DIMACS Series in Discrete Mathematics
and Theoretical Computer Science

Experimental Progress in Computation by Self-Assembly of DNA Tilings

Thomas H. LaBean, Erik Winfree, and John H. Reif

Abstract. Approaches to DNA-based computing by self-assembly require the use of DNA nanostructures, called tiles, that have efficient chemistries, expressive computational power, and convenient input and output (I/O) mechanisms. We have designed two new classes of DNA tiles, TAO and TAE, both of which contain three double-helices linked by strand exchange. Structural analysis of a TAO molecule has shown that the molecule assembles efficiently from its four component strands. Here we demonstrate a novel method for I/O whereby multiple tiles assemble around a single-stranded (input) scaffold strand. Computation by tiling theoretically results in the formation of structures that contain single-stranded (output) reported strands, which can then be isolated for subsequent steps of computation if necessary. We illustrate the advantages of TAO and TAE designs by detailing two examples of massively parallel arithmetic: construction of complete XOR and addition tables by linear assemblies of DNA tiles. The three helix structures provide flexibility for topological routing of strands in the computation, allowing the implementation of string tile models.

1. Introduction

In his seminal paper on molecular computation [**Adl94**], Adleman demonstrated a generate-and-separate algorithm for solving Directed Hamiltonian Path Problems (DHPP). In the first phase of his algorithm, a combinatorial library of DNA molecules representing potential solutions to the problem is generated by random hybridization and ligation of a small number of oligonucleotide species. In the second phase, a sequence of affinity separation and other laboratory procedures isolates the DNA molecules representing the actual solution; the sequence of this molecule, and thus the answer to the mathematical problem, can then be determined. However, the generate-and-separate strategy has two difficulties. First, for moderately large problems, a naïve generation strategy must generate an astronomical number of DNA strands to represent all potential solutions. Second, the number of laboratory procedures, each time consuming and error-prone, grows with the size of the problem.

The first difficulty can be avoided by interspersing the generate procedures and the separation procedures [**FB97, FB99**]. Simple examples of this approach include DNA implementations of dynamic programming [**BB98**] and breadth-first search [**MAS99, YS99**]. However, this approach usually comes at the cost of making the second difficulty more severe, by multiplying the number of procedures required.

Our work has focused on another approach: computation by self-assembly. Self-assembly is the spontaneous self-ordering of substructures into superstructures driven by annealing of, in this case, Watson-Crick base-pairing DNA sequences. Computation by self-assembly entails the building up of superstructures from starting units such that the assembly process itself performs the actual computation. Adleman made use of a simple form of computation by self-assembly in his original experiment [**Adl94**]: his algorithm does not blindly generate all possible sequences of vertices; instead, the oligonucleotide sequences and the logic of Watson-Crick complementarity guide the self-assembly processes so that only valid paths are generated. We have generalized this approach to two-dimensional (2D) self-assembly processes, where it can be seen that computation by self-assembly is Turing-universal [**Win96**]. The building blocks in these constructions are branched DNA complexes, which we call *DNA tiles*, consisting of several individual DNA oligonucleotides that associate with well-defined secondary structure. DNA tiles are designed to contain several short sections of unpaired, single-strand DNA (ssDNA) extending from the ends of selected helices that function as binding domains, which we call *pads*. Individual tiles interact by annealing with other specific tiles via their ssDNA pads to self-assemble into desired superstructures. The use of pads with complementary base sequences allows the neighbor relations of tiles in the final assembly to be intimately controlled; thus the only large-scale superstructures formed during assembly are those that encode valid mappings of input to output. Consequently, the second difficulty mentioned previously has been addressed: rather than implementing a DNA computing algorithm using a sequence of multiple laboratory procedures, our approach essentially uses only four: mixing the input oligonucleotides to form the DNA tiles, allowing the tiles to self-assemble into superstructures, ligating strands that have been co-localized, and then performing a single separation to identify the correct output.

We now give a brief description of the mathematical tiling and DNA construction techniques underlying our work.

1.0.1. *Computation by Tiling.* A class of domino tiling problems were defined by Wang [**Wan61**]. One is given a finite set of unit size square tiles, each of whose sides are labeled with symbols over a finite alphabet (the pads). Additional restrictions may include the initial placement of a subset of these tiles, and the dimensions of the region where tiles must be placed. The problem is to place the tiles, chosen with replacement and without rotation, to completely fill the given region so that each pair of abutting tiles have identical symbols on their contacting sides. Proofs of the undecidability of the infinite tiling problem rely on constructions wherein tiling patterns simulate single-tape Turing Machines [**Buc62, Wan63, Ber66, Rob71, Wan75**]. Other results include reductions of NP-complete problems to finite-size tiling problems [**LP81**]; tilings and patterns in general are surveyed in [**GS87**].

1.0.2. *DNA as a Construction Material.* Nano-fabrication of structures in DNA was pioneered by Seeman starting in the 1980s (e.g., see [**See82**]). Seeman and his students have built a multitude of ingenious constructions including: DNA branched junctions [**SCK89, WMKS91, DZS92**], knots [**SCD^{+}93**], Borromean rings [**MSS97**], a cube [**CS91**], and a truncated octahedron [**ZS94**] (reviewed in e.g.

[**See98**]). Recently, they have reported the construction from DNA of a nanomechanical device capable of controlled movement [**MSSS99**].

The most relevant constructs for our current purposes are the double-crossover (DX) complexes, which consist of two double-helices interlocked by exchange of oligonucleotide strands at two separate crossover points [**FS93**]. In DX complexes, a particular oligonucleotide strand base-pairs with its complement in one helix then crosses over to continue base-pairing with its complementary section on a second, adjacent helix; the multiple crossover points constrain the helix axes to be co-planar. Because DX complexes can be used as DNA tiles in self-assembly reactions, we will often refer to them as such.

DX complexes come in five varieties that differ from one another in the geometry of strand exchange and the topology of the strand paths through the tile. Anti-parallel crossovers cause a reversal in direction of strand propagation through the tile following exchange of strand to a new helix. We will primarily concern ourselves with tiles containing anti-parallel crossovers (DAO and DAE) rather than those with parallel crossovers (DPE, DPOW, DPON), because laboratory characterizations of parallel tiles showed them to be less stable structurally than anti-parallel tiles [**FS93**]. DX tiles provide up to four ssDNA pads for encoding associations with neighboring tiles. DX complexes have been shown to be sufficiently rigid to avoid formation of circular structures as seen with the more flexible dsDNA [**LYQS96**]. Recently, 2D lattices consisting of hundreds of thousands of DX units have been constructed and observed by atomic force microscopy [**WLWS98**], and DX complexes and lattices have been used successfully as substrate for enzymatic reactions including cleavage and ligation [**LSS99**].

