A manually-checkable proof for the NP-hardness of 11-color pattern self-assembly tileset synthesis

Aleck Johnsen1 · Ming-Yang Kao1 · Shinnosuke Seki2,3

Published online: 24 November 2015
© Springer Science+Business Media New York 2015

Abstract Patterned self-assembly tile set synthesis (PATS) aims at minimizing the number of distinct DNA tile types used to self-assemble a given rectangular color pattern. For an integer $k$, $k$-PATS is the subproblem of PATS that restricts input patterns to those with at most $k$ colors. We give an efficient 1-IN-3-SAT verifier, and based on that, we establish a manually-checkable proof for the NP-hardness of 11-PATS; the best previous manually-checkable proof is for 29-PATS.

Keywords DNA pattern self-assembly · Tile complexity · Manually-checkable proof

This is an improved full version of our previous works presented at conferences Johnsen et al. (2013), Seki (2013).

Shinnosuke Seki
s.seki@uec.ac.jp

Aleck Johnsen
aleckjohnsen2012@u.northwestern.edu

Ming-Yang Kao
kao@northwestern.edu

1 Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208, USA
2 Helsinki Institute for Information Technology (HIIT) and Department of Computer Science, Aalto University, P. O. Box 15400, 00076 Aalto, Finland
3 Department of Communication Engineering and Informatics, The University of Electro-Communications, 1-5-1, Chofugaoka, Chofu 1828585, Tokyo, Japan

Springer
1 Introduction

Tile self-assembly is an algorithmically rich model of “programmable crystal growth.” Well-designed molecules (square-like “tiles”) with specific binding sites can uniquely form a single target shape even subject to the chaotic nature of molecules floating in a well-mixed chemical soup. Such tiles were first implemented as DNA double-crossover molecules in laboratories in 1998 (Winfree et al. 1998).

Shape building is one primary goal of self-assembly; pattern painting\(^1\) is another. Various designs of pattern assemblers have been proposed including periodic gold nanoparticle arrays (Zhang et al. 2006), binary counters (Barish et al. 2005; Cook et al. 2004; Evans 2014), Sierpinski triangles (Cook et al. 2004; Rothemund et al. 2004), multiplexers (Cook et al. 2004), and \textbf{NP}\(-\text{-complete problem solvers} (Brun 2008; Brun 2012; Czeizler and Popa 2013; Johnsen et al. 2013; Kari et al. 2015a; Seki 2013). For the theory and practice of color pattern assembly, a variant of tile assembly system (TAS) in the Tile Assembly Model (Winfree 1998), called the \textit{rectilinear TAS} (RTAS), has been studied (Czeizler and Popa 2013; Göös et al. 2014; Johnsen et al. 2013; Kari et al. 2015a; Ma and Lombardi 2008; Ma and Lombardi 2009; Seki 2013). An RTAS consists of an L-shape seed (scaffold) and a finite number of tile types; an example RTAS in Fig. 1 has two white tile types and two black tile types. It is provided with an inexhaustible supply of copies of each of its tile types, which are called \textit{tiles}, and lets these tiles attach to the seed one after another, assembling a rectangular pattern (the example RTAS in Fig. 1 thus self-assembles a binary counter pattern). The problem of \textit{patterned self-assembly tile set synthesis} (PAT\(\text{S}\)) aims at minimizing the number of tile types necessary for an RTAS to uniquely assemble a given rectangular pattern. \textit{PAT}\(\text{S}\) is \textbf{NP}\(-\text{-hard} (Czeizler and Popa 2013; Kari et al. 2015b).

When the number of colors in the pattern is bounded by a constant \(k\), \textit{PAT}\(\text{S}\) becomes practically meaningful, as “\textit{any given logic circuit can be formulated as a colored rectangular pattern with tiles, using only a constant number of colors}” (Molecular Programming 2012). This restricted problem called \(k\)-\textit{PAT}\(\text{S}\) is as hard as \textit{PAT}\(\text{S}\). \(60\)-\textit{PAT}\(\text{S}\) was first shown to be \textbf{NP}\(-\text{-hard} (Seki 2013) (2-\textit{PAT}\(\text{S}\) was claimed so in Ma and Lombardi (2009), but the proof was incorrect). Later, 29-\textit{PAT}\(\text{S}\) was also shown to be \textbf{NP}\(-\text{-hard} (Johnsen et al. 2013). These proofs are manually-checkable, i.e., the verification of their correctness does not necessitate computer programs or theorem provers. A computer-assisted proof has been recently proposed for the \textbf{NP}\(-\text{-hardness of 2-\textit{PAT}\(\text{S}\) (Kari et al. 2015a).

All of these proofs and ours in this paper take a common strategy as follows:

1. Choose an \textbf{NP}\(-\text{-complete problem.
2. Design a set \(T\) of \(n\) tile types with \(k\)-colors as a verifier of the above chosen problem such that
   - from a seed that encodes a given instance \(\phi\) of the problem and a possible solution in a predetermined format, an RTAS with \(T\) as its tile type set uniquely self-assembles a rectangular \(k\)-color pattern \(P_{\text{yes}}\) if \(\phi\) is a yes instance; and

\(^1\) In this paper, by pattern, we always mean a rectangular color pattern.
Fig. 1 Binary counter pattern assembly by an RTAS, based on the TAS design of Winfree (2000). Left Four tile types constitute the half-adder with the two inputs A, B from the west and south, the output S to the north, and the carryout C to the east. Right Copies of the “half-adder” tile types turn the L-shape seed into the binary counter pattern.

3. Design an auxiliary pattern **GADGET** such that
   - using $T$, an RTAS can self-assemble **GADGET**; and
   - in order for an RTAS with at most $n$ tile types to uniquely self-assemble **GADGET**, it is necessary for its tile type set to be equal to $T$ (modulo glue renaming).

4. Reduce $\phi$ to a rectangular $k$-color pattern $P(\phi)$ that mainly consists of $P_{\text{yes}}$ and **GADGET** and can be uniquely self-assembled by an RTAS using $T$ if $\phi$ is a yes instance.

Therefore, $T$ is sufficient for $\phi$ being a yes instance, but always insufficient for a no instance. More formally, if $\phi$ is a yes instance, then an RTAS can uniquely self-assemble $P(\phi)$ using $T$, and otherwise, no RTAS with at most $n$ tile types can uniquely self-assemble $P(\phi)$. Indeed, in order to uniquely self-assemble $P(\phi)$ with at most $n$ tile types, an RTAS cannot help but employ $T$ due to **GADGET**, but then the resulting assembly would provide a solution to make $\phi$ a yes instance ($T$ is designed to force this to happen). It hence holds that there exists an RTAS with at most $n$ tile types that uniquely self-assembles $P(\phi)$ if and only if $\phi$ is a yes instance. The **NP**-hardness of $k$-PATS is thus proved. This strategy also yields an inapproximability ratio

$$\frac{n + 1}{n}$$

for $k'$-PATS for any $k' \geq k$, i.e., there is no polynomial-time algorithm to output an RTAS that uniquely self-assembles a given pattern using fewer than $(1 + 1/n)$ times the minimum number of tile types used by an RTAS to achieve the same task. If $\phi$
being a no instance is proved to imply the necessity of $n + m$ tile types for $P(\phi)$ for some $m > 1$ instead of $n + 1$, then the unachievable ratio above is improved to $1 + \frac{m}{n}$.

The computer-assisted proof by Kari et al (2015a) has two drawbacks. One is a well-known critique of computer-assisted proofs in general, i.e., such a proof provides no insight into the unique minimality of $T$ for GADGET. Some readers of this paper may still (understandably) feel like a computer as they work through our proof, but at least our proof represents a guided proof-checking tour, which provides readers with “oracle access” to know exactly which parts of the design are needed for each small step. The second drawback is that combinatorial explosion prevents the scale-up of the brute-force check by their computer-assisted proof; their program ran for almost 1 CPU year in order to check “only” all possible sets of 13 tile types. As such, improving inapproximability ratios seems intractable by computer programs (it is worth mentioning that decreasing the denominator of (1) instead of increasing its nominator is not promising, as suggested in Culik and Kari (1997)).

A need thus arises for the design of a small tile type set $T$ working as an NP-hard problem verifier that balances the following goals:

- **Color Economy** $T$ should have as few colors as possible so that it can be used to prove the NP-hardness of $k$-PATS for small $k$;
- **Unique Minimality and its Manual Checkability** $T$ should have enough colors for the unique minimality of $T$ to hold for some pattern like GADGET, and moreover, this unique minimality can be checked without using computers.

These goals are competing. If all tile types in $T$ are colored white, then $T$ cannot be employed to produce patterns. Two colors are hence necessary, and actually sufficient for unique minimality and this sufficiency can be verified by computers as shown by Kari et al. (2015a). However, the manual-checkability has required the tile types to be drawn with more than two colors so far (Johnsen et al. 2013; Kari et al. 2015a; Seki 2013). On the other hand, with a large number of colors, the unique minimality and the manual checkability of this property are readily achieved, but the resulting $T$ cannot fulfill its intended purpose of proving the NP-hardness of $k$-PATS for as small $k$ as possible or the solution invisibility mentioned above. This invisibility requires $T$ to be colored in such a way that, being used by an RTAS as a verifier, the resulting pattern specifically suppresses any information about the solution to be verified. This requirement has so far prevented us from just redrawing the tile types used in the computer-assisted proof of Kari et al. (2015a) to turn their proof into a manually-checkable one.

For an NP-hard problem verifier that strikes a good balance among these goals, we propose a monotone 1-in-3-SAT verifier consisting of just 19 tile types. 1-in-3-SAT is a variant of 3SAT proved NP-hard by Schaefer (1978), in which one is to decide if there exists an assignment that makes exactly one (compare to “at least one” in 3SAT) of the three literals in each clause true, and the monotonicity promises an input to be free of negative literals.

By adding two more tile types to this verifier and coloring them properly with 11 colors, we design an extended monotone 1-in-3-SAT verifier (see Fig. 2). This verifier enables us to design a polynomial-time reduction from 1-in-3-SAT to 11-PATS whose correctness we can verify manually.
The 19 tile types verify an instance of 1-in-3-SAT. One can easily convert this verifier to a solver by adding just 2 more tile types for a total of 21 tile types. We can further convert the solver even to a 3-SAT solver with 5 more tile types. The resulting 3-SAT solver consists of just 26 tile types in total, and by a wide margin this outperforms known 3-SAT solvers such as the systems with respectively 147 and 64 tile types given by Brun (2008, 2012).

