The complexity of fixed-height patterned tile self-assembly

Shinnosuke Seki[1] and Andrew Winslow[2]

[1] Algorithmic "*Oritatami*" Self-Assembly Lab.,
University of Electro-Communications, Tokyo, Japan
[2] Université Libre de Bruxelles, Brussels, Belgium

CIAA 2016, July 19-22

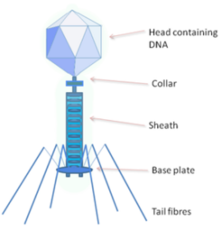
Self-assembly is everywhere!

snow crystal

pattern

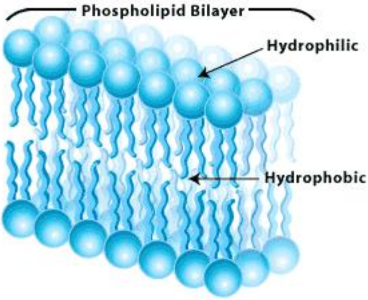
galaxy

city



virus capsid

An example of self-assembly
Lipid bilayer



Water (external environment) affects components (lipids), but does not intend to lead them to the membrane structure.

DNA self-assembly

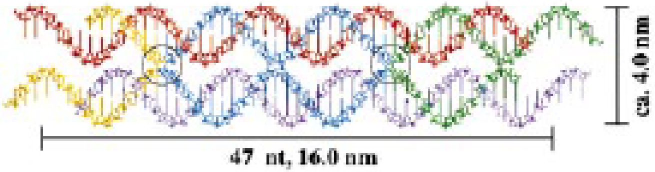
Engineering Goal

Driven by

- Watson-Crick complementarity A-T, C-G.
- Thermodynamics
- Kinetics
- ...

DNA self-assembly
DNA tile implementation

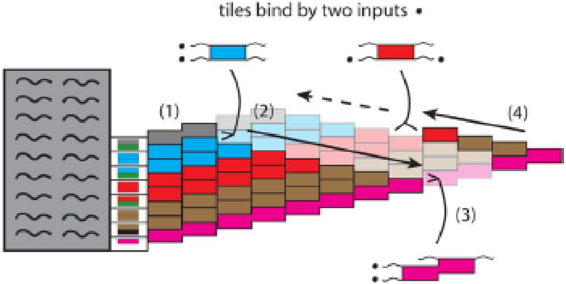
Interactive DNA tiles are implemented *in vitro* as a DNA double-crossover molecule [Winfree et al., *Nature*, 1998]



4 single strands (red, yellow, purple, green), called **sticky ends**, enable the "tile" to interact with other "tiles".

DNA self-assembly

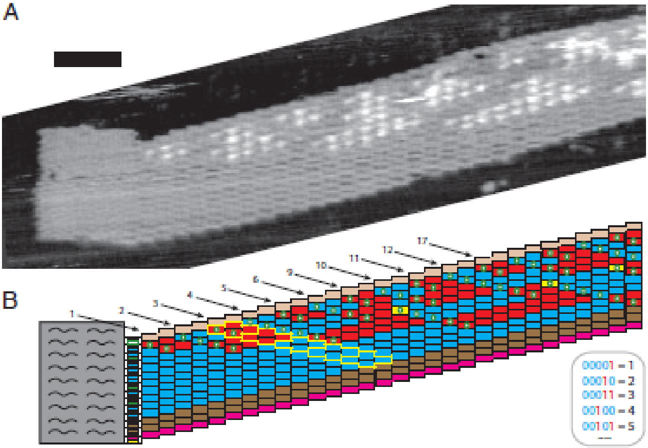
Binary counter [Barish et al., *PNAS*, 2009]



The gray box to the left is the seed (scaffold for assembly process) made of DNA origami [Rothemund, *Nature*, 2006].

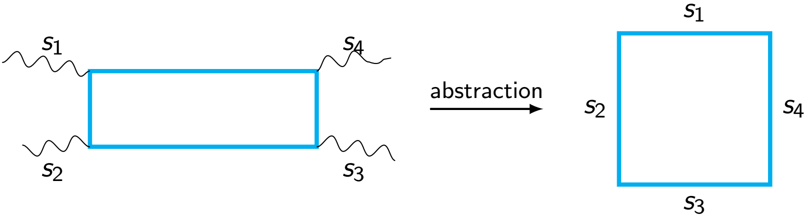
DNA self-assembly

Binary counter [Barish et al., *PNAS*, 2009]



Abstract Tile-Assembly Model (aTAM) [Winfree 1998]

Abstraction of DNA tile

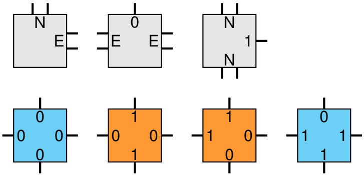


A *square tile type t* is an element of $\Gamma \times \Gamma \times \Gamma \times \Gamma \times \mathbb{N}$, where

- Γ is a set of glues (DNA sequences),
- The last integer specifies its color, representing some chemical property.

Rectilinear tile assembly system (RTAS)

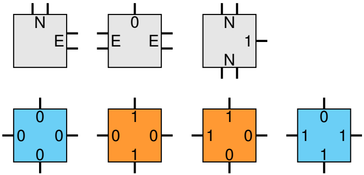
The rectilinear TAS (RTAS) is a variant of Winfree's aTAM system suitable for assembling rectangular patterns.