1.0.3. *Parallelism, Speed and Tiling Depth.* The massive parallelism inherent in DNA-based computers has, since its inception, driven thinking in the field. In computation by self-assembly, parallelism reveals itself in many ways. Each superstructure may contain information representing a different calculation (*global parallelism*). Growth on each individual superstructure may occur at many locations simultaneously (*local parallelism*). Self-assembly may be restricted such that certain assembly reactions can proceed only after others have been completed (*serial self-assembly*). Alternatively, self-assembly reactions may be limited by no such restrictions (*free self-assembly*). As examples, BCA tiles utilize local parallelism and serial self-assembly [**Win96**]; DHHP tiles utilize both local and global parallism and serial self-assembly [**WYS98**]; and self-assembly of linear, hairpin, and branched DNA molecules to generate regular, bilinear, and context-free languages makes use of global parallelism and free self-assembly [**WYS98, Eng99**], as do the proposals of Jonoska et al [**JKS99, JKS98**].

It it expected, but unproven, that free self-assembly is faster, more robust, and less error-prone than serial self-assembly. Three measures assist discussion: the *depth* of a superstructure is the maximum number of self-assembly reactions experienced by any substructure (the depth of the graph of reaction events), and the *size* of a superstructure is simply the number of tiles it contains. A superstructure consisting of $n \times m$ tiles, where $n > m$, is said to have m *layers*. The depth (size) of a tiling system is the maximum depth (size) of any superstructure it generates. Likewise for the number of layers. For example, in a tiling system where $A + B \rightarrow AB, C + D \rightarrow CD, AB + CD \rightarrow ABCD$, we know that the depth is ≥ 2 and the size is ≥ 4. Again, although it needs further study, tiling systems with low depth, small size, and few layers are considered desirable, motivating the search for efficient computations performed by such systems. Tiling systems that produce only superstructures with k layers, for some constant k, are said to use *linear self-assembly.* As an example, the two tiling systems for addition discussed here produce $3 \times n$ and $1 \times n$ superstructures for n-bit sums and hence are linear, but known tiling systems for multiplication produce $n \times n$ for n-bit products [**WER99**], and hence are not linear. The BCA and DHHP tile systems also are not linear.

Reif [**Rei99**] proposed DNA self-assembly methods of linear size and small depth to solve a number of fundamental problems (e.g., prefix computation, permutation, integer addition and subtraction, multiplication by a constant number, finite state automata simulation, and string fingerprinting) that form the basis for the design of many parallel algorithms. Furthermore, Reif showed how these elementary operations can be combined to perform more complex computations, such as bitonic sorting and general circuit evaluation in $O(\log n)$ experimental steps. To increase the likelihood of successful assembly, Reif also proposed step-wise assembly, which controls the assembly in distinct stages.

A second approach for small-depth computations is to compress several tile layers into single tiles, so that the simplest form of linear self-assembly suffices. This approach works particularly well when the topology and routing of the strands in the DNA tiles is carefully considered, leading to the notion of *string tiles* [**WER99**]. The TAO and TAE tiles reported here are particularly useful as string tiles, and this approach will be discussed in Section 3.

1.0.4. *Previous Methods and New Improvements in I/O.* Input and output are critical to the practical use of DNA-based computing. Winfree [**Win96**] used the first and last layers of the assembly for input and output, respectively. Reif [**Rei99**] proposed the use of DNA nano-structures known as *frames* for input and output; this allows the input and output to be ssDNA strands rather than assemblies of tiles. Precedent for Reif's proposal for output can be found in techniques used to confirm experimental constructions of exotic 3-dimensional structures, wherein oligonucleotides were ligated together to form new, longer strands which, if present at the end of the construction process, reported on the success of formation of the desired structure (for example see [**LYQS96**]). We also make use of reporter strands, both to monitor proper assembly and to output the results of a computational assembly. Further, we advance a similar concept for input by describing and testing scaffold strands, which are long DNA strands capable of serving as nucleation points for assembly. Preformed, multimeric scaffold strands are added to the hybridization/annealing mixture in place of the monomeric oligo corresponding to the tile's reporter segment. Tiles assemble around the scaffold strand, automatically forming a chain of connected tiles which can subsequently be used as the input layer in a computational assembly (see Subsection 2.2 below).

2. TAO Tile and 2D Assemblies

Besides expanding the repertoire of DNA tiles, our specific goals in prototyping a new tile type were to provide four ssDNA pads per tile and simultaneously to allow a reporter strand to trace its way diagonally through the tile. This latter property is essential for several of the computational constructions we will consider later, wherein the diagonal strand, when ligated to its counterparts in neighboring tiles, produces a reporter strand that traces through a diagonal arrangement of tiles and records the values of the input and output strings from all the tiles.

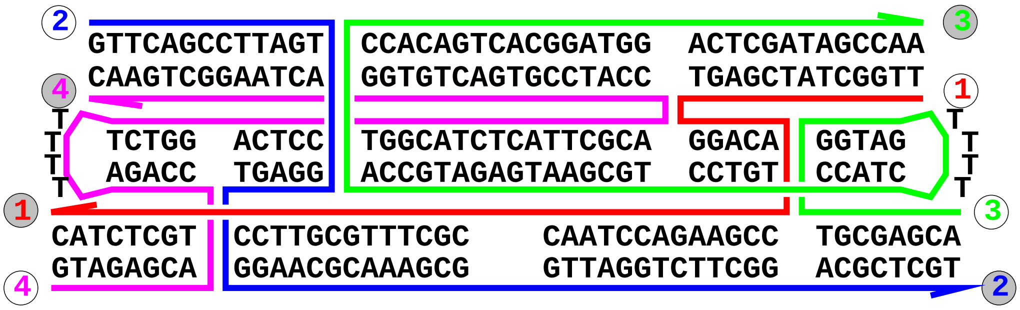

FIGURE 1. **Strand and Sequence Trace through TAO Tile.** The four oligonucleotides are numbered and labeled with white circles on their $5'$ ends and dark circles on the $3'$ ends. Base-pairing partners are shown one above the other for each of the three helices. The two `TTTT` segments are hairpins on the ends of the middle helix. Horizontal lines show chain direction along the helices but do not show the helical twist, therefore, position of the strands around the circumference of the helices is not accurately represented. Vertical lines represent the strand crossover points. Oligos 1 and 2 are 52 bases in length; oligos 3 and 4 are 72 bases long. Through most of our discussion oligo-1 is considered the reporter strand, however, oligo-2 is symmetrical to oligo-1 and can also be used in a reporter role. Approximate tile dimensions estimated from modeling are $65 \times 140 \times 20$Å.

Building on previous nomenclature [**FS93**], we refer to our tile design as TAO: "Triple" helix, with Anti-parallel crossovers, and an Odd number of helical half turns between crossovers (see Figure 1). Its three double-helices are designed to be co-planar. Anti-parallel crossovers indicate a reversal of strand direction following exchange to a new helix. An odd number of half-turns between crossovers ensures the desired reporter strand path through the tile (compare, for example, with the strand trace for TAE tiles in Section 3).

The prototype TAO, at 42 bases in length, goes through 4 full helical turns. Spacing between crossover points on the top helix (as represented in Figure 1) is 16 bases, for 3 helical half-turns; on the middle helix, 5 bases for 1 half-turn; and on the bottom helix, 26 bases for 5 half-turns. The shortest dsDNA arms between crossover and tile edge are 8 bases on the bottom helix. The prototype TAO was designed without ssDNA sticky ends for ease of structural analysis. Tiles functional for superstructure assembly will be equipped with $5'$ ssDNA pads of about 5 bases in length. In the 2D assemblies, described below, only 4 pads will be required, therefore the TAO as shown contains hairpins on both ends of the middle helix. These hairpins reduce the number of oligonucleotides needed for construction of the tile; however, they can be eliminated if more pads are required or if altered strand traces through the tile are desired.