The remainder of this paper is organized as follows. Sect. 2 is for preliminaries. In Sect. 3, we engineer the monotone 1-in-3-SAT verifier. From the verifier, we will derive a 1-in-3-SAT solver and 3-SAT solver in Sect. 3.1. Section 4 is devoted to the proof of the NP-hardness of 11-pats. That is, we will give a manually-checkable proof based on the verifier for the NP-hardness of 11-pats. For the clarity of logic flow, combinatorial proofs of auxiliary lemmas for the main proof are included in Sect. 5. Section 6 concludes the paper by proposing open problems for future research.

2 Rectilinear TAS and constant-colored PATS

A (rectangular) pattern $P$ (of width $w$ and height $h$) is a function from the rectangular domain $\{(x, y) \mid x \in \{0, 1, \ldots, w - 1\}, y \in \{0, 1, \ldots, h - 1\}\}$ to a set of named colors, e.g., \{black, white\}. We denote the codomain of this pattern by $\text{color}(P)$, that is, $\text{color}(P)$ consists of the colors that appear at least once on $P$. We say that $P$ is $k$-colored if $|\text{color}(P)| \leq k$.

The self-assembly of a binary counter shown in Fig. 1 illustrates how a rectilinear TAS works. We now introduce some notations for the rectilinear TASs. A tile type is a square of some color whose four sides are each labeled with a glue. Assumed not to be rotatable, a tile type is identified by its color and four labels read in the counter-clockwise order starting at north ($N$); for instance, the second black tile type in Fig. 1 (Left) is $(1, 1, 0, 0, \text{black})$. Given a tile type $t$ and a direction $d \in \{N, W, S, E\}$, $t(d)$ denotes the label at the side $d$. A rectilinear TAS (RTAS, in short) is a pair $T = (T, \sigma_L)$ of a set $T$ of tile types and an L-shape seed $\sigma_L$ of width $w$ and height $h$ for some $w, h \geq 1$. The size of this RTAS is defined to be the cardinality of its tile type set, i.e., $|T|$. As shown in Fig. 1, the L-shape seed $\sigma_L$ is an assembly of tiles not included in $T$ so that its $x$-axis is provided with north labels and its $y$ axis is provided with east labels. Its domain is $\{(0, 0)\} \cup \{(x, 0) \mid 1 \leq x \leq w\} \cup \{(0, y) \mid 1 \leq y \leq h\}$. RTASs assume an infinite supply of copies of their tile types, each copy being referred to as a tile. With them, it tessellates the region $\{(x, y) \mid 1 \leq x \leq w, 1 \leq y \leq h\}$ delimited by the L-shape seed, according to the following rule:

**RTAS Tiling Rule** A tile can attach at a position $(x, y)$ if and only if its west label matches the east label of the tile on $(x - 1, y)$ and its south label matches the north label of the tile on $(x, y - 1)$.

Under this rule, a position does not become attachable until its west and south neighbor positions are tiled. Initially, the sole attachable position is hence $(1, 1)$. See the L-shape seed in Fig. 1 (Right); a tile of type $(1, 1, 0, 0, \text{black})$ can attach at $(1, 1)$, while label-mismatching prevents any tile of the other three types from attaching there. The attachment makes the two positions $(1, 2)$ and $(2, 1)$ attachable. In this manner, the
tiling proceeds from south-west to north-east rectilinearly until no attachable position is left. Since tile types are colored, if every position in the delimited domain has been tiled upon attachment termination, then the tiling shows a rectangular pattern and we consider it an output of the RTAS and call it a *terminal pattern*. The $5 \times 9$ binary counter pattern in Fig. 1 is terminal. If an RTAS admits a unique terminal pattern $P$, then we say that it uniquely self-assembles the pattern $P$.

In the binary counter example in Fig. 1, each attachable position admits a *unique* tile type whose copy (tile) can attach there, and we say that this RTAS is directed. Formally, an RTAS $(T, \sigma_L)$ is directed if for any distinct $t_1, t_2 \in T$, either $t_1(W) \neq t_2(W)$ or $t_1(S) \neq t_2(S)$ holds (the directedness of RTAS was originally defined in a different but equivalent way, see, e.g., Göös et al. (2014)). For technical convenience, we also say that such a tile type set $T$ is directed. Observe that a directed RTAS uniquely self-assembles a pattern as long as it can tile the region delimited by its seed.

The pattern self-assembly tile set synthesis (*pats*), proposed by Ma and Lombardi (2008), aims at computing the minimum size directed 2 RTAS that uniquely self-assembles a given rectangular color pattern. Recall that the size of an RTAS $(T, \sigma_L)$ is measured solely by the cardinality of $T$ and is independent of the seed. By restricting the number of colors allowed for input patterns, a practically meaningful subproblem of *pats* is formulated as follows.

**Definition 1** $k$-colored *pats* ($k$-pats)

**Given:** a $k$-colored pattern $P$.

**Find:** a smallest directed RTAS that uniquely self-assembles $P$.

### 3 Efficient design of a 1-IN-3-SAT verifier

The aim of this section is to give a small set of tile types with which an RTAS can verify an instance of monotone 1-IN-3-SAT. This set plays an essential role in our manually-checkable proof of the NP-hardness of 11-*pats* in Sect. 4.

See Fig. 2 (Left) for the monotone 1-IN-3-SAT verifier, which consists of 19 tile types (for now, ignore the yellow and blue tile types). It is designed in such a way that, starting from an L-shape seed that encodes a given monotone 1-IN-3-SAT instance $\phi$ over $m$ variables $v_1, v_2, \ldots, v_m$ and a Boolean assignment $b = (b_1, b_2, \ldots, b_m) \in \{F, T\}^m$ in a predetermined format on its glues, a directed RTAS with this tile type set evaluates $\phi$ according to $b$.

We describe the evaluation in detail now using an example. Let $\phi = (v_1 \lor v_2 \lor v_3) \land (v_1 \lor v_2 \lor v_4)$ and $b = (F, F, T, T)$, which satisfies $\phi$ in the 1-IN-3-SAT sense$^3$. See Fig. 3 for the evaluation of $\phi$ according to $b$ by the RTAS.

The L-shape seed serves as an input interface for the RTAS. Clauses of $\phi$ are written on the seed’s $x$-axis as a sequence of glues $\lor$ (variable in clause), $\land$ (variable not in

---

$^2$ Unlike the original definition in Ma and Lombardi (2008), a solution to *pats* is required to be directed here, but this difference does not change the problem as a minimum RTAS is always directed (Göös et al. 2014).

$^3$ In contrast, for example, $(T, F, T, F)$ does not satisfy $\phi$ in the 1-IN-3-SAT sense because it satisfies more than one literal of the first clause of $\phi$. 
A 1-IN-3-SAT verifier consisting of 19 tile types of 9 colors: cyan (4), CE (3), white (2), black (2), DGNL-white (2), DGNL-black (2), Init (2), Sat (1), and red (1), where the numbers in parentheses indicate how many tile types in $T_{val}$ are colored with the corresponding colors. With the yellow and blue tile types on the right, these 19 tile types constitute a set $T_{eval}$ of 21 tile types of 11 colors, which is employed in our proof for the NP-hardness of 11-pats (Color figure online).

The clauses $(v_1 \lor v_2 \lor v_3)$ and $(v_1 \lor v_2 \lor v_4)$ of $\phi$, for instance, are first converted into $vvvn$ and $vvnv$, respectively. We then pre-pad each of them from the left by $h$ glues for some $h \geq 0$ so that the verification starts at the height $h$ (this offset $h$ will be used in the reduction in Sect. 4), and finally post-pad them with an incremental number of $n$ glues so that a clause is evaluated on the row just above those on which previous clauses were evaluated. The resulting encodings of the two clauses of $\phi$ are $n^h v v v n^0$ and $n^h v v n v n^1$, respectively, where the $n$’s for padding are underlined. Connecting them by $c$’s, putting $c$ at the beginning and at the end, and then further pre-padding the resulting encoding with as many $n$’s as variables involved (in this example, 4 $n$’s are pre-padded as $\phi$ involves the four variables $v_1, v_2, v_3, v_4$) results in $n^4 c n^h v v v n n^0 c n^h v v n v n c$. This string is the encoding of the clauses of $\phi$.

The assignment $b$ is written rather on the seed’s y-axis as FFTT as the assignment to the first variable $v_1$ comes at the bottom. We post-pad it with as many $F$’s as the number of clauses of $\phi$ like FFTT-F$^2$ for this example. When used in the reduction, it is pre-padded also by some $h$ glues for the sake of GADGET; this will be explained in Sect. 4. Note that in Fig. 3, the assignment is lower-cased ($F/T \rightarrow f/t$) as a preprocess. It is this purpose that the pre-padded 4 $n$’s on the $x$-axis encoding serve for, but this preprocess is irrelevant for now and will be addressed later in Fig. 9.

Signals $v$ and $n$, carrying information about the membership of variables in clauses, are propagated northward through black and white tiles (2 types each), respectively. The clauses become visible to us in this way. Cyan tiles (4 types) propagate Boolean signals ($F/T$) horizontally as well as vertically.

---

4 $n$ does not denote a negated variable. Recall that monotonicity requires that variables are never negated in clauses.
Fig. 3 Starting from the L-shape seed, indicated by gray tiles, that encodes the instance $\phi = (v_1 \lor v_2 \lor v_3) \land (v_1 \lor v_2 \lor v_4)$ and an assignment $b = (F, F, T, T)$, a directed RTAS evaluates $\phi$ according to $b$ using tiles of the verifier in Fig. 2 (Left). The assembly results in the subpattern which we refer to as CIRCUIT in Sect. 4 and is located to the northeast of GADGET on $P(\phi)$ (Color figure online)
At the crossover of these signals, variables are evaluated diagonally by DGNL-black tiles (2 types); like a diagonal mirror, they reflect the signals from the west (the assignment) to the north. The three signals thus evaluated per clause are propagated to the north via cyan tiles and then evaluated by CE tiles (3 types). At an encounter with a T signal, CE tiles change the evaluation from f to s (satisfied), and without another encounter with T signal, CE tiles propagate the evaluation to the east until it is validated by a Sat tile at the top of Init column, which initializes the assignment signals for the validation of the next clause. The post-padding enables clauses to be evaluated on different rows.

