1 DNA pattern assembly

We close one significant conjecture in DNA pattern assembly, which is the NP-hardness of minimizing a system that uniquely self-assembles a given binary pattern, using a computer-assisted proof.

The *rectilinear* TASM (RTAS) is a variant of Winfree’s aTAM system for assembling (rectangular) patterns. An RTAS is a pair \((T, \sigma_u)\) of a set \(T\) of tile types and an L-shape seed \(\sigma_u\). According to the following rule, starting from the seed \(\sigma_u\), it attaches a tile one after another:

**RTAS’ tiling rule:** A tile can attach at a position \((x, y)\) if and only if its west glue matches the east glue of the tile on \((x - 1, y)\) and its south glue matches the north glue of the tile on \((x, y - 1)\).

This rule suggests that a position does not become attachable until its west and south neighbor positions are tiled.

For instance, see below an RTAS with 4 tile types (half-adder). At the initial time point, an orange tile of the type whose west glue is 1 can attach at \((1, 1)\), while no tile of the other three types can due to label-mismatching. The attachment makes \((1, 2)\) and \((2, 1)\) attachable:

![Halfadder tile types](image)

When no tile can attach any more, the assembly is terminal. If there exists one pattern \(P\) such that the pattern of any of the terminal assemblies of an RTAS is \(P'\), then the RTAS uniquely self-assembles the pattern \(P\).

**Problem:** \((k\text{-colored Pats \ (k\text{-pats)})}\).

**Given:** A \(k\)-colored pattern \(P\);

**Output:** a smallest RTAS that uniquely self-assembles \(P\).

The best known result is the NP-hardness of 11-PATS\(\dagger\) whose proof is manually-checkable. Using a massively-parallel computer program, we close the conjecture as follows.

**Theorem.** 2-Pats is NP-hard.

2 2-Pats is NP-hard

M-SAT is a variant of SAT, in which a CNF Boolean formula \(F\) without negations and an integer \(k\) are given and asked whether or not \(F\) can be satisfied by only allowing \(k\) variables to be true. M-SAT is NP-complete.

Our proof for the NP-hardness of 2-Pats is the reduction of M-SAT to the decision variant of 2-Pats. The following set \(T\) of 13 tile types (2 black and 11 white) plays a significant role in this reduction:

![Black and white tiles](image)

All colors that appear on these tiles other than black and white are only encoded as information in the glue, and only shown as colored "signals" crossing the tiles for logical clarity to help display the logical algorithm being followed by the assembly.

These tiles, being used by an RTAS, verify an instance \((\phi, k)\) of M-SAT, which is encoded on the L-shape seed of the RTAS. The verification results in an assembly similar to that below. The seed encodes \(\phi\) and the resultant assembly is forced by the logic encoded in the glue to try to satisfy \(\phi\), resulting in the correct black and white pattern if and only if \(\phi\) can be satisfied with \(k\) true variables.


Department of Information and Computer Science
School of Science
Aalto University, Finland

**Lemma.** If an RTAS with at most 13 tile types uniquely self-assembles a pattern including GADGET, then its tile type set must be isomorphic to \(T\).

Due to this lemma, deciding whether there exists an RTAS with at most 13 tile types that uniquely self-assembles \(P(\phi)\) is equivalent to deciding whether an RTAS with the tile type set \(T\) uniquely self-assembles \(P(\phi)\).

Since GADGET is a part of \(P(\phi)\), if \(\phi\) is not satisfiable, the RTAS cannot uniquely self-assemble \(P(\phi)\).

3 Computer-assisted verification of Lemma

The verification of the lemma requires approximately a year of computation time. A C++ program was developed for that, whose code is freely available online:


This program was also implemented in a functional language, which runs in a massively-parallel fashion. We proved its correctness by formal proofs. Certifying parallel programs is one of the greatest challenges of program proofs nowadays.

Contact information for comments & improvement ideas: Shinnosuke Seki

Email: shinnosuke.seki@aalto.fi