The specific sequences used for the prototype TAO tile were designed using stochastic hill-climbing to search for nucleotide sequences that would provide the desired base-pairing while avoiding undesired pairings. The particular sequences chosen are drawn from a huge potential pool of sequences; we can, therefore, maintain the desired TAO structure while using different internal sequences to represent different digit values for tiles in the computational superstructure. The computational assemblies described below make use of binary encoding schemes (tiles hold values 1 or 0); however, for some calculations it may be useful to utilize expanded encodings (for example base-10). The tile assemblies are completely compatible with expanded alphabets, and would simply require a larger number of starting tile types.

2.1. TAO Tile Construction Results. We have successfully built and analyzed a TAO tile in the laboratory and detailed the experimental data [**LYK+99**]. Briefly, four oligonucleotides (with sequences given in Figure 1) were chemically synthesized and purified by polyacrylamide gel electrophoresis (PAGE) under denaturing conditions. DNA concentrations of the pure stocks were measured by UV-absorption at 260 nm. Stoichiometric and non-stoichiometric mixtures of oligos (buffered at pH 7.9 - 8.0 and containing 10 mM $MgCl_2$) were heated to 95°C for 5 minutes, then annealed by cooling slowly to ambient temperature over the course of 1 to 1.5 hours. Following annealing the tiles were subjected to various tests of structural integrity.

First, we demonstrated formation of individual tiles of designed strand stoichiometry and molecular weight as determined by PAGE under non-denaturing conditions. Further, we subjected the complex to random backbone cleavage analysis (using hydroxyl radical chemistry) and showed cleavage and protection patterns on the nucleotide level as predicted from the tile design. That is, bases predicted to be buried within the tile, either in crossover points or helix faces occluded by neighboring helix, showed protection and decreased sensitivity to backbone scission. As expected, bases predicted to be exposed to solvent water showed the highest rates of scission.

We performed Ferguson analysis, a technique involving the measurement of electrophoretic migration rates in gels of various polyacrylamide concentrations. Ferguson analysis provides a means of comparing hydrodynamic properties of complexes with those of related complexes. Our results showed an appropriate decrease in electrophoretic mobility for the TAO tile compared with a DX tile and a dsDNA complex of the same overall length (42 bases). We also measured the thermal stability of the tile by monitoring UV absorption during heat induced melting of tile structure. The particular TAO tile we examined appears to be stable at temperatures below approximately 50°C.

In summary, the prototype TAO has been shown to form with the proper oligonucleotide stoichiometry, desired molecular weight, and expected structural signatures on the whole complex and individual nucleotide levels. The tile also displays acceptable thermal stability for use as a component in the construction of computational tilings.

2.2. Results of Lattice Assembly on Preformed Scaffold Strands. In the 2D assembly experimental designs described below, we will make use of random formation of input tile layers. These input layers are capable of providing a reporter strand to enable read-out of the specific inputs from each computational complex. For many applications it may prove useful to encode specific inputs rather than randomly assembling large populations of input layers. Additionally, it may be necessasry to use the output of one self-assembly step as the input to a subsequent step. If preformed reporter strands of desired structure could be used as a scaffold on which to assemble a tile layer, they could be used to input specific values into computational assemblies. To test tile assembly onto preformed inputs, we constructed scaffold strands. The scaffold strand is the structural equivalent of a reporter strand, but in our test experiments it is formed by conventional, double-strand nucleic acid techniques rather than resulting from a tile assembly.

Scaffold strands from oligos-1 and -2 of the TAO tile (as shown in Figure 1, above) were prepared by the following procedure: anneal tile strand with a 24-base bridge oligo, ligate to form concatamers, and gel purify ssDNA concatamers of desired length. The tile oligo sequences and bridge annealing positions are as follows:

```
        5'                                                                                                  3'
        ----------------------------------------------------
Oligo-1:TTGGCTATCGAGTGGACACCGAAGACCTAACCGCTTTGCGTTCCTGCTCTACTTGGCTATCGAGTGGACACCGAAGACCTAACCGCTTTGCGTTCCTGCTCTAC
Bridge 1:                                       AAGGACGAGATGAACCGATAGCTC
                                              3'                        5'

        5'                                                                                                  3'
        ----------------------------------------------------
Oligo-2:GTTCAGCCTTAGTGGAGTGGAACGCAAAGCGGTTAGGTCTTCGGACGCTCGTGTTCAGCCTTAGTGGAGTGGAACGCAAAGCGGTTAGGTCTTCGGACGCTCGT
Bridge 2:                                       AGCCTGCGAGCACAAGTCGGAATC
                                              3'                        5'
```

The overlined sequence represents a single copy of tile oligo which is then repeated (without overlining) to show how two copies of a tile oligo are held together by a bridge oligo functioning as a splint. The bridge oligo (24 bases in length) anneals to the $5'$ end of one copy of tile oligo and to the $3'$ end of another copy, to make long multimers of the following form:

In the annealing/ligation reaction for scaffold strand formation, we used approximately 400 pmoles of phosporylated strand-1 (or strand-2), 100 pmoles unphosphorylated strand, and 400 pmoles bridge strand. These ratios gave acceptable concentrations of concatamers of desired sizes for purification by denaturing PAGE. We constructed and purified scaffold strands from oligo-1 and oligo-2 in lengths ranging from $2x$ to $10x$ (i.e. $10x$ strands have 10 tandem copies of the sequence required for a TAO tile). Following preparation and purification of desired scaffold strands, simple tile lattices were formed by replacing monomeric tile oligos with multimeric scaffolds in the tile formation annealing step. Tile lattices assemble around these scaffold strands, which act as tethers connecting adjacent tiles. These assemblies were analyzed by PAGE and atomic force microscopy (AFM); results are shown in Figure 2. The gel results (panel **A**) are exactly as expected. Scaffold strands $2x$-oligo-1 and $3x$-oligo-1 are 104 and 156 bases, respectively. They are ssDNA and run faster than dsDNA markers of the same length. The $2x$-TAO and $3x$-TAO assemblies should contain 252 and 378 base-pairs, respectively. The bands run slightly faster than dsDNA of the same molecular weight due to their more compact shape. The AFM study (panel **B**) was harder to interpret. Here, two scaffold strands ($2x$-oligo-1 and $2x$-oligo-2) were included simultaneously, which logically could result in many different superstructures. Indeed several different morphologies were observed (data not shown). The structures shown in Figure 2 are approximately 280Å in width, as expected for a lattice of the form shown in the diagram below. Imaged structures vary between ~ 600 and ~ 900Å in their long