See Fig. 3 for the emerging pattern CIRCUIT. Note the invisibility of the assignment b on the pattern. The assignment can be retrieved only by examining its underlying assembly and in particular its glues (and cannot be retrieved by observing the color pattern only). In fact, from two seeds that encode different satisfying assignments in the above format, tiles in the verifier assemble the same pattern CIRCUIT. Note that, starting from the seed that encodes an unsatisfying assignment, the RTAS cannot complete any rectangular pattern due to the lack of the UNSAT counterpart of SAT tile type or the CE tile type receiving s from the west and T from the south to handle a second true literal in the verifier.

### 3.1 1-IN-3-SAT solver and 3-SAT solver

Before applying this verifier for the NP-hardness proof, we briefly describe a way to transform this verifier to a solver for 1-IN-3-SAT and for 3-SAT. Readers can skip this subsection because what we discuss here is irrelevant to the NP-hardness proof. We replace the encoding of a specific assignment to the y-axis (FFTT in the above example) with m I’s for some new glue I (input). Then we introduce the following two tile types $t_F$ and $t_T$:

![Tiles of these types attach on the first column nondeterministically, which amounts to the nondeterministic guess of an assignment.](image)

Tiles of these types attach on the first column nondeterministically, which amounts to the nondeterministic guess of an assignment.

The resulting monotone 1-IN-3-SAT solver can be further modified to solve even instances of 3-SAT. The differences between monotone 1-IN-3-SAT and 3-SAT are the existence of negative literals and the satisfiability condition (at least one instead of exact). The modification requires the following five tile types:

![The first four are the analogues of DGNL-black and black tile types for negative literals. They allow, for instance, a clause $(v_1 \lor \overline{v_2} \lor v_4)$ to be encoded as $\overline{v} v n \overline{v}$ on the seed.](image)

The yellow tile type allows a Sat tile to attach at the prescribed position even when more than one literal of a clause becomes true.
4 Manually-checkable proof for the NP-hardness of 11-PATS

Using the verifier engineered in Sect. 3, we now give a polynomial-time reduction from monotone 1-in-3-SAT to 11-PATS.

The reduction employs the strategy mentioned in the Introduction. We augment the 1-in-3-SAT verifier of 19 tile types in Fig. 2 (Left) with 2 extra tile types (yellow and blue) in Fig. 2 (Right) and employ the resulting extended set $T_{\text{eval}}$ of 21 tile types as a verifier.

A given instance $\phi$ of monotone 1-in-3-SAT is converted to an 11-colored rectangular pattern $P(\phi)$ consisting of primary and secondary subpatterns, as blueprinted in Fig. 4. The primary subpattern CIRCUIT is the assembly region where $\phi$ is thus validated (evaluated to be true) by the verifier according to some satisfying assignment $b$. Unless $\phi$ is satisfiable, no assignment exists. The secondary subpattern GADGET plays a critical role in the reduction due to its following property:

Property 1 If a directed RTAS $(T, \sigma_L)$ with some set $T$ of at most 21 tile types uniquely self-assembles a pattern including GADGET, then $T$ must be isomorphic to $T_{\text{eval}}$ (modulo glue renaming). Therefore, no set of strictly less than 21 tile types can be employed to uniquely self-assemble the pattern.

With GADGET included in the reduced pattern $P(\phi)$, Property 1 forces a directed RTAS to employ $T_{\text{eval}}$ in order to uniquely self-assemble $P(\phi)$, if only 21 tile types are available. Moreover, tiles in $T_{\text{eval}}$ require an assignment satisfying $\phi$ to assemble the primary subpattern CIRCUIT of $P(\phi)$. Consequently, $\phi$ is satisfiable if and only if $P(\phi)$ is uniquely self-assembled by a directed RTAS with at most 21 tile types.

Having described informally how the reduction works, we explain it in detail in the remainder of the paper.
4.1 Secondary subpattern GADGET

We have shown that if $\phi$ is satisfiable, then a directed RTAS can self-assemble CIRCUIT using $T_{\text{eval}}$. Figures 5 through 9 illustrate how $T_{\text{eval}}$ self-assembles other parts of $P(\phi)$ (see in Fig. 4 how those parts are integrated into $P(\phi)$). Examining these figures in detail is sufficient to be convinced that if $\phi$ is satisfiable, then a directed RTAS uniquely self-assembles the pattern $P(\phi)$ using $T_{\text{eval}}$.

It is much harder to prove the converse that if a directed RTAS with at most 21 tile types uniquely self-assembles $P(\phi)$, then $\phi$ is satisfiable. This difficulty is primarily due to the huge number of possible tile type sets as well as possible seeds for the

![GADGET Diagram](image-url)

**Fig. 5** The leftmost part of the secondary subpattern GADGET of the reduced pattern $P(\phi)$. The constants $c$ and $r$, which are independent of $\phi$, are set large enough to make the proof work (Color figure online)
RTAS. The role of GADGET is to make all tile type sets but $T_{eval}$ useless (Property 1). That is, with at most 21 tile types available, the RTAS must employ $T_{eval}$ to uniquely self-assemble $P(\phi)$. The RTAS still has the freedom of choice in its seed. However, at the top of the y-axis, the seed’s glues must be of the form $(F/T)^m F^k$ (see Fig. 9), where $m$ and $k$ are the numbers of variables and clauses in $\phi$, respectively. This is because the west glue of the cyan tiles in $T_{eval}$ is either $F$ or $T$ and that of the red(F) tile type is $F$. Each glue sequence for $(F/T)^m$ among all possible $2^m$ candidates corresponds to a Boolean assignment to the $m$ variables of $\phi$. The above-mentioned invisibility of the assignment allows the RTAS to make this choice, but the chosen one must satisfy $\phi$ in order to assemble CIRCUIT of $P(\phi)$ completely. Thus, $\phi$ is satisfiable. The proof of NP-hardness of 11-pats is completed in this way.

Before verifying Property 1 in Sect. 4.2, we explain the structure of GADGET and how it is integrated, together with CIRCUIT, into the pattern $P(\phi)$. GADGET is composed of three parts: leftmost part including an important subpattern LB4 (Fig. 5), the middle one (Fig. 6), and the rightmost one. The subpattern LB4 is parameterized by two constants $c$ and $r$, which are set large enough for the sake of our proof of Lemma 2 below (their actual values are specified at the beginning of the proof). Note that these constants are independent of the size and clauses of $\phi$. The rightmost part mainly consists of the sixteen instances of a subpattern template shown in Fig. 10 (Left) and their sixteen black analogues. We group these thirty-two subpatterns into eight categories by their color on diagonal ($DGNL$-white or $DGNL$-black) and the Boolean values of the bottom two positions ($FF$, $FT$, $TF$, or $TT$), bundle the 4 subpatterns in each category into a subpattern, and name the resulting eight subpatterns like $DGNLwFF$ (omitting $DGNL$).

GADGET is carefully designed so that, being assembled from tiles in $T_{eval}$, it exposes

- only $F$ glues to the north; and
- only $f/t$ glues to the east, except at the top where the glue is $F$.

The north $F$ glues enable cyan tiles to attach to their north and propagate the assignment above GADGET toward CIRCUIT invisibly. With $m$ $n$ glues on the $x$ axis of the seed,\(^5\) the east $f/t$ glues let white tiles assemble the foundation of $JOINT$ on which $DGNL$-white tiles attach diagonally in collaboration with cyan and white tiles and lower-case the assignment signals $(F/T \rightarrow f/t)$ (see Fig. 9). CIRCUIT and GADGET are thus integrated into the pattern $P(\phi)$.

### 4.2 Verification of Property 1

This subsection verifies Property 1, and hence, concludes the proof of the NP-hardness of 11-pats. The verification is done through the following task: given 21 tile types

---

\(^5\) Recall that $m$ is the number of variables involved in $\phi$. 
which have not been colored or labelled yet, color and label them so that, using the resulting tile type set, a directed RTAS can uniquely self-assemble GADGET.

4.2.1 Coloring

We handle coloring first. We will observe that the given 21 tile types must be colored as $T_{\text{eval}}$ does, i.e., 4 cyan, 3 CE, 2 white, DGNL-white, black, DGNL-black, Init each, and 1 Sat, yellow, red (F), and blue (T) each. In fact, we only have room to choose colors for 10 of them because with each color, at least one tile type must be colored.

We begin with the need for one more Init tile type. For the sake of contradiction, suppose there were only one Init tile type. See the rightmost column in Fig. 6. At
its bottom, 2 red (F) and $2r-1$ blue (T) positions are found, and on top of them is one more red position (at the height $2r+2$). Since their western neighbors are all $\text{Init}$, with only one $\text{Init}$ tile type, a directed RTAS would need to fill the blue positions with $2r-1$ tiles of pairwise distinct types in order to attach a red tile precisely at the height $2r+2$ (the hardcoded height). This would cost the RTAS an unaffordable $2r-2$ extra blue tile types (recall that $r$ was set large enough). Thus, we need to color one uncolored tile type by $\text{Init}$ and 9 tile types remain uncolored.

To their west is a white column (the third from the right). With only one white tile type, we find that the red position on top of the $2r-1$ blue positions must again be hardcoded from below through the $\text{Init}$ and red (F)/blue (T) columns. However, this is unaffordable, provided $r$ is set sufficiently large. The same argument based on the fourth, fifth, and sixth leftmost columns in Fig. 6 justifies the need of at least 2 black tile types. Among the 9 uncolored tile types, one has been colored white and another has been colored black. As a result, 7 tile types remain uncolored.

Before coloring them, we present one lemma on $\text{Init}$, white, and black tile types.

**Lemma 1** Let $\text{col} \in \{\text{Init}, \text{white}, \text{black}\}$. If a directed RTAS with at most 21 tile types including exactly 2 tile types $t_1, t_2$ of color $\text{col}$ uniquely self-assembles a pattern that includes GADGET, then $t_1(\emptyset) \neq t_2(\emptyset), t_1(\text{E}) \neq t_2(\text{E})$, and $t_1(\text{S}) = t_2(\text{S})$.

**Proof** We prove this lemma only for $\text{col} = \text{Init}$ (analogous arguments work for the other two colors). The need for $t_1(\text{E}) \neq t_2(\text{E})$ has been already seen; otherwise hardcoding would be necessary in order to place the red tile at the specific height.
Suppose \( t_1(S) \neq t_2(S) \). This distinctness forces the RTAS to assemble the second rightmost column in Fig. 6 periodically either as \( t_1t_2t_1t_2 \cdots \) or as \( t_1t_2t_1 \cdots \). In any case, the column exposes a periodic sequence of east glues, and hence, the placement of the red tile at the specific height would require an unaffordable cost in hardcoding by blue tile types. Therefore, \( t_1(S) = t_2(S) \) must hold, and this implies \( t_1(\overline{W}) \neq t_2(\overline{W}) \) in order for the RTAS to be directed. □

Next, we focus on cyan tiles. As of now, just 1 tile type was colored cyan. We will show that due to the subpattern LB4 in Fig. 5, designated by a dotted rectangle, we need either 3 more cyan tile types, or 2 more cyan tile types and 2 more tile types whose colors are either red (F) or blue (T). The latter costs one more extra tile type, which will turn out unaffordable later.

