self-healing Turing machine". This system only assembles from one specific input direction and the size of the system is $O(T)$ where T is the time required for the computation.

Given any finite connected shape, we use a double string copy block for each location (x, y). The string on each side of the double string copy block contains the information x, y and the K bits of information required to define the Turing machine that determines whether a specific location is empty. Two adjacent double string copy blocks are connected by two self-healing Turing machines, one for each direction. If the double string copy block at (i, j) is attached, then these Turing machines will allow all adjacent double string copy blocks to attach. This process will continue until all the blocks attach. However, the resolution loss is now $O(T)$ where T is the computation time required for the universal Turing machine to simulate the Turing machine described by the K bits of information and the L-shaped structure to initialize the assembly must have size $O(K + \log n)$, where n is the length of the shape.



(a)



(b)

Figure 6: (a) There are $k \times k$ gadgets in a square. Each of the small square in this figure denotes a 4×4 gadget. (b) The structure of the 4×4 gadgets.

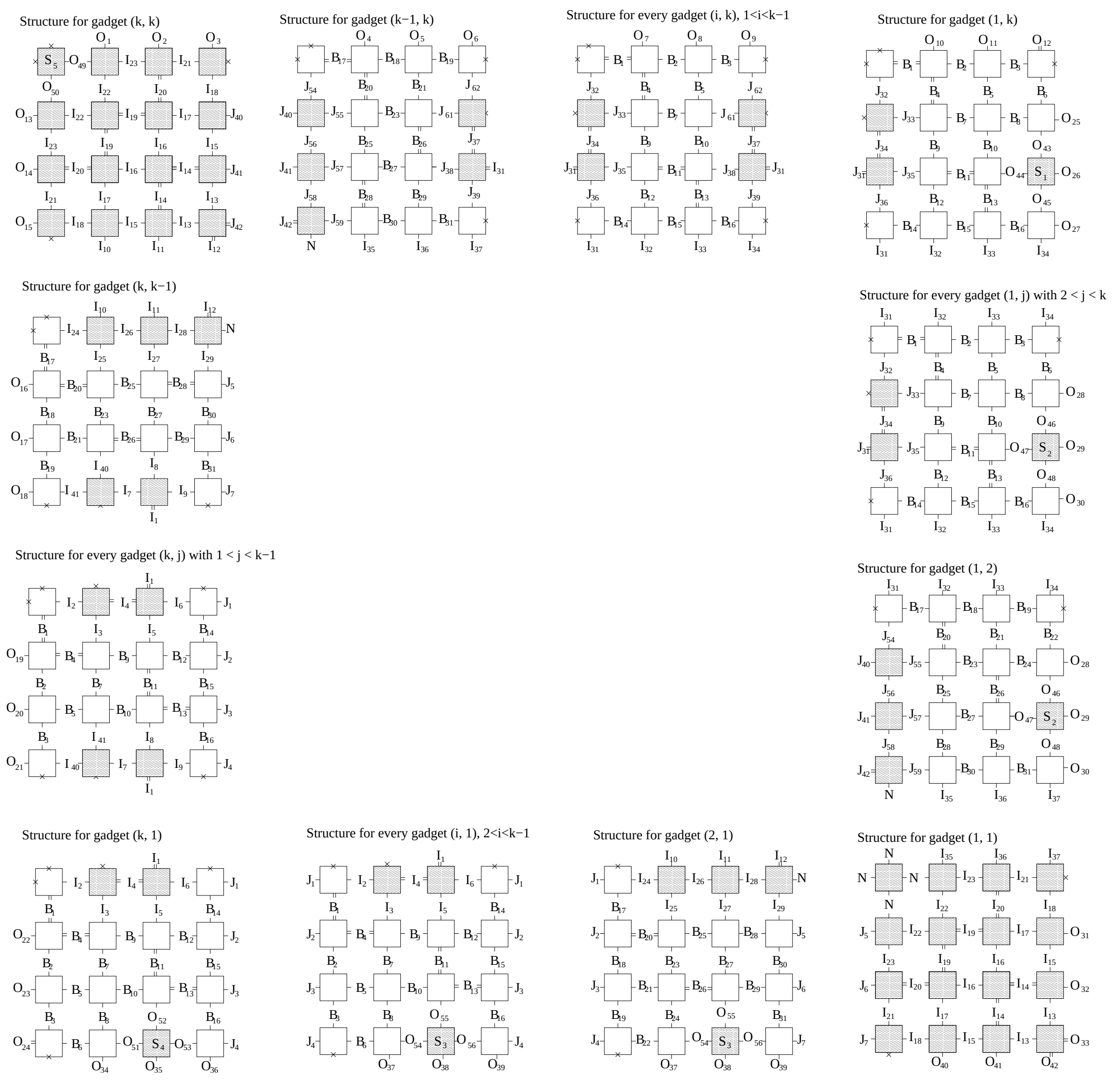


Figure 7: The structure of the gadgets on the boundary of the string copy block.

References

[1] L. Adleman, Q. Cheng, A. Goel, and M.-D. Huang. Running time and program size for self-assembled squares. *Proceedings of the 33rd ACM Symposium on Theory of Computing*, pages 740–748, July 2001.

[2] L. Adleman, Q. Cheng, A. Goel, M.-D. Huang, and H. Wasserman. Linear self-assemblies: Equilibria, entropy, and convergence rates. *To appear in proceedings of the sixth Int'l Conf. on Difference and Differential Equations (ICDEA)*, 2001.