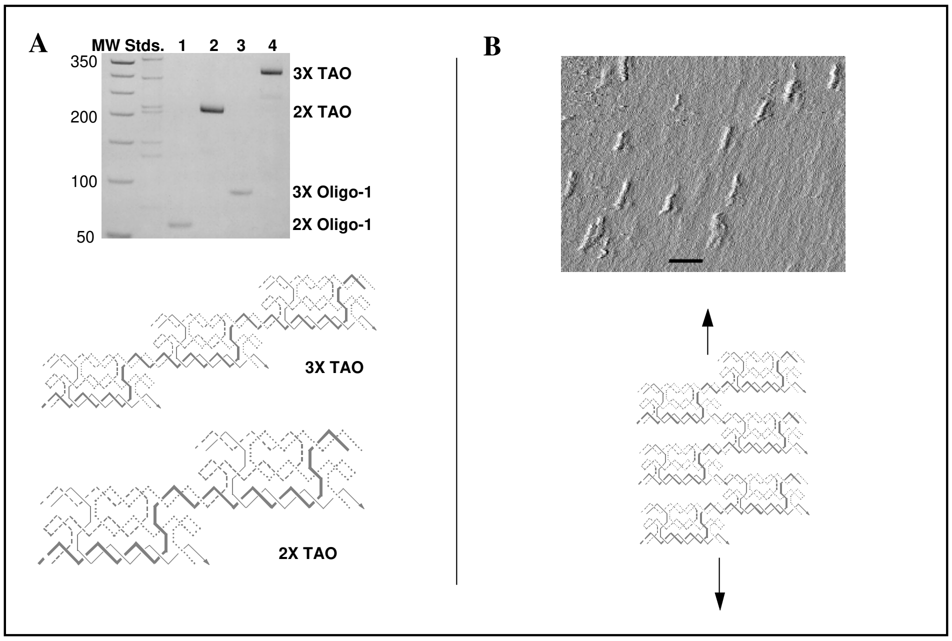

FIGURE 2. **Use of scaffold strands in small non-computational assemblies.** Panel **A** shows a polyacrylamide gel (run under non-denaturing conditions) which demonstrates assembly of TAO tile layers around $2x$ and $3x$ scaffold strands; below are diagrams of three tile ($3x$ TAO) and two tile ($2x$ TAO) assemblies constructed around $3x$-oligo-1 and $2x$-oligo-1 scaffold strands. Lanes 1 and 3 of the gel are scaffold strand alone; lanes 2 and 4 are scaffold strands annealed with oligos 2, 3, and 4 to form tile assemblies. Molecular weight markers (MW Stds.) are dsDNA with sizes indicated in basepairs. Panel **B** shows an AFM image of superstructures constructed by simultaneously including two scaffold strands ($2x$-oligo-1 and $2x$-oligo-2) during the tile formation annealing. The length of the black bar inset on the image is 1000Å. The diagram below shows an arrangement of tiles compatible with the image; arrows indicate directions in which lattice assembly can continue.

dimension and therefore appear to contain between ~ 14 and ~ 20 tiles each. AFM data gave Z-dimension heights for the observed structures just under 20Å above the mica surface, as expected for a flat lattice constructed from dsDNA. Further study of assemblies around a single scaffold strand will reolve any ambiguities.

We have demonstrated the use of scaffold strands as a means of assembling a DNA lattice. We might wish to answer another question regarding tile assembly around scaffold strands: is there a maximum length beyond which scaffold strands would function poorly due to, for example, tangling of partially formed lattices?

As of yet, we have not tested the upper length limit. The maximum size scaffold tested so far contained 6 tandem repeats; examination by PAGE indicated successful formation of the 756 base-pair assembly. It would be useful to test tile formation on scaffolds of hundreds or thousands of tandem repeats.

2.3. 2D Assemblies for XOR and addition. We will now outline our procedures for using the TAO tile described above to perform massively parallel arithmetic. The simplest 2D assembly computer will perform cumulative XOR computation using a 2-layer architecture: one input tile layer and one output layer. We will also describe the slightly more complicated 2D assemblies required to execute bit-wise XOR and addition of two binary inputs, using 3-layer superstructures. For computations on specific inputs, these procedures will make use of the scaffold strands mentioned in the previous section. Otherwise, the input tiles will randomly assemble and thereby generate a molecular look-up table in which each reporter strand encodes the inputs and outputs of a random calculation. A sufficient number of DNA tile molecules will provide full coverage of all possible n-bit input strings. Such look-up tables may be useful as input for further computations as they represent a unique library of sequences with a complex structural theme.

Recall that the exclusive-or (XOR) operation is a Boolean function that, given two inputs, yields 0 if the inputs hold the same value, and returns 1 if the inputs disagree in value. The input to a cumulative XOR computation is a sequence of Booleans $a_1, a_2, a_3...$, and the output is a sequence of Booleans $b_1, b_2, b_3...$ where $b_1 = a_1$ and for $i > 1$, $b_i = b_{i-1}$ XOR a_i. Thus the i^{th} output is the cumulative XOR of the 1^{st} through the i^{th} inputs.

A 2-layer tile system for computing cumulative XOR is shown in Figure 3 and Figure 4. As shown, it consists of 8 tile types: 2 input tiles (with value x and pads clockwise from lower left: C, A_x, and C, where $x \in \{0, 1\}$), 4 output tiles (with value (x XOR y) and pads B_y, $B_{(x \text{ XOR } y)}$, and A_x, where $x, y \in \{0, 1\}$), and 2 corner tiles (with pads C' and C, and B_0 and C'). The two types of input tile differ only in the sequences for the upper-left pad and the internal sequence encoding the tile's value. The other two pads (upper-right and lower-left) are complementary on all input tiles, so that assembly of the input layer can randomly generate all possible n-bit inputs. Output tiles use the bottom two pads to read inputs (one from the next input tile and one from the previous output tile), while the upper-right pad encode outputs. Four output tile types are required, one for each possible pair of input values in the XOR truth table. The reporter strand is shown as a diagonal line passing through each tile by the following path: in one pad, through the sequence encoding the tile's value, and out through a pad on the opposite side of the tile. The corner tiles function primarily to route the reporter strand and thus connect its input and output halves.

In addition to connecting the input and output halves of the reporter strand, the corner tiles set the boundary conditions for the computation. The two corner tiles put the first input bit (a_1) into position for formation of the binding slot for the first output bit (b_1). Of the two corner tiles depicted in Figure 3 and Figure 4, the one farthest left (two pads in one side and reporter strand looped between them) can be a TAO. For the other corner tile (two pads on the top helix connected by reporter strand) we require a different strand topology which can be provided by a TAE tile as described below in Section 3 or by a DAE tile [**FS93**].