**Lemma 2** If a directed RTAS with 21 tile types uniquely self-assembles a pattern including GADGET, then it has either

1. at least 4 cyan tile types, or
2. the 3 cyan tile types shown in Fig. 11, 1 red (F) tile type, 1 blue (T) tile type, and 2 tile types whose color is either red (F) or blue (T).

Our proof of this lemma is so technical that presenting it at this point may distract the reader's attention from the essence of the reduction. Its proof is in Section 5.1. For the sake of argument to deny the second choice in Lemma 2 later, we briefly observe how the 3 cyan tiles $A$, $B$, $C$ in the choice deliver signals. As shown in Fig. 11 (Right), $B$ and $C$ tiles alternately attach and deliver signals in a zigzag manner. Note that they cannot expose two 1 glues to the east consecutively; a 1 glue is vertically sandwiched by 0 glues. This is essential for preventing GADGET from being assembled as long as the second choice is made.
Fig. 11  Left The only set of 3 cyan tile types with which one can self-assemble the subpattern LB4. Right B and C tiles deliver a signal via b and 1 glues in a zigzag manner toward northeast (Color figure online)

Fig. 12  This subpattern of GADGET can assemble in this way using the 2 CE tile types and 2 yellow tile types shown here (Color figure online)

Among the 7 uncolored tile types, the first choice in Lemma 2 colors 3 of them by cyan, while the second choice colors 4 of them. Now note that, not depending on which choice was made, we must color one of the uncolored tile types by CE and another by CE or yellow. Just above LB4, we find six yellow positions stacked vertically with a CE position on top of them, and to their west is a pillar of CE’s. Note that we have colored only one tile type by yellow so far, and there are at most 4 tile types left uncolored. With only one CE type, no directed RTAS could put a CE tile at the top of the six yellow tiles. One uncolored tile type is to be colored by CE.

The next lemma shows that at least one of the uncolored tile types must be colored with either CE or yellow. Its proof is in Sect. 5.2.

**Lemma 3** If a directed RTAS with at most 21 tile types uniquely self-assembles a pattern including GADGET, then it contains at least 2 CE tile types and the total number of CE and yellow tile types is at least 4. Moreover, if it contains exactly 2 CE tile types \( t_1, t_2 \) and exactly 2 yellow tile types \( t_3, t_4 \), then these four tile types are labelled as depicted at the right bottom of Fig. 12.

This lemma indicates one non-isomorphic way to color 4 tile types and assign them with glues so that the resulting tiles uniquely self-assemble the pattern in Fig. 12, which is a subpattern of GADGET, shown in Fig. 5. However, this situation shall be proven improper in order for a directed RTAS with 21 tile types to self-assemble the
whole \textit{GADGET} in the end. In any case, this lemma implies that among the at most 6 uncolored tile types, one must be colored either CE (expected) or yellow (unexpected) (Fig. 13).

In the figure below, we summarize pictorially how the 21 tile types have been colored so far, where a dotted square indicates an uncolored tile type.

![Figure showing colored tile types]

We next exclude the second choice of Lemma 2. For the sake of contradiction, suppose that with this choice, a directed RTAS could self-assemble \textit{GADGET}. Then as of now, only one tile type remains uncolored, and hence, one of the following statements must hold:

- there are only 1 DGNL-white and 2 white tile types;
- there are only 1 DGNL-black and 2 black tile types.

The 16 subpatterns of \textit{GADGET} in Fig. 10 play a role in denying the first statement. Consider the task for the RTAS to assemble these 16 subpatterns with only 1 DGNL-white and 2 white tile types. Their assemblies are trivially identical at the main diagonal consisting of four DGNL-white positions (1, 1) - (4, 4). Recall that the 2 white tile types have distinct west glues (Lemma 1). Hence, all white positions on the first diagonal below the main diagonal are filled with tiles of the same type. This argument works also for the second and third diagonal below the main one. As a result, the 16 assemblies are identical with respect to their fourth column from the left. Since the RTAS is directed, this means that the types of tiles at the bottom of the rightmost two columns (Init and red(F)/blue(T)) completely determine which of the 16 subpatterns emerges. However, even with coloring the last uncolored tile type with Init, at most 12 (= 3 × 4) combinations of types would be possible, that is, four of the 16 subpatterns could never assemble, reaching a contradiction. Likewise, the second statement is denied by the black analogue of these 16 subpatterns. The second choice of Lemma 2 has been thus excluded. As a result, the 21 tile types have been colored partially as follows:

Now we complete the coloring by proving that one of the remaining 2 uncolored tile types must be colored DGNL-white and the other DGNL-black. For the sake of contradiction, suppose only one DGNL-white tile type is available. Among the 16 instances of the template shown in Fig. 10 (Left), consider the eight whose right bottom corner is Init-red(F). With only two white tile types, as argued just above, the types of the Init and red(F) tiles attaching there completely determine which of the possible 8 red(F)-blue(T) patterns assembles above. However, no matter how we color

\begin{footnote}
If the second choice in Lemma 2 is made, then there are only 2 tile types left uncolored at this point.
\end{footnote}
the remaining 2 uncolored tile types, the number of combinations of Init tile types and red(F) tile types cannot exceed 6, and hence, at least 2 of the 8 subpatterns could not be assembled, reaching a contradiction. Hence, we cannot do without coloring one more tile type by white. This means that either there is only one red tile type or there is only one blue tile type. Consider the first case. See Fig. 13 for parts of four instances.

At their northeast corner, we find all of FF, FT, TF, and TT (they are vertically aligned), and which of them appears is completely determined by how the downward-diagonal consisting of the top-left DGNL-white position, middle Init position, and bottom-right red (F) position assembles. For that, 4 Init tile types are required, but there are at most 3 Init tile types available, reaching a contradiction. The argument based on the blue analogues of the subpatterns leads to the same contradiction, provided there is only one blue tile type. Consequently, we need to color one of the 2 uncolored tile type by DGNL-white. Based on the 16 instances of the black analogue of the template, on which white and DGNL-white positions are colored black and DGNL-black, respectively, the argument above creates the need for one more DGNL-black tile type.

We have proved that if a directed RTAS with at most 21 tile types uniquely self-assembles a pattern including GADGET, then the tile types must be colored as:

We will show that the color of the last one must be CE in the next subsection.

### 4.2.2 Glue assignment

Having colored the 21 tile types almost completely, we discuss how to assign glues to the 21 tile types so that the directed RTAS with the resulting tile type set can uniquely self-assemble a pattern including GADGET.

First, we determine the glue assignment of tile types which do not share their color with another tile type, that is, the Sat, red (F), and blue (T) tile types. We denote these tile types by $t_{Sat}$, $t_F$, and $t_T$, respectively. All Sat, red, and blue positions on GADGET...
are filled with $t_{\text{Sat}}, t_F$, and $t_T$ tiles, respectively. On GADGET, red and blue positions are vertically stacked so that $t_F(N) = t_F(S) = t_T(N) = t_T(S) = c$ for some glue $c$. For the sake of directedness, this forces $t_F(W) = F$ and $t_T(W) = T$ for some distinct
glues $F$, $T$. A $\text{Sat}$ position is found to the north and to the west of a red position (see Fig. 6) so that $t_{\text{Sat}}(S) = t_F(N) = c$ and $t_{\text{Sat}}(E) = t_F(W) = F$. Sharing the south glue with $t_F$ and $t_T$, $t_{\text{Sat}}(W)$ must differ from $t_F(W)$ or $t_T(W)$ for the sake of directedness; let $t_{\text{Sat}}(W) = s$ for some new glue $s$. Their glues have been partially determined as follows:

Next we show how the 2 $\text{Init}$ tile types, which we denote by $t_{\text{Init}F}$ and $t_{\text{Init}T}$, are assigned with glues. See Fig. 6, where we find a column of $\text{Init}$ positions sandwiched by two columns of red and blue positions, at which $t_F$ and $t_T$ tiles attach, respectively. Thus, without loss of generality, the tile type at an $\text{Init}$ position between red positions is $t_{\text{Init}F}$ while the tile type at an $\text{Init}$ position between blue positions is $t_{\text{Init}T}$. This implies $t_{\text{Init}F}(N) = t_{\text{Init}F}(S) = t_{\text{Init}T}(N) = t_{\text{Init}T}(S)$. This glue is actually $c$ because a red position is found on top of the $\text{Init}$ column. For the sake of directedness, we need new glues $\hat{f}$, $\hat{t} \neq s$, $F$, $T$ as respective west glues of $t_{\text{Init}F}$ and $t_{\text{Init}T}$. Now the three horizontally adjacent positions red-$\text{Init}$-red imply $t_F(E) = t_{\text{Init}F}(W) = \hat{f}$ and $t_{\text{Init}F}(E) = t_F(W) = F$. Similarly, $t_T(E) = t_{\text{Init}T}(W) = \hat{t}$ and $t_{\text{Init}T}(E) = t_T(W) = T$. The glues of the 5 tile types have been thus partially determined as follows:

We apply the same argument on a white column adjacent to an $\text{Init}$ column which is sandwiched by red(F)/blue(T) columns (see Fig. 6) to assign the two white tile types $t_{\text{Sat}F}$, $t_{\text{Sat}T}$ with glues as $t_{\text{Sat}F}(W) = t_{\text{Sat}F}(E) = \hat{f}$, $t_{\text{Sat}T}(W) = t_{\text{Sat}T}(E) = \hat{t}$, and $t_{\text{Sat}F}(S) = t_{\text{Sat}T}(S) = n$ for some glue $n$, which must differ from $c$ for directedness. Now the black column in Fig. 6 enforces the following glue assignment to the two black tile types $t_{bf}$ and $t_{bt}$:

where the glue $v$ must differ from $n$.