[3] R. Barish, P. Rothemund, and E. Winfree. Two computational primitives for algorithmic self-assembly: Copying and counting. *Nano Letters*, pages 2586–2592, 2005.

[4] Y. Baryshnikov, E. Coffman, and P. Momcilovic. DNA-based computation times. In *Proceedings of the Tenth International Meeting on DNA Based Computers*. Milano, Italy, June 2004.

[5] Y. Baryshnikov, E. Coffman, and P. Winkler. Linear self-assembly and random disjoint edge selection. Technical report, Electrical Engineering Department, Columbia University, 2004.

[6] H. Chen, Q. Cheng, A. Goel, M.-D. Huang, and P. Moisset de espanes. Invadable self-assembly: Combining robustness with efficiency. *Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms*, pages 883–92, Jan 2004.

[7] H. Chen and A. Goel. Error free self-assembly using error prone tiles. *Lecture Notes in Computer Science (Proceedings of the 10th International Meeting on DNA Based Computers, Jun 2004)*, 3384:62–75, 2005.

[8] H. Chen, A. Goel, R. Schulman, and E. Winfree. Error correction for DNA self-assembly: Preventing facet nucleation. *To appear in the proceedings of the 11th International Meeting on DNA Based Computers (short abstract)*, June 2005.

[9] Q. Cheng, A. Goel, and P. Moisset. Optimal self-assembly of counters at temperature two. *Proceedings of the first Conference on Foundations of nanoscience: self-assembled architectures and devices*, Apr 2004.

[10] M. Lagoudakis and T. LaBean. 2d DNA self-assembly for satisfiability. In *Proceedings of the 5th DIMACS Workshop on DNA Based Computers in DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, volume 54. MIT: Cambridge, 1999.

[11] J. H. Reif, S. Sahu, and P. Yin. Compact error-resilient computational DNA tiling assemblies. In *Proceedings of the Tenth International Meeting on DNA Based Computers*. Milano, Italy, June 2004.

[12] P. Rothemund and E. Winfree. The program-size complexity of self-assembled squares (extended abstract). In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 459–468. ACM Press, 2000.

[13] P. W. K. Rothemund. Folding dna to create nanoscale shapes and patterns. *Nature*, 440:297–302, 2006.

[14] R. Schulman and E. Winfree. Programmable control of nucleation for algorithmic self-assembly. *Proceedings of the tenth International Meeting on DNA Based Computers*, pages 319–328, 2004.

[15] W. Sherman and N. Seeman. A precisely controlled DNA biped walking device. *Nano Letters*, 4:1203–1207, 2004.

[16] J.-S. Shin and N. Pierce. A synthetic DNA walker for molecular transport. *J. Am. Chem. Soc.*, 126:10834–10835, 2004.

[17] J.-S. Shin and N.A. Pierce. A synthetic DNA walker for molecular transport. *Journal of American Chemicals Society*, (126):10834–10835, 2004.

[18] D. Soloveichik and E. Winfree. Complexity of compact proofreading for self-assembled patterns. *Proceedings of the eleventh International Meeting on DNA Based Computers*, pages 305–324, 2005.

[19] D. Soloveichik and E. Winfree. Complexity of self-assembled shapes. *DNA Computers 10*, 3384:344–354, 2005.

[20] M. Stojanovic and D. Stefanovic. A deoxyribozyme-based molecular automaton. *Nature Biotechnology*, 21(9):1069–1074, Sep 2003.

[21] G. Tenenbaum. *Introduction to Analytic and Probabilistic Number Theory.* 1995.

[22] Y. Tian, Y. He, Y. Chen, P. Yin, and C. Mao. A DNAzyme that walks processively and autonomously along a one-dimensional track. *Angewandte Chemie*, 117:4429–4432, 2005.

[23] H. Wang. Proving theorems by pattern recognition ii. Bell Systems Technical Journal, 1961. 40:1-42.

[24] E. Winfree. *Algorithmic Self-Assembly of DNA.* PhD thesis, California Institute of Technology, Pasadena, 1998.

[25] E. Winfree. Self-healing tile sets. *Nanotechnology: Science and Computation*, pages 55–78, 2006.

[26] E. Winfree and R. Bekbolatov. Proofreading tile sets: Error correction for algorithmic self-assembly. *Ninth international meeting on DNA based computers*, 2003.

[27] E. Winfree, F. Liu, L. Wenzler, and N. Seeman. Design and self-assembly of two-dimensional DNA crystals, 6 pages. *Nature*, (394):539–544, Aug 1998.

[28] E. Winfree, X. Yang, and N. Seeman. Universal computation via self-assembly of DNA: Some theory and experiments. In *Proceedings of the Second Annual Meeting on DNA Based Computers*. Princeton University, June 1996.

[29] P. Yin, H. Yan, X. G. Daniell, A. J. Turberfield, and J. H. Reif. A unidirectional DNA walker moving autonomously along a linear track. *Angew. Chem. Int. Ed.*, (43):4906–4911, Sep 2004.

[30] B. Yurke, A. Turberfield, A. Mills Jr, F. Simmel, and J. Neumann. A DNA-fuelled molecular machine made of DNA. *Nature*, (406):605–608, Aug 2000.