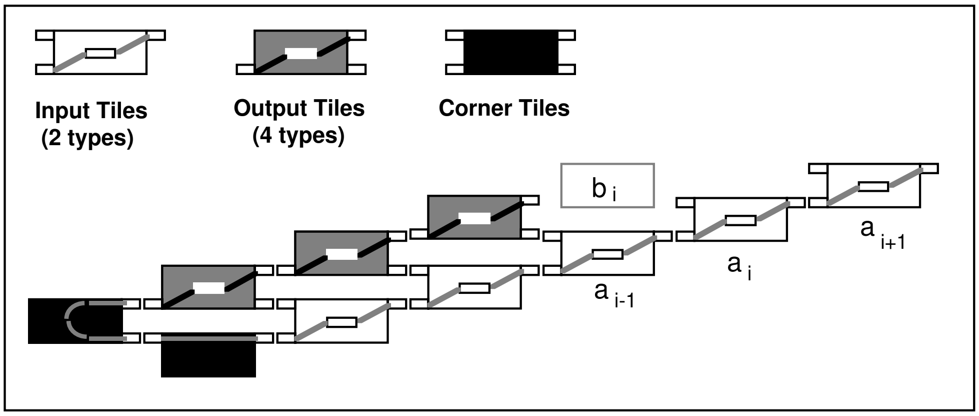

FIGURE 3. **2D Assembly for Cumulative XOR.** The tiles schematics represent TAO tiles. ssDNA pads are shown sticking off the body of the tiles. The small central box on the input and output tiles represents a nucleotide sequence encoding the value (0 or 1) of the tile. The reporter strand is shown as a diagonal line passing through each tile. The figure shows a partially assembled superstructure in which the next step is the binding of an output tile into the slot labeled b_i, followed by continued assembly of the output layer up and to the right.

Figure 3 shows a partially assembled superstructure in which the next step is the binding of an output tile into the slot labeled b_i, followed by continued assembly of the output layer up and to the right. To properly bind, the output tile must match its lower-right pad with the upper-left pad of tile a_i and its lower-left pad with the upper-right pad of the previous output tile (b_{i-1}). Following the annealing of tile b_i, a new slot b_{i+1} will be formed with one pad from b_i and one pad from a_{i+1}.

The 2-layer 2D assemblies for cumulative XOR calculation, as proposed, will function with TAO tile structures, but could alternatively use simpler double-crossover tiles containing a hairpin on one helix [**WER99**]. We chose to use TAOs in our design in order to test the tiles and assembly techniques that will be required for 3-layer superstructures for bit-wise XOR and addition.

We now move to discussion of the 3-layer assemblies as shown in Figure 5. The components and methodology for 3-layer superstructures closely follows that for the 2-layer cumulative XOR assembly. Again, we rely on random assembly of input layers, but this time we need two input layers per superstructure, and we need corner tiles on both ends of the superstructure in order to properly join both the input and the output portions of the reporter strand into a continuous piece of DNA.

The pads on input and corner strands are essentially as before, except that the second set of input tiles uses its own labels. For addition, the eight output tiles make use of all four pads as well as the (internal) value encoded on the reporter strand: the value is (x XOR y XOR c) and, clockwise from lower left, the pads are

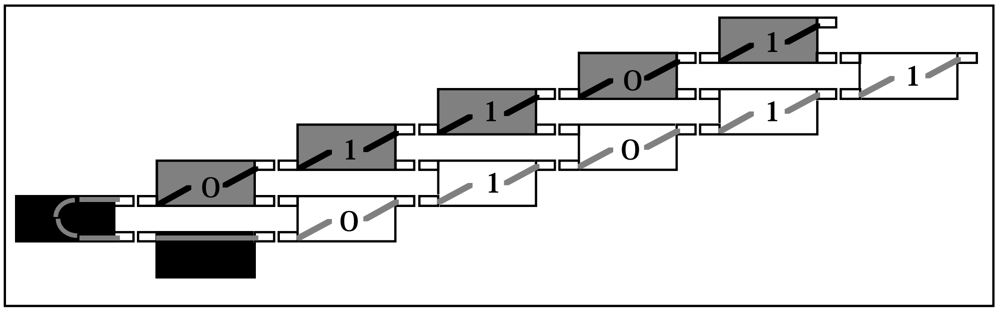

FIGURE 4. **Example I/O for cumulative XOR.** This pictured assembly has input 01011 and output 01101. In examining this cartoon of the assembly, we read both input and output strings from left to right. However, the reporter strand starts ($5'$ end) in the input tile most distant from the corner tiles and proceeds ($5'$ to $3'$) through the input tiles in reverse order, then loops through the corner tiles and continues through the output tiles in forward order. Therefore, the ordering of information on the reporter is: $a_n, a_{n-1}, a_{n-2}, ..., a_1, b_1, b_2, b_3, ..., b_n$.

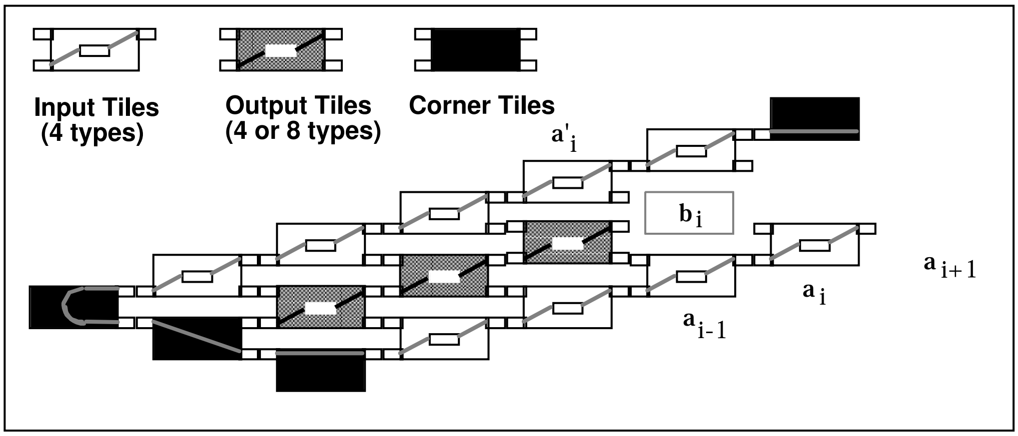

FIGURE 5. **Bit-wise XOR and Addition.** This schematic representation of a 3-layer assembly maintains the conventions used in the previous figures. The superstructure will assemble such that $b = a + a'$ or $b_i = a_i$ XOR a'_i depending on which set of output tiles is used. The reporter strand traces through the entire assembly in a spiral, starting in the n^{th} tile of the lower input layer and down through the 1^{st}, through the corner tiles, through the upper input layer (1^{st} to n^{th}), through distal corner tiles and down through the output layer (n^{th} to 1^{st}).

C_c, A'_y, $C_{(x+y+c\geq 2)}$, and A_x, where $x, y, c \in \{0,1\}$. Thus, the pads along the middle diagonal represent the carry bit during the addition. For bit-wise XOR, there is no need to pass a carry bit, so lower-left pads are complementary to upper-right pads on all output tiles, therefore only four tile types are required.

The total number of tile types required for the cumulative XOR assembly is at least eight: 2 inputs, 4 outputs, and 2 corner tiles for the lower-left end of the superstructure. It may also prove useful to include corner tiles which terminate the upper-right end of the assembly, thus "bookending" the computation and perhaps increasing accuracy by ensuring that input and output strings correspond in length. The bit-wise XOR and addition assemblies require at least 13 and 17 distinct tile types, respectively. Again expansion of these repertoires by an increase in the number of corner tiles may be desirable to ensure proper nucleation and termination of the superstructures. The 1-layer assembly for addition described below in Section 3 requires exactly 10 tile types.