As seen above, tile types of a unique color is useful for determining the glue assignment. Recall the tile type whose color was not determined but just proved to be either CE or yellow. We now prove that it must be colored CE, and the tile type set contains only 1 yellow tile type. See the bottom left corner of Fig. 5, where there is the pattern red(F)-CE-$\text{Sat}$. Without the third type, CE tiles just let a 1-bit signal pass through from west to east (Lemma 3) without changing, but then this pattern would
imply the contradictory equation $f = s$ (recall that these glues must be distinct; otherwise, the $Sat$ tile type would share both its west and the south glues with an $Init$ tile type, that is, the RTAS would not be directed any more). Thus, we must color the tile type by $CE$.

Now the tile type set contains only one yellow tile type $t_y$. See Fig. 5, in which yellow positions are adjacent to each other horizontally and vertically and a yellow position is to the west of a $Sat$ position at the bottom of $LB4$. Thus, we have $t_y(W) = t_y(E) = t_{Sat}(W) = s$ and $t_y(N) = t_y(S)$, and moreover, $t_y(S) \neq t_{Sat}(S) = c$ must hold since $t_y$ and $t_{Sat}$ have the same west glue $s$. Let $t_y(S) = T$ for some glue $T \neq c$.

Note that, at this point, we cannot exclude the possibility that the south glue $T$ is equal to $n$ or $v$. It is not until the glue assignment of all the 21 tile types is completely determined at the end of this section that we can distinguish $T$ from them.

We now shift the attention to the glue assignment to the 3 $CE$ tile types. Since a $CE$ position is horizontally sandwiched by two yellow positions in Fig. 5, one $CE$ tile type, say $t_{CEss}$, has $s$ glues on its west and east edges. Thus, its south glue must differ from those of the yellow and $Sat$ tile types. Let $t_{CEss}(S) = F$ for some glue $F \neq T, c$. Note that like the glue $T$, the distinction of $F$ from $n$ or $v$ will not be made until the end of this section. In Fig. 6, we find positions of the pattern red($F$)-$CE$-$Init$-red. At the red positions, $t_F$ tiles attach, so the type of a tile attaching at the $Init$ position is $t_{InitF}$. Thus, the $CE$ tile there must have the glue $f$ at both of its west and east sides, and hence, the $CE$ tile cannot be of type $t_{CEss}$. We denote its type by $t_{CEff}$. Then $t_{CEff}(W) = t_{CEff}(E) = f$.

On the top row of $LB4$ in Fig. 5, there is a pattern red($F$)-$CE$-$CE$-yellow. The tile types at the red and yellow positions are $t_F$ and $t_{yellow}$, respectively, $t_F(E) = f$, and $t_{yellow}(W) = s$. In order to assemble this pattern, therefore, we need the third $CE$ tile type $t_{CEfs}$ with $t_{CEfs}(W) = f$ and $t_{CEfs}(E) = s$. See the figure below.

The north and south glues of the above three tile types are determined here. See Fig. 5, where we find yellow-$CE$-$CE$-yellow positions self-stacked vertically. Tiles with color $CE$ attaching there must have $s$ glues on their west and east edges, and hence are of type $t_{CEss}$. Thus, $t_{CEss}(N) = t_{CEss}(S) = F$. In the same figure, we find a $CE$ position whose east and south neighbors are yellow. A $t_{CEss}$ tile cannot attach there due to its south glue mismatch; neither can a $t_{CEff}$ tile due to its east glue mismatch. The remaining type $t_{CEfs}$ must be assigned with glues properly so as for a $t_{CEfs}$ tile to attach there. Thus, $t_{CEfs}(S) = T$. See the figure below.
In Fig. 5, we can see three vertically-stacked CE positions sandwiched horizontally by yellow positions. Hence, $t_{\text{CEss}}$ tiles attach there. Denote the position of the CE-color tile above them by $(x, y)$. Since its eastern neighbor is yellow, the type of CE tile attaching there must be also $t_{\text{CEss}}$. A $t_{\text{CEff}}$ tile attaches to its north neighbor position $(x, y + 1)$ and a $t_{\text{CEfs}}$ tile attaches to its western neighbor position $(x - 1, y)$. For these last two placements, observe that the type of tiles at all CE positions just above the stair-like yellow positions is $t_{\text{CEfs}}$, and tiles at all the consecutive CE positions to the west of $t_{\text{CEfs}}$ must be of type $t_{\text{CEff}}$. Around $(x, y)$, tiles assemble as follows:

\[
\begin{array}{cccc}
  & \text{CEff} & \text{CEff} & \text{CEff} \\
  & \text{CEfs} & \text{CEss} & \text{CEss} \\
  & \text{CEss} & \text{CEss} & \text{CEss} \\
\end{array}
\]

where the position $(x, y)$ is indicated by the box. Thus, $t_{\text{CEff}}(S) = F$, which in turn gives $t_{\text{CEff}}(N) = t_{\text{CEfs}}(N) = F$. So the glues of the 3 CE tile types have been determined completely as follows:

\[
\begin{array}{cccc}
  & \text{CEff} & \text{CEff} & \text{CEfs} \\
  & \text{CEff} & \text{CEfs} & \text{CEss} \\
  & \text{CEfs} & \text{CEss} & \text{CEss} \\
  & \text{CEfs} & \text{CEss} & \text{CEss} \\
\end{array}
\]

These three tile types have the following useful property.

\textbf{Property 2} Let $x, y \in \mathbb{N}_0$ and $d \geq 1$. If $\text{GADGET}(x, y)$ is yellow, and for all $1 \leq i \leq d$, $\text{GADGET}(x+i, y)$ is CE, then the tile type at $(x+d, y)$ is $t_{\text{CEss}}$.

Before proceeding to the glue assignment of the remaining 8 tile types (4 cyan, 2 DGNL-white, and 2 DGNL-black), we determine the north glue of $t_{\text{Sat}}$. In Fig. 6, there is one Sat position whose north neighbor is CE and whose northwestern neighbor is yellow. Due to Property 2, the tile type attaching at this CE position is $t_{\text{CEss}}$, and hence, $t_{\text{Sat}}(N) = t_{\text{CEss}}(S) = F$. We summarize all the 13 tile types whose glues are completely determined so far as follows:

\[
\begin{array}{cccc}
  & \text{F} & \text{CEff} & \text{CEfs} \\
  & \text{c} & \text{c} & \text{f} \\
\end{array}
\]

\[
\begin{array}{cccc}
  & \text{c} & \text{f} & \text{Sat} \\
  & \text{c} & \text{c} & \text{c} \\
\end{array}
\]

\[
\begin{array}{cccc}
  & \text{c} & \text{f} & \text{Init} \\
  & \text{c} & \text{c} & \text{c} \\
\end{array}
\]

\[
\begin{array}{cccc}
  & \text{CE} & \text{CE} & \text{CE} \\
  & \text{v} & \text{v} & \text{v} \\
  & \text{v} & \text{v} & \text{v} \\
\end{array}
\]

We now determine the glues of 4 cyan tile types. First, see the tenth column from the right in Fig. 6, on which there is a cyan position surrounded by CE’s, Sat, and red
positions. Due to Property 2 and the fact that the north glue of any CE tile is $F$, the tile attaching there must have $F$ glues on all of its four sides as:

\[
\begin{array}{c|c|c}
F & F & F \\
\hline
\text{t}_{sbFF} & F & F \\
\end{array}
\]

We denote this type by $t_{sbFF}$.

Focus on the horizontal tandem of cyan positions just above LB4 in Fig. 5. At its first and second positions, $t_{sbFF}$ tiles attach. The tile attaching at the third position must have $F$ glue on its west side and $T$ glues on its north and south sides, and hence, it is not of type $t_{sbFF}$. We denote its type by $t_{sbFT}$, which is assigned with glues partially as follows.

At the southwestern corner of LB4, a $t_{sbFF}$ tile attaches, and hence, the tile attaching to its north must have west glue $T$ and south glue $F$. Thus, the tile is of type neither $t_{sbFF}$ nor $t_{sbFT}$; denote it as $t_{sbTF}$. Also denote the fourth cyan tile type as $t_{sbTT}$. The four cyan tile types have been assigned with glues partially so far as follows.

We claim that $t_{sbTT}(W) = T$. Suppose this is not true. Then $t_{sbTF}$ would be the sole cyan tile type whose west glue is $T$. See Fig. 6, where there is a pattern blue–white–Init–cyan–blue, and it assembles as $T_w$–$t_{wt}$–$t_{Init}$–cyan–$t_T$. This raises a need for a cyan tile type whose west and east glues are both $T$. Hence, $t_{sbTF}(E) = T$. Then at the northwestern corner of LB4, two cyan tiles of this type attach and expose the same glues to the north. Accordingly, the assembly at the CE positions to their north is $t_{CEff}^2$ or $t_{CEss}^2$, but either case would cause a glue mismatch with the red tile to the west or yellow tile to the east, reaching a contradiction. The claim $t_{sbTT}(W) = T$ has been thus verified.

On the ninth column of Fig. 6, we find a cyan position surrounded by yellow, Sat, and red positions from the north, west, and east, respectively. The north, west, and east glues of the tile attaching there must be $T$, $F$, $F$, respectively. Because $t_{sbFF}(N) = F$ and $t_{sbTF}(W) = t_{sbTT}(W) = T$, this tile’s type is $t_{sbFT}$. This gives $t_{sbFT}(E) = F$, as illustrated below.

See the cyan positions below the $t_{sbFT}$ tile. The tile attaching just below it must have north glue $T$ and west and east glues $F$, and hence, it is also of the type $t_{sbFT}$.
this way, we determine that at the top five positions of this cyan ninth column, \(t_{sbFT}\) tiles attach. Consider the sixth position from the top. The tile attaching there must have \(T\) glues on its north, west, and east sides. Hence, it is of type either \(t_{sbTF}\) or \(t_{sbTT}\) and has north glue \(T\). We next identify another cyan position at which a tile of one of these types must attach and moreover its north glue must be \(F\). Due to the requirement of different north glues, cyan tiles attaching at these positions must be of different types. Such a cyan position is found at the northeastern corner of \(LB4\). Its north neighbor is \(CE\) and its east neighbor is blue. Due to Property 2, the tile attaching there must have north glue \(F\) and east glue \(T\), and hence, is of type \(t_{sbTF}\) or \(t_{sbTT}\). As such, \(t_{sbTF}\) and \(t_{sbTT}\) tiles attach at these positions exclusively, and hence, we have:

\[
- t_{sbTF}(E) = t_{sbTT}(E) = T, \quad \text{and}
- \{t_{sbTF}(N), t_{sbTT}(N)\} = \{F, T\}.
\]

The latter equality means that the north glue of any cyan tile is \(F\) or \(T\). As a result, \(t_{sbTT}(S)\) must be either \(F\) or \(T\) because below these positions are cyan positions but it actually must be \(T\) for the sake of directedness. The four cyan tile types have been assigned with glues partially so far as follows.

\[
\begin{array}{c|c|c}
\text{F} & t_{sbFF} & \text{F} \\
\text{F} & t_{sbFT} & \text{F} \\
\text{T} & t_{sbTF} & \text{T} \\
\text{T} & t_{sbTT} & \text{T}
\end{array}
\]

Suppose that the north glue of \(t_{sbTF}\) is \(T\) and that of \(t_{sbTT}\) is \(F\). Using tiles of these types, however, we cannot assemble \(LB4\). These spurious tile types have the following property:

\[
- F/T \text{ signals are propagated horizontally;}
- \text{an F/T signal is propagated from the south to the north when crossing a horizontal F signal; and}
- \text{an F/T signal is flipped when crossing a horizontal T signal.}
\]

Due to the first statement in this property, the cyan portion of the fourth column of \(LB4\) exposes the following sequence of glues to the east: \(F_T^{2r-1} F_T^{2r-1} T_T\) (from the bottom to the top), which is the same as the one exposed to the east by the \(\text{Init}\) portion of the second column of \(LB4\). The north glue \(T\) of the yellow tile attaching at the bottom of the fifth column of \(LB4\) crosses an odd number of \(T\) signals while propagating northward, and is flipped and exposed as \(F\) to the top yellow position of the column. The \(T\) glue at the south prevents a yellow tile from attaching there then, reaching a contradiction. Consequently, \(t_{sbTF}(N) = F\) and \(t_{sbTT}(N) = T\). Now the glue assignment to the cyan tile types has been accomplished as follows:

\[
\begin{array}{c|c|c}
\text{F} & t_{sbFF} & \text{F} \\
\text{F} & t_{sbFT} & \text{F} \\
\text{T} & t_{sbTF} & \text{T} \\
\text{T} & t_{sbTT} & \text{T}
\end{array}
\]

Now only the 2 \text{DGNL-white} and 2 \text{DGNL-black} tile types are not yet assigned with glues. See the pattern in Fig. 10 (Right), where we find a \text{DGNL-white} position next to
the pattern Init-red(F). The tile attaching there hence has east glue f and south glue n, regardless of which white tile attaches below. We denote its type by \( t_{DGNLwF} \); then \( t_{DGNLwF}(S) = n \) and \( t_{DGNLwF}(E) = f \). As for the other type \( t_{DGNLwT} \), consider another instance of the template in Fig. 10 with blue (T) at the top of the rightmost column. Then we obtain \( t_{DGNLwT}(S) = n \) and \( t_{DGNLwT}(E) = t \). Black analogues of these instances assign the DGNL-black tile types \( t_{DGNLbF} \) and \( t_{DGNLbT} \), with glues partially as \( t_{DGNLbF}(S) = v \), \( t_{DGNLbT}(S) = v \), \( t_{DGNLbF}(E) = f \), and \( t_{DGNLbT}(E) = t \). These four tile types have been assigned with glues partially so far as follows.

On the tenth column from the right in Fig. 6, there is a DGNL-white position. The tile attaching there is of type \( t_{DGNLwF} \), and hence, it is assigned with glues as \( t_{DGNLwF}(N) = t_{DGNLwF}(W) = F \). As for the other type \( t_{DGNLwT} \), observe that the fourth and fifth columns in Fig. 7 have two DGNL-white positions in total. At both of them, tiles of this type attach. See the CE positions just above them (on the sixth row from the bottom). CE tiles attaching there must have \( t_{DGNLwT}(N) \) as their south glue. Since \( t_{CEfs} \) tiles cannot attach next to each other and the south glue of the other CE tile types is F, we get \( t_{DGNLwT}(N) = F \). For the sake of directedness, its west glue must be T in order to enable a tile of this type to attach to the east of cyan positions. The glue assignment to \( t_{DGNLwF} \) and to \( t_{DGNLwT} \) has been completed as follows.

We shift our attention to the DGNL-black tile types \( t_{DGNLbF} \) and \( t_{DGNLbT} \). On the fourth column in Fig. 6, there is a DGNL-black position, at which a \( t_{DGNLbF} \) tile attaches. Recall that the first two columns in the figure are identical to the rightmost two columns of Fig. 5. Therefore, we can apply Property 2 to determine \( t_{DGNLbF}(N) = F \). Its west glue is determined as \( t_{DGNLbF}(W) = F \) since to its west are found cyan tiles, whose west and east glues are the same, and then a Sat tile, with east glue F. As for the assignment to the other type \( t_{DGNLbT} \), see Fig. 8, in which there is a DGNL-black position whose north neighbor is yellow. The type of the DGNL-black tile is \( t_{DGNLbT} \), and \( t_{DGNLbT}(N) = T \). Its west glue cannot be F for the sake of directedness. Since the east glues of cyan tiles are either F or T and DGNL-black positions only ever appear to the east of cyan tiles, in order for \( t_{DGNLbT} \) tiles to attach, \( t_{DGNLbT}(W) = T \) must hold. The glue assignment to \( t_{DGNLbF} \) and to \( t_{DGNLbT} \) has been completed as follows.

Now that all the 21 tile types have been assigned with glues completely, we should distinguish F and T from n or v. Compare the DGNL-white tile type whose west glue is F and the two cyan tile types whose west glues are F. For the directedness, these
tile types imply \( n \neq F \) and \( n \neq T \). In the same way, comparing DGNL-black tile type with the west glue \( F \) with these cyan tile types distinguishes \( v \) from \( F \) or \( T \). The tile type set is now isomorphic to the set \( T_{val} \) (Property 1). This concludes the proof of the NP-hardness of 11-PATS.

5 Proofs of technical lemmas

In this section, we prove the lemmas unproven so far.

5.1 Proof of Lemma 2

Lemma 2 If a directed RTAS with 21 tile types uniquely self-assembles a pattern including GADGET, then it has either

1. at least 4 cyan tile types, or
2. the 3 cyan tile types shown in Fig. 11, 1 red (F) tile type, 1 blue (T) tile type, and 2 tile types whose color is either red (F) or blue (T).

Proof The color pattern of the top row of LB4 is

\[
\text{Sat} - \text{red}(F) - [\text{CE}]^2Y^3[\text{CE}]Y^2[\text{CE}]^cY[\text{CE}]^2 - \text{Sat},
\]

where \( Y \) indicates a yellow position and \( c \) is some constant. The rightmost column of LB4 is represented from bottom to top as \( \text{red}(F)^2\text{blue}(T)^{2r-1}\text{red}(F)^{2r-1}\text{blue}(T)^2\text{Sat} \), where \( r \) is some constant. Recall that at the beginning of Sect. 4.1, we claimed that the constants \( c \) and \( r \) are set large enough for the sake of this proof. In fact, we set \( c = 25 \) and \( r = 13 \) (i.e., \( 2r-1 = 25 \)). Note that their values are set independently of \( \phi \).

We will prove that if only 3 cyan tile types \( A, B, C \) are available for the RTAS, then they have to be assigned with glues as shown in Fig. 11. Below, we focus on the cyan region of LB4; hence, for instance, by “top row,” we refer to the top row of the cyan region, unless otherwise noted.

First we deny the possibility that the west glues of \( A, B, C \) are pairwise distinct or all the same. Indeed, with the pairwise-distinctness, the RTAS cannot help but assemble the top row in one of the following ways:

\[
\begin{align*}
AA \cdots & \quad \text{if } A(\text{E}) = A(\bar{\text{w}}). \\
ABAB \cdots & \quad \text{if } A(\text{E}) = B(\bar{\text{w}}) \text{ and } B(\text{E}) = A(\bar{\text{w}}). \\
ABB \cdots & \quad \text{if } A(\text{E}) = B(\bar{\text{w}}) = B(\text{E}). \\
ABCABC \cdots & \quad \text{if } A(\text{E}) = B(\bar{\text{w}}), B(\text{E}) = C(\bar{\text{w}}), \text{ and } C(\text{E}) = A(\bar{\text{w}}). \\
ABCBC \cdots & \quad \text{if } A(\text{E}) = B(\bar{\text{w}}), B(\text{E}) = C(\bar{\text{w}}), \text{ and } C(\text{E}) = B(\bar{\text{w}}). \\
ABCC \cdots & \quad \text{if } A(\text{E}) = B(\bar{\text{w}}), B(\text{E}) = C(\bar{\text{w}}) = C(\text{E}). \\
\end{align*}
\]

or their analogues obtained by changing the roles of \( A, B, C \). As a result, the cyan region exposes a periodic sequence of glues of period at most 3 (possibly except the first
one or two glues) to the north. Recall that at the point where this lemma is concerned, only 7 tile types remain uncolored, and hence, even if we color all of them by CE, we have only 8 CE tile types. Consider the task for the RTAS to assemble the top row of LB4, or specifically, its subpattern [CE]²Y. If the sequence of glues exposed by cyan region is of period 1 (all the glues are the same), then the position of Y would need to be hardcoded by tiling all the c = 25 CE positions with tiles of distinct types, but there are only at most 8 CE tiles. Even with period 3, after 24 CE positions, pumping would occur and a yellow tile would never attach, reaching a contradiction. In this way, the choice of the value for c makes it impossible for the RTAS to assemble the top row if the cyan region consisting of all but the rightmost two columns exposes a periodic sequence of glues of period at most 3 to the north, and the choice of the value for r is motivated analogously by the assemblability of the rightmost column.

Likewise, the south glues of A, B, C are not pairwise-distinct due to the same problem occurring on the east with large r. Therefore, A(0) = B(0) = C(0) must not hold; otherwise, the directedness of the RTAS would imply the contradictory pairwise-distinctness of their south glues. This also shows that two cyan tile types are not hold; otherwise, the directedness of the RTAS would imply the contradictory sequence of glues of period at most 3 to the north, and the choice of the value for r is motivated analogously by the assemblability of the rightmost column.