The 2D assemblies described here allow read-in of specific inputs by scaffold strands (Section 2.2, above) and read-out of input and output strings on reporter strands. The reporters can be purified following assembly and ligation, since they have a large, characteristic size and can be separated from other DNA in the test tube by denaturing PAGE. To determine a particular output string, a reporter strand can be used as template for PCR (polymerase chain reaction). Adleman [**Adl94**] made use of graduated PCR using six different pairs of primers to gauge the distance between nodes on the Hamiltonian path encoded on his DNA. For the binary representation we use, we need only two primer pairs (2 PCR reactions) for our read-out, both containing a constant primers whose binding site is located in a corner tile, and each containing one of two primers corresponding to the value encoding sequence words (0 or 1) located in each tile. The reaction containing corner-tile primer plus value=1 primer, will provide distances from the corner tile to each tile whose value is set to 1. Since these distance increments are uniform, values for each digit in the I/O strings can be read directly. Likewise, using primer corresponding to the sequence word for bit value=0, we will obtain the digit locations in the input and output for bits whose value is 0.

Two potential problems may prevent the 2D assembly computations from proceeding as described. First, the scheme requires that assembly progress in two distinct stages: random input assembly, followed by specific output assembly. If output tiles begin to anneal with each other or with partially formed input chains, there exists the potential for improper assembly. The ssDNA pad sequences must be designed such that input tiles associate with each other at a slightly lower temperature than that which allows output tile annealing. While this temperature separation is possible in theory it may be difficult to achieve in practice. The second potential problem involves the "slot filling" association of output tiles into the forming computational assembly. The proper annealing of output tile requires matching with two (or 3 for addition) ssDNA pads, however, with some probability an improper output tile may fill a slot based on matching with a single pad (or 2 pads for addition). If the incorrect output tile fails to dissociate it will block the processor, returning either a truncated answer or an answer which no longer corresponds to the input. Results of an initial experiment have shown that dsDNA with two matching sticky-end achieves preferential binding over single match DNA [**WYS98**], however this critical result has yet to be demonstrated in more complex assemblies involve larger tile structures. Although linear assemblies with few layers are likely to be less affected by these problems than fully 2D assemblies, these potential problems have induced interest in alternative strategies for computation by self-assembly, such as the one given in the next section.

3. String Tile Assemblies

The concept of string tile assemblies derives from Eng's observation that allowing neighboring tiles in an assembly to associate by *two* sticky ends on each side, he could increase the computational complexity of languages generated by linear self-assembly [**WER99**]. By allowing contiguous strings of DNA to trace through individual tiles and the entire assembly multiple times, surprisingly sophisticated calculations can be performed with 1-layer linear assemblies [**WER99**]. For example, we describe below a string tile system that computes addition or bit-wise XOR.

String tiling, in essence, is the collapse of multi-layer assemblies into simpler superstructures by allowing individual tiles to carry multiple segments of the reporter strand. The 2-layer lattice for cumulative XOR can be reduced to unbranched dsDNA. The 3-layer addition assembly collapses to a linear row of three-helix (or perhaps even two-helix) tiles. By allowing a reporter strand to wind through the assembly, passing through each tile multiple times, we can encode in each tile an entire row of the truth table. Input and output values are recorded on words internal to the tile, while information passed to neighboring tiles (e.g. carry bits) are encoded on the ssDNA pads.

String tile arithmetic implementations have a number of advantageous properties.

1. Input and output strings assemble simultaneously.
2. Each row in the truth table for the function being calculated is represented as a single tile type, where all inputs and outputs are encoded on that tile. Each bit-wise operation is directly encoded in the structure of a tile. Contrast this with the 2D systems described above, in which input bits and output bits each had their own tile types.
3. Adjacent tiles only have to agree on the value of the carry bit between them; one tile passes and one tile receives the bit. The set of pads for carry=0 is distinct from the carry=1 pads.
4. Adjacent tiles associate via multiple sticky ends, all of which either agree or disagree, and therefore there is no need to differentiate between single and double pad matches (as required in the 2D assemblies).

The TAE tile shown in Figure 6 has four half-turns of DNA helix between the crossover points connecting the top helix with the middle helix and between the crossover points connecting the middle helix with the bottom helix. This could be trimmed to two half-turns without loss of the desired strand routing; however, the longer spacing provides an increased number of base positions for the internal sequences that encode the (input and output) values of the tile. Longer sequences also may be useful for improving primer binding for graduated PCR read-out of the reporter strand. The longest oligonucleotide in the tile (shown in light gray in Figure 6) may be useful as a diagonal reporter strand in certain constructions. It is not required for the assembly described below and may make purification of the desired reporter (black) more difficult, since it winds around reporter segments

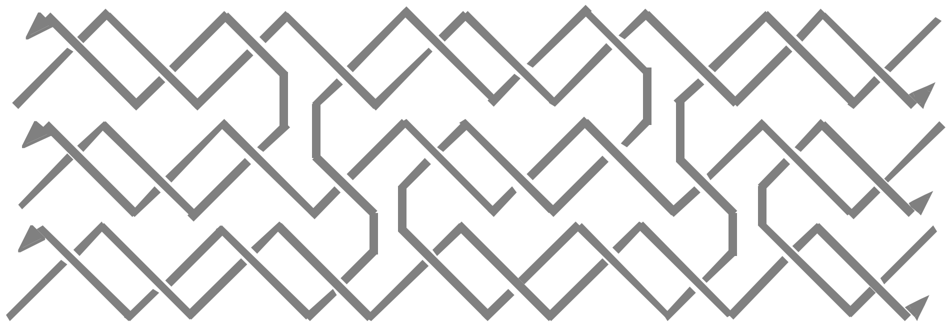

FIGURE 6. **TAE Tile.** The structure of the TAE resembles the TAO in that it is constructed of three double-helices linked by strand exchange, however, it contains an Even (rather than Odd) number of helical half-turns between crossover points linking each pair of helices. Even spacing of crossovers allows reporter strands (shown in black) to stretch straight through each helix from one side of the tile to the other. These 3 reporter segments will be used for building up a long strand which records inputs and outputs for the entire assembly.

repeatedly. It should be possible to introduce a strand break in the long oligo, perhaps on the middle helix, thus increasing the number of oligos in the tile but decreasing the strand tangling. At the nick, the strands should be lacking a $5'$ phosphate, so that the nick does not get sealed by ligase.

Figure 7 presents details for a linear assembly capable of calculating addition or bit-wise XOR. The scheme is illustrated using TAE tiles, however it is not particular to TAEs. The system is a variation of one outlined previously which utilized DPE tiles (Double-crossover (2 helices), Parallel crossovers, with an Even number of half-turns between) to execute a similar strand trace through the assembly [**WER99**]. TAE tiles may be preferable to DPE because early studies on DX complexes containing parallel crossovers showed them to be less structurally stable than their anti-parallel counterparts [**FS93**]. It is possible that DPE tiles could be stabilized by increasing the lengths of dsDNA arms flanking the strand exchange points; however, this has not yet been demonstrated experimentally.

I_{A_i} and I_{B_i} indicate sequences encoding binary values for the two input bits for the current operation. O_i encodes the value of the output bit. The sticky end pads on the right encode the value for the carry bit coming into this operation, while the left-hand pads pass the new carry value on to the next bit operation. Each of the nine tile locations has two possible sequence words; one each for values 1 and 0. One tile type is required for each of the eight rows in the truth table, plus one tile type for each of the two ends of the assembly. The truth table provides the allowed pairings of values for each of the nine tile locations. The c_i column provides the value for all three right-hand pads $(c1_i, {}^{\sim}c2_i, c3i)$ while column c_{i+1} gives the value for the three left-hand pads $({}^{\sim}c1_i, c2_i, {}^{\sim}c3_i)$. Sequence words for $c1$, $c2$, and $c3$ are distinct from one another so that each tile binding reaction involves annealing

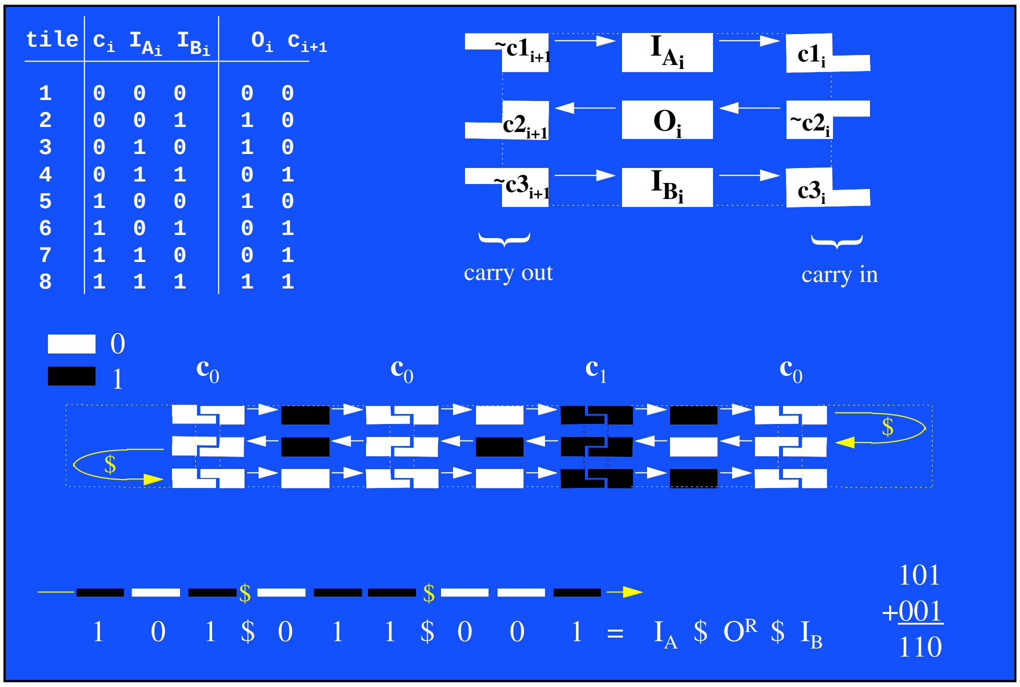

FIGURE 7. **String Tile Addition with TAE Building Blocks.** An example tile is shown schematically in the upper right corner; it is of the TAE form with the trace of the three reporter segments indicated by arrows. The truth table of values for c_i, I_{A_i}, I_{B_i}, and O_i is shown in the upper left; each row corresponds to a type of tile. An example of an assembled structure is given at the bottom, along with its interpretation. The reporter strand follows a Z-shaped path from the upper left to the lower right.

of three separate pairs of pads. Note that $\tilde{}c1$ represents the complement of $c1$. When two tiles bump into each other, they either agree at all 3 pads and bind or they disagree at all three pads and fail to bind. Binding that is not in register (e.g. $c1$ bound to $\tilde{}c3$) is thereby disallowed. An example computational assembly showing addition of two 3-bit numbers is shown at the bottom. The linear assembly is composed of five tiles, starting on the right: an end or terminal tile, then a type 4 tile, then a type 5, then a type 3, then a left-end tile. Shown below the assembly is a summary of the bit values in the order they are recorded on the reporter strand (carry values, although also written on the reporter strand, are not shown in this simplified drawing). The '\$' symbols indicate marker sequences from the end tiles which mark the boundaries between input and output strings and will be useful as primer binding sites during read-out by graduated PCR.

The ordering of I/O digit values on the reporter strand is depicted in Figure 7. Following the strand from $5'$ to $3'$ we encounter: I_A (n^{th} bit through 1^{st} bit), O (1^{st} through n^{th}), I_B (n^{th} through 1^{st}). This orientation allows the inputs to be read with the highest digit first (on the left) but may be undesirable since the output digits are listed on the reporter strand in the opposite direction from the inputs. Other digit value orderings are possible, and with the use of slightly more complex tiles (4 or 5 helices) we can encode outputs in the same orientation as inputs [**WER99**].

The linear assembly for addition and bit-wise XOR presented in Figure 7 displays many benefits over 2D assembly methods: simpler superstructure (1 layer), fewer tile types (exactly ten types are sufficient), shorter reporter strands, neighbor tile matching by 3 pads instead of one, etc. The string tile addition scheme outlined in Figure 7 may have the greatest chance of success of all the DNA computational assembly procedures discussed. We are currently working to experimentally test this system.

4. Discussion and Conclusions

We have described the design of two novel DNA tile types, TAO and TAE, as well as summarizing the experimental analysis of a TAO. We have outlined the use of reporter strands and scaffold strands for I/O in computational assemblies, and presented experimental data demonstrating formation of lattice structures around scaffold strands. We have also detailed three computational assemblies, including a new string tile system which simplifies many aspects of DNA tilings and shows promise for experimental construction. We have been working primarily on the implementation of arithmetic for prototyping functional DNA-based computers. It is our intention that these experiments and prototypes will enable development of materials and techniques that may advance DNA-based computing beyond the realm of demonstration and into useful applications.

Acknowledgments

We wish to sincerely thank: Ned Seeman and Hao Yan for their considerable and invaluable roles in construction and analysis of the TAO tile; Tony Eng and Grzegorz Rosenberg for development and enlightening discussion of string tile concepts; and Wolfgang Frey and Ashutosh Chilkoti for assistance with atomic force microscopy. This work was supported in part by grants NSF/DARPA CCR-9725021, CCR-9633567, NSF IRI-9619647, and ARO contract DAAH-04-96-1-0448.

References

[Adl94] Leonard M. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266:1021–1024, November 11, 1994.

[BB98] Eric B. Baum and Dan Boneh. Running dynamic programming algorithms on a DNA computer. In Landweber and Baum [**LB98**], pages 77–85.

[Ber66] Robert Berger. The undecidability of the domino problem. *Memiors of the AMS*, 66:1–72, 1966.

[Buc62] J. Richard Buchi. Turing machines and the entscheidungsproblem. *Mathematiche Annalen*, 148:201–213, 1962.

[CS91] Junghuei Chen and Nadrian C. Seeman. The synthesis from DNA of a molecule with the connectivity of a cube. *Nature*, 350:631–633, 1991.