Having determined A(0) = B(0) ≠ C(0), let A(0) = B(0) = 0 and C(0) = 1 for some distinct glues 0, 1. For the sake of directedness, A(S) ≠ B(S) must hold; let A(S) = a and B(S) = b for some distinct glues a, b. Without loss of generality, we can assume C(S) = a. We denote the north and east glues of A, B, C as follows:

\[
\begin{align*}
0 & \quad A & e_1 & \quad n_1 \\
 & a & & \\
0 & \quad B & e_2 & \quad n_2 \\
 & b & & \\
1 & \quad C & e_3 & \quad n_3 \\
 & a & & \\
\end{align*}
\]

Recall that already at the beginning of this proof, we have denied \(n_1 = n_2 = n_3\) and \(e_1 = e_2 = e_3\).

We claim that the north glues of A, B, C must be either a or b. Firstly, if \(n_1 \notin \{a, b\}\), then any row but the topmost one cannot help but assemble with only B and C tiles. With \(n_2 = n_3\), the second topmost row exposes a sequence of all the same glues to the north, and hence, the top row would assemble periodically, reaching a contradiction. On the other hand, \(n_2 \neq n_3\) forces the RTAS to assemble the rightmost column as \(BBB \cdots, CCC \cdots, BCBC \cdots\), or \(CBCB \cdots\) up to its second topmost position, and this is enough for a contradiction. Thus, \(n_1\) is a or b. Secondly, if \(n_2 \notin \{a, b\}\), then any row except the top assembles with only A and C tiles. Since A and C have the same south glue, these rows assemble periodically either as \(A \cdots A, ACAC \cdots, ACC \cdots\), or their analogues which begin with C. With \(n_1 = n_3\), the top row would assemble periodically in one of these ways, a contradiction. On the other hand, \(n_1 \neq n_3\) forces the rightmost column to be assembled with tiles of sole type up to the third topmost positions, a contradiction. Finally, if \(n_3 \neq a, b\), then the rightmost column would assemble periodically with only A and B tiles as \(A \cdots A, ABAB \cdots, AB \cdots B\), or their analogues which begin with B, reaching a contradiction. Therefore, \(\{n_1, n_2, n_3\} = \{a, b\}\). Analogously, we can prove \(\{e_1, e_2, e_3\} = \{0, 1\}\).
Now we will prove that the only set of 3 cyan tile types with which a directed RTAS can self-assemble LB4 is the one in Fig. 11, provided no other cyan tile types are available.

Case 1 \( e_1 = e_2 = 0, e_3 = 1 \) (or \( n_1 = a, n_2 = b, n_3 = a \)). In order for the rightmost column not to expose a periodic sequence of glues to the east, at least one \( C \) tile must be placed in the column. Since a \( C \) tile has the same east and west glue and this glue is found only on this tile type, a row assembles with just \( C \) tiles. All of the rows above would be mono-type as well, reaching a contradiction. So we have proved that these tiles must not copy their west glues to the east. Analogously, they must not copy their south glues to the north, that is, it does not hold that \( n_1 = a, n_2 = b, \) and \( n_3 = a \).

Case 2 \( e_1 = e_2 = 1, e_3 = 0 \) (or \( n_1 = b, n_2 = a, n_3 = b \)). In this case, the tile types are as follows.

```
A 0 a 1
  a
B 1 b
C a 0
```

Any row admits a \( C \) tile at every other position. The bottom row assembles either as \( C[A/B]C[A/B] \) \( \cdots \) or as \( [A/B]C[A/B]C \) \( \cdots \). With \( n_3 = b \), any row but the bottom one would assemble periodically as \( CBCB \) \( \cdots \), reaching a contradiction. Thus, \( n_3 = a \), and this implies \( n_1 = b \) due to Case 1. If \( n_2 = b \), then to the north of both \( A \) and \( B \) tiles are tiles of type \( B \). This means that the second lowest row assembles as \( CBCB \) \( \cdots \), and so would all the rows above, reaching a contradiction. Thus \( n_2 = a \).

```
A 1 b
  a
B 0 a
C b 1
```

To the north of an \( A \) tile is always a tile of type \( B \). This and the fact that at every row, \( C \) tiles appear at every other position imply that if at a row, an \( A \) tile attaches, then the assembly of the row just above is the same as that of the current row with \( A \) and \( B \) swapped (for instance, above the row \( CACBCA \), the row \( CBCACB \) assembles). Thus, if both an \( A \) tile and a \( B \) tile attach at the bottom row, then the rightmost column is either the alternation of \( A \) and \( B \) tiles or consists of only \( C \) tiles, a contradiction. If the bottom row assembles as an alternation of \( B \) and \( C \) tiles, then each of the rows above is either an alternation of \( A \) and \( C \) tiles or \( B \) and \( C \) tiles, and hence, the topmost row would expose a periodic sequence of glues to the north, reaching a contradiction. Even if the bottom row assembles as an alternation of \( A \) and \( C \) tiles, this contradiction would arise.

Due to Cases 1 and 2 and the fact that the north glues of these cyan tile types must not be all the same, \( n_1 \neq n_3 \) is necessary.

Case 3 \( e_1 = 0, e_2 = e_3 = 1 \) (or \( n_1 = a, n_2 = n_3 = b \)). The tile types are as follows:
It is clear that every row assembles as $A^+ BC^+$, $A^+ B . BC^*$, $A^*$, or $C^*$, written in the syntax of regular expression, that is, the asterisk indicates zero or more occurrences of tiles of that type and $+$ indicates one or more occurrences of tiles of that type. In particular, the assembly of the top row is $A^+ BC^+$ because it must expose to the north a non-periodic sequence of glues and its breach of periodicity must not be at the very beginning or very end. Then it exposes to the south the glue sequence of the form $a^+ ba^+$. In order for the row below to be assembled so as to expose a compatible sequence of north glues, its assembly must also be as $A^+ BC^+$ and moreover, the $B$ must be vertically aligned with the $B$ at the top row. In this way, we can see that all rows must assemble as $A^+ BC^+$, but then the rightmost column would consist of only $C$ tiles and expose a periodic sequence of glues to the east, a contradiction.

Case 4 $e_1 = 1, e_2 = 0, e_3 = 1$ (or $n_1 = n_2 = b, n_3 = a$). The tile types are as follows:

This case can be denied in a very similar manner as in Case 3. In this case, any row cannot help but assemble as $B^* AC^*$. In particular, the top row must assemble as $B^+ AC^+$ so that the sequence of glues it exposes to the south is $b^+ aa^+$. If $n_2 = a$, then similar to Case 3, all the rows assemble as $B^+ AC^+$ and more strongly, all of their $A$’s are aligned vertically. Then the rightmost column would expose a periodic sequence of glues to the east, a contradiction. Hence, $n_2 = b$. Then, the top two rows assemble as:

which force $n_1 = b$ and $n_3 = a$. Thus, to the south of a $C$ tile, only a $C$ tile can attach. As a result, the rightmost column would expose a periodic sequence of glues to the east, reaching a contradiction.

The analysis of Case 4 has denied the possibility that $n_1 = n_2 = b$ and $n_3 = a$. Now the glue $n_2$ has been fixed to $a$, and the glue $e_3$ to $0$ as follows:

Case 5 $e_1 = 1, e_2 = e_3 = 0$ (or $n_1 = b, n_2 = n_3 = a$). The tile types are as follows:
The assembly of any row is represented as a substring of \((ACB^*)^*\). We claim that any row but the top or bottom assembles with the following two conditions.

(a) Two \(B\) tiles do not get next to each other, and
(b) \(ACAC\) never appears.

In other words, we claim that the assembly of any row but the top or bottom is a substring of \((ACB)^*\). Recall the necessity of \(n_1 \neq n_3\), that is, one of them is \(a\) and the other is \(b\). The first condition is certified by observing that no row can expose two consecutive \(b\) glues to the north. As for the second, suppose that we found \(ACAC\) on a row. If \(n_1 = b\), then \(n_3 = a\), and to the north of \(A\) tiles, \(B\) tiles must attach as:

\[
\begin{array}{c}
\text{B @ B} \\
\text{A C A C}
\end{array}
\]

However, then no tile could attach at the position @. The other case of \(n_3 = b\) leads to the same contradiction. The second condition has been thus certified.

Since \(B\) and \(C\) tiles have the same east glue, on the second rightmost column, an \(A\) tile must occur. We focus on one of such \(A\) tiles, which is marked as \(\boxed{A}\) from now on. Due to the above condition, around \(\boxed{A}\), the assembly is \(BAC\boxed{A}\).

Now observe how tiles attach around this assembly; we consider only the subcase when \(n_3 = b\) (i.e., \(n_1 = a\)); the other subcase \(n_1 = b\) and \(n_3 = a\) is essentially symmetric. In this subcase, the type of a tile above a \(C\) tile is \(B\).

The row above, if any, assembles as

\[
\begin{array}{c}
\text{B A C B} \\
\text{C B A C}
\end{array}
\]

The assembly of rows above proceed in this way as follows:

\[
\begin{array}{c}
\text{C B A C} \\
\text{A C B A} \\
\text{B A C B} \\
\text{C B A C}
\end{array}
\]

The rows below assemble in the same way as:

\[
\begin{array}{c}
\text{C B A C} \\
\text{A C B A} \\
\text{B A C B} \\
\text{C B A C}
\end{array}
\]
As a result, the rightmost column would assemble periodically, reaching a contradiction.

Now only the tile type set depicted in Fig. 11 has remained valid. We reproduce it for the sake of arguments as follows:

\[
\begin{array}{c}
0 & a & 0 \\
\hline
A & a & a \\
0 & b & 1 \\
B & a & b \\
1 & c & a \\
C & b & 1 \\
\hline
\end{array}
\]

Observe that the south neighbor of a \(B\) tile is always of type \(C\). Thus, assembled with tiles of these types, \(LB4\) does not expose two consecutive 1 glues eastward. This property plays an important role in proving the need of 4 tile types of color red(F) or blue(T) in order to assemble \(LB4\) with cyan tiles of these 3 types.

We actually prove that with at most 3 red(F)/blue(T) tile types, the rightmost column of \(LB4\), consisting of \(F\)s and \(T\)s, cannot be assembled. Suppose there were at most 3 red(F)/blue(T) tile types. Then either there is a sole red(F) tile type with at most 2 blue(T) tile types, or there is a sole blue(T) tile type with at most 2 red(F) tile types.