[DZS92] Shou Ming Du, Siwei Zhang, , and Nadrian C. Seeman. DNA junctions, antijunctions, and mesojunctions. *Biochemistry*, 31:10955–10963, 1992.

[Eng99] Tony Eng. Linear DNA self-assembly with hairpins generates the equivalent of linear context-free grammars. In Rubin and Wood [**RW99**].

[FB97] Bin Fu and Richard Beigel. Molecular computing, bounded nondeterminism, and efficient recursion. *Automata, Languages and Programming*, 1256:816–826, 1997.

[FB99] Bin Fu and Richard Beigel. A comparison of resource-bounded molecular computation models. *Algorithmica*, 24(2):87–95, 1999.

[FS93] Tsu-Ju Fu and Nadrian C. Seeman. DNA double-crossover molecules. *Biochemistry*, 32:3211–3220, 1993.

[GS87] Branko Grünbaum and G. C. Shephard. *Tilings and Patterns*. W. H. Freeman and Company, New York, 1987.

[JKS98] Nataša Jonoska, Stephen A. Karl, and Masahico Saito. Three dimensional DNA structures in computing. In Lila Kari, Harvey Rubin, and David H. Wood, editors, *Proceedings of the* 4^{th} *DIMACS Meeting on DNA Based Computers, held at the University of Pennsylvania, June 16-19, 1998*, pages 189–200, preliminary, 1998.

[JKS99] Nataša Jonoska, Stephen A. Karl, and Masahico Saito. Creating 3-dimensional graph structures with DNA. In Rubin and Wood [**RW99**], pages 123–135.

[LB98] Laura F. Landweber and Eric B. Baum, editors. *DNA Based Computers II: DIMACS Workshop, June 10-12, 1996*, volume 44, Providence, RI, 1998. American Mathematical Society.

[LP81] H. R. Lewis and C. H. Padadimitriou. *Elements of the theory of computation*. Prentice-Hall, 1981.

[LSS99] Furong Liu, Ruojie Sha, and Nadrian C. Seeman. Modifying the surface features of two-dimensional DNA crystals. *Journal of the American Chemical Society*, 121(5):917–922, 1999.

[LYK+99] Thomas H. LaBean, Hao Yan, Jens Kopatsch, Furong Liu, Erik Winfree, John H. Reif, and Nadrian C. Seeman. Construction, analysis, ligation, and self-assembly of DNA triple crossover complexes. *in preparation*, 1999.

[LYQS96] Xiaojun Li, Xiaoping Yang, Jing Qi, and Nadrian C. Seeman. Antiparallel DNA double crossover molecules as components for nanoconstruction. *Journal of the American Chemical Society*, 118(26):6131–6140, 1996.

[MAS99] Nobuhiko Morimoto, Masanori Arita, and Akira Suyama. Solid phase DNA solution to the Hamiltonian Path Problem. In Rubin and Wood [**RW99**].

[MSS97] C. D. Mao, W. Q. Sun, and N. C. Seeman. Assembly of Borromean rings from DNA. *Nature*, 368:137–138, March 13, 1997.

[MSSS99] C. D. Mao, W. Q. Sun, Z. Y. Shen, and N. C. Seeman. A nanomechanical device based on the B-Z transition of DNA. *Nature*, 397:144–146, January 14, 1999.

[Rei99] John Reif. Local parallel biomolecular computing. In Rubin and Wood [**RW99**], pages 217–254.

[Rob71] Raphael M. Robinson. Undecidability and nonperiodicity of tilings of the plane. *Inventiones Math.*, 12:177–909, 1971.

[RW99] Harvey Rubin and David Harlan Wood, editors. *DNA Based Computers III: DIMACS Workshop, June 23-25, 1997*, volume 48 of *DIMACS: Series in Discrete Mathematics and Theoretical Computer Science*, Providence, RI, 1999. American Mathematical Society.

[SCD+93] Nadrian C. Seeman, Junghuei Chen, Shou Ming Du, John E. Mueller, Yuwen Zhang, Tsu Ju Fu, Yinli Wang, Hui Wang, and Siwei Zhang. Synthetic DNA knots and catenanes. *New Journal of Chemistry*, 17:739–755, 1993.

[SCK89] Nadrian C. Seeman, Junghuei Chen, and Neville R. Kallenbach. Gel electrophoretic analysis of DNA branched junctions. *Electrophoresis*, 10:345–354, 1989.

[See82] Nadrian C. Seeman. Nucleic-acid junctions and lattices. *Journal of Theoretical Biology*, 99(2):237–247, 1982.

[See98] Nadrian C. Seeman. DNA nanotechnology: novel DNA constructions. *Annual Review of Biophysics and Biomolecular Structure*, 27:225–248, 1998.

[Wan61] Hao Wang. Proving theorems by pattern recognition. II. *Bell System Technical Journal*, 40:1–42, 1961.

[Wan63] Hao Wang. Dominoes and the AEA case of the decision problem. In *Proc. Symp. Math. Theory of Automata*, pages 23–55, New York, 1963. Polytechnic Press.

[Wan75] Hao Wang. Notes on a class of tiling problems. *Fundamenta Mathematicae*, 82:295–305, December 1975.

[WER99] Erik Winfree, Tony Eng, and Grzegorz Rozenberg. String tile models for DNA computing by self-assembly. *in preparation*, 1999.

[Win96] Erik Winfree. On the computational power of DNA annealing and ligation. In Richard J. Lipton and Eric B. Baum, editors, *DNA Based Computers: DIMACS* *workshop, April 4, 1995*, volume 27, pages 199–221, Providence, RI, 1996. American Mathematical Society.

[WLWS98] Erik Winfree, Furong Liu, Lisa A. Wenzler, and Nadrian C. Seeman. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394:539–544, 1998.

[WMKS91] Yinli Wang, John E. Mueller, Börries Kemper, and Nadrian C. Seeman. Assembly and characterization of five-arm and six-arm DNA branched junctions. *Biochemistry*, 30:5667–5674, 1991.

[WYS98] Erik Winfree, Xiaoping Yang, and Nadrian C. Seeman. Universal computation via self-assembly of DNA: Some theory and experiments. In Landweber and Baum [**LB98**].

[YS99] Hiroshi Yoshida and Akira Suyama. Solution to 3-SAT by breadth first search. In Erik Winfree and David K. Gifford, editors, *Proceedings of the* 5th *DIMACS Meeting on DNA Based Computers, held at MIT , June 14-15, 1999*, preliminary, 1999.

[ZS94] Yuwen Zhang and Nadrian C. Seeman. The construction of a DNA truncated octahedron. *Journal of the American Chemical Society*, 116:1661–1669, 1994.

(T. LaBean) Department of Computer Science, Duke University, Durham, NC 27708,
E-mail address: thl@cs.duke.edu

(E. Winfree) Department of Molecular Biology, Princeton University, Princeton, NJ 08544

Departments of Computer Science and Computation and Neural Systems, California Institute of Technology, Pasadena, CA 91125
E-mail address: winfree@hope.caltech.edu

(J. Reif) Department of Computer Science, Duke University, Durham, NC 27708,
E-mail address: reif@cs.duke.edu