Here we only show that the rightmost column cannot assemble in the first case, as the argument for 1 blue(T) tile type can follow the same steps at analogous indexes. Let \(t_F\) be the red(F) tile type and \(t_{T1}, t_{T2}\) be the blue(T) tile types. At all red(F) positions, \(t_F\) tiles are to attach. Hence, \(t_F(N) = t_F(S)\). Observe the \(2d - 1\) consecutive red(F) positions on this column. Due to the above-mentioned property of east glues of cyan tiles, \(t_F\) tiles forming this portion receive glue 0 from the west. Thus, \(t_F(W) = 0\), and \(t_F(S)\) must differ from \(a\) or \(b\); let \(t_F(N) = t_F(S) = c\).

\[
\begin{array}{c}
0 & F & T1 & T2 \\
\hline
\end{array}
\]

Observe the lowest blue(T) position. Without loss of generality, the tile type attaching there is \(t_{T1}\). Then \(t_{T1}(S) = c\), and hence, \(t_{T1}(W)\) is not 0 due to the directedness; since cyan tiles can expose only 0 or 1 to their east, \(t_{T1}(W) = 1\). Since a tile attaching at its north neighbor cannot receive a glue 1 from the west, its type cannot be \(t_{T1}\), and so it is \(t_{T2}\). Hence, \(t_{T2}(W) = 0\), and this requires \(t_{T2}(S)\) to be distinct from \(a, b, c\). Let \(t_{T2}(S) = d\), and then \(t_{T1}(N) = t_{T2}(S) = d\). These tile types have been assigned with glues partially so far as follows.

\[
\begin{array}{c}
0 & F & 1 & T1 & 0 & T2 \\
\hline
\end{array}
\]

The column has assembled from the bottom as \(t_{T2}t_Ft_{T1}t_{T2}\). Due to the lack of a third blue(T) tile type, the north of \(t_{T2}\) must be either \(c\) or \(d\). If it were \(d\), then the column assembles as \(2^d t_{T1}t_{T2}^{2d-2}\), but then it still exposes glue \(d\) to its north and only a \(t_{T2}\) tile would attach, reaching a contradiction. Otherwise, the column assembles as
\( t_2^2 (t_{T1} t_{T2})^{d-1} t_{T1} \) and even in this case, its north neighbor would be colored blue(T) by choice of an odd number of consecutive blue(T) positions, reaching a contradiction. 

\[\square\]

5.2 Proof of Lemma 3

**Lemma 3** If a directed RTAS with at most 21 tile types uniquely self-assembles a pattern including GADGET, then it contains at least 2 CE tile types and the total number of CE and yellow tile types is at least 4. Moreover, if it contains exactly 2 CE tile types \( t_1, t_2 \) and exactly 2 yellow tile types \( t_3, t_4 \), then these four tile types are labelled as depicted at the right bottom of Fig. 12.

**Proof** Here, we prove Lemma 3. Since it refers to Fig. 12, we reproduce it here as Fig. 14.

We have already shown that any directed RTAS with at most 21 tile types needs at least two CE tile types in order to self-assemble GADGET. If it has exactly two of them, say \( t_1, t_2 \), then as done in Lemma 1, we can prove that \( t_1(W) \neq t_2(W) \) and \( t_1(E) \neq t_2(E) \), while \( t_1(S) = t_2(S) \). Let \( t_1(W) = a \) and \( t_2(W) = b \) for some distinct labels \( a, b \), and let \( t_1(S) = t_2(S) = 0 \).

With three CE tile types, the first statement of this lemma is trivial. Hence, it suffices to prove that if the RTAS has exactly two CE tile types \( t_1, t_2 \), then it must have at least 2 yellow tile types. For the sake of contradiction, suppose that there were only one yellow tile type. See Fig. 15, where a subpattern of GADGET is depicted, with CE tiles being colored just white for clarity. Without loss of generality, the type of CE tile at \( (3, 4) \) is \( t_1 \). Being self-stacked, the sole yellow tile type has the same north and south glues, and moreover, these glues are the same as the south glues of \( t_1 \) and \( t_2 \). Thus, the west glue of the yellow tile type must differ from \( a \) or \( b \); let it be \( c \) (see Fig. 15 (right)). Then \( t_1(E) = c \), and hence, a \( t_1 \) tile cannot be adjacent to another \( t_1 \) tile horizontally, so the tile type at \( (2, 4) \) is \( t_2 \). However, then neither \( t_1 \) nor \( t_2 \) tiles can be at \( (1, 4) \) due to the east glue mismatch. Therefore, if the RTAS has only 2 CE tile types, it must have at least 2 yellow tile types \( t_3 \) and \( t_4 \).

![Fig. 14](This is a reproduction of Fig. 12 (Color figure online))
We prove the second statement of the lemma. Without loss of generality, the type of the yellow tile at (4, 4) is $t_3$. As proved above, $t_3(S)$ is not 0; let $t_3(S) = 1$. Independent of the tile type at (4, 5), $t_3(N) = 0$. Thus, the tile type at (4, 3) is not $t_3$ but $t_4$, and hence, let $t_4(N) = t_3(S) = 1$. As shown in Fig. 14 (left), then the tiles at (1, 2) and (3, 3) are of type $t_3$ and their south neighbors are of type $t_4$. Thus, $t_3(E) = t_4(W)$, and this glue is either $a$ or $b$ (see the positions (1, 2) and (2, 2)). This means $t_4(S) \neq 0$ or more strongly $t_4(S) = 1$ since otherwise no yellow tile could attach to the south of a $t_4$ tile.

As illustrated in Fig. 14 (right), any yellow column is to self-assemble in such a way that all but its topmost position is filled with $t_4$ tiles. Since $t_3(S) = t_4(S) = 1$, their west glues must disagree, and thus, the white west neighbor of a $t_3$ tile is always of type $t_1$ whereas that of a $t_4$ tile is always of type $t_2$. Now the resulting assembly of the pattern looks partially as depicted in Fig. 14 (right). In particular, $t_1$ tiles attach at both (1, 3) and (2, 3) and a $t_3$ tile attaches at (3, 3), and hence, $t_3(W) = t_1(E) = t_1(W) = a$. The assembly $t_4 t_2 t_4$ of the bottom row implies $t_4(W) = t_4(E) = t_2(W) = t_2(E) = b$. Finally, $t_3(E) = t_4(W) = b$. The glue assignment has been completed as shown in Fig. 14 (right).

6 Further research

In this paper, we have proposed a manually-checkable proof for the NP-hardness of 11-pats. It is natural to try to improve this result further to a manually-checkable proof for that of 2-pats. Such a proof would complement the computer-assisted certificate for the NP-hardness of 2-pats by Kari et al. (2015a).

Also of interest is to prove an inapproximability ratio for $k$-pats larger than $22/21 \approx 1.048$ for as small $k$ as possible.

Acknowledgements We are very thankful to the anonymous referees for their valuable comments on the earlier versions of this paper. This work is supported in part by NSF Grants CCF-1049899 and CCF-1217770 to M-Y. Kao and by HIIT Pump Priming Grant 902184/T30606, Academy of Finland, Postdoctoral Researcher Grant 13266670/T30606, JST Program to Disseminate Tenure Tracking System, MEXT, JAPAN, No. 6F36, and JSPS Grant-in-Aid for Research Activity Start-up No. 15H06212 to S. Seki.

Springer
References

Barish R, Rothemund PWK, Winfree E (2005) Two computational primitives for algorithmic self-assembly: copying and counting. Nano Lett 5(12):2586–2592
Brun Y (2008) Solving NP-complete problems in the tile assembly model. Theor Comput Sci 395:31–46
Brun Y (2008) Solving satisfiability in the tile assembly model with a constant-size tileset. J Algorithms 63(4):151–166
Brun Y (2012) Efficient 3-SAT algorithms in the tile assembly model. Nat Comput 11:209–229
Cook M, Rothemund PWK, Winfree E (2004) Self-assembled circuit patterns. In: Proceedings of the 9th International Workshop on DNA Based Computers (DNA 9), LNCS, vol. 2943. Springer, p 91–107
Culik K, Kari J (1997) On aperiodic sets of Wang tiles. Foundations of computer science potential— theory—cognition, LNCS, vol 1337. Springer, Berlin, pp 153–162
Czeizler E, Popa A (2013) Synthesizing minimal tile sets for complex patterns in the framework of patterned DNA self-assembly. Theor Comput Sci 499:23–37
Evans CG (2014) Crystals that count! physical principles and experimental investigations of DNA tile self-assembly. Ph.D. thesis, California Institute of Technology
Göös M, Lempläinen T, Czeizler E, Orponen P (2014) Search methods for tile sets in patterned DNA self-assembly. J Comput Syst Sci 80:297–319
Johnsen A, Kao MY, Seki S (2013) Computing minimum tile sets to self-assemble patterns in 29-colors. In: Proceedings of the 24th International Symposium on Algorithms and Computation (ISAAC 2013), LNCS, vol. 8283. Springer, p 699–710
Kari L, Kopecki S, Étienne Meunier P, Patitz MJ, Seki S (2015a) Binary pattern tile set synthesis is NP-hard. In: Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP 2015), LNCS, vol. 9134. Springer, p 1022–1034
Kari L, Kopecki S, Seki S (2015b) 3-color bounded patterned self-assembly. Nat Comput 14(2):279–292
Ma X, Lombardi F (2008) Synthesis of tile sets for DNA self-assembly. IEEE Trans Comput-Aided Des Integr Circuits Syst 27(5):963–967
Ma X, Lombardi F (2009) On the computational complexity of tile set synthesis for DNA self-assembly. IEEE Trans Circuits Syst II 56(1):31–35
Rothemund PWK, Papadakis N, Winfree E (2004) Algorithmic self-assembly of DNA Sierpinski triangles. PLoS Biol 2(12):e424
Schaefer TJ (1978) The complexity of satisfiability problems. In: Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC 1978), p 216–226
Seki S (2013) Combinatorial optimization in pattern assembly (extended abstract). In: Proceedings of the 12th International Conference on Unconventional Computation and Natural Computation (UCNC 2013), LNCS, vol. 7956. Springer, p 220–231
Stefanovic D, Turberfield A (eds) (2012) The 18th International Conference on DNA Computing and Molecular Programming, Aarhus, Denmark, 14–17 August 2012
Winfree E (1998) Algorithmic self-assembly of DNA. Ph.D. thesis, California Institute of Technology
Winfree E (2000) Algorithmic self-assembly of DNA: theoretical motivations and 2D assembly experiments. J Biomol Struct Dyn Special Issue S2:263–270
Winfree E, Liu F, Wenzler LA, Seeman NC (1998) Design and self-assembly of two-dimensional DNA crystals. Nature 394:539–544
Zhang J, Liu Y, Ke Y, Yan H (2006) Periodic square-like gold nanoparticle arrays templated by self-assembled 2D DNA nanogrids on a surface. Nano Lett 6(2):248–251