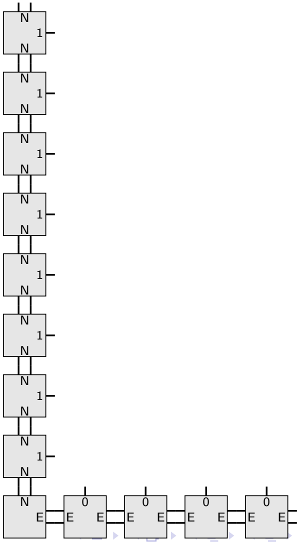


Rectilinear tile assembly system (RTAS)

The rectilinear TAS (RTAS) is a variant of Winfree's aTAM system suitable for assembling rectangular patterns.

- Initial assembly (seed) is of L-shape;

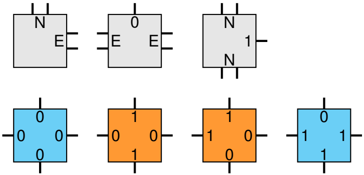


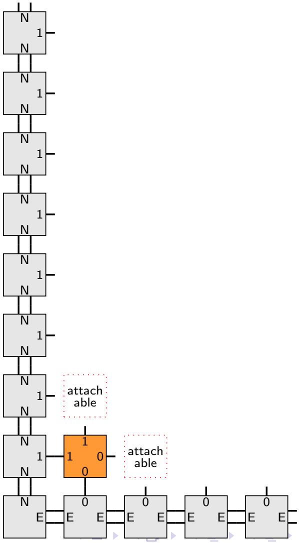


Rectilinear tile assembly system (RTAS)

The rectilinear TAS (RTAS) is a variant of Winfree's aTAM system suitable for assembling rectangular patterns.

- Initial assembly (seed) is of L-shape;
- A tile attaches if both its west and south glues match.

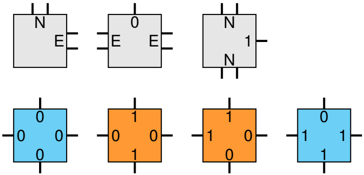


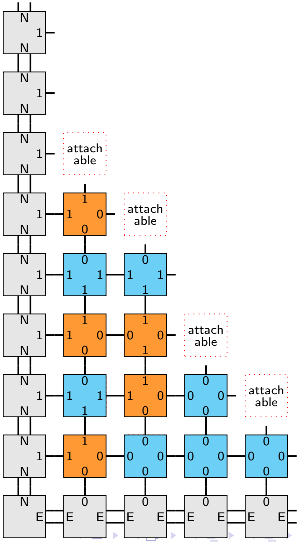


Rectilinear tile assembly system (RTAS)

The rectilinear TAS (RTAS) is a variant of Winfree's aTAM system suitable for assembling rectangular patterns.

- Initial assembly (seed) is of L-shape;
- A tile attaches if both its west and south glues match.

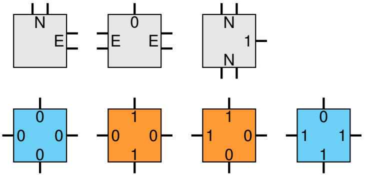


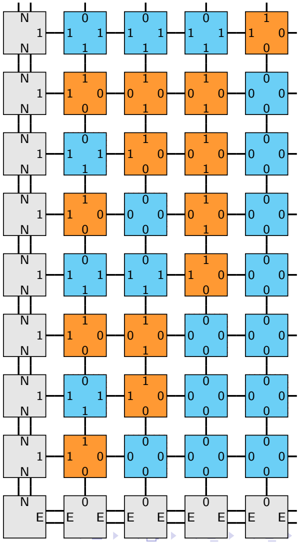


Rectilinear tile assembly system (RTAS)

The rectilinear TAS (RTAS) is a variant of Winfree's aTAM system suitable for assembling rectangular patterns.

- Initial assembly (seed) is of L-shape;
- A tile attaches if both its west and south glues match.





Rectilinear tile assembly system (RTAS)

Unique assembly by RTAS

An RTAS is a pair $\mathcal{T} = (T, \sigma_L)$, where

T a finite set of tile types

σ_L an L-shape seed

An RTAS uniquely self-assembles a pattern P if the pattern of any of its terminal assembly is P.

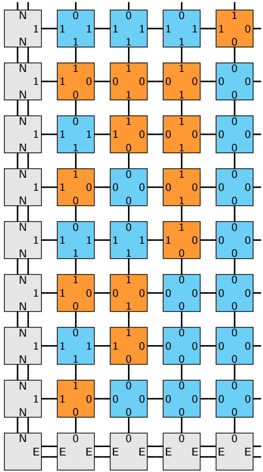
Rectilinear tile assembly system (RTAS)

Uniformity

An RTAS is uniform if all the glues on the x-axis of the seed are identical and so are those on the y-axis.

Example

The RTAS to assemble the binary counter was uniform (see right).



Constant colored PATS
Definition

"Any given logic circuit can be formulated as a colored rectangular pattern with tiles, using only a constant number of colors [Czeizler & Popa, DNA 2012]".

c-colored PATS (c-PATS)

GIVEN: a c-colored pattern P
FIND: a minimum RTAS (i.e., as few tile types as possible) that uniquely self-assembles P.

Constant colored PATS

Hardness

Theorem [Kari, Kopecki, Meunier, Patitz, S. ICALP 2015]

2-PATS is **NP**-hard.

- Computer-assisted proof.
- At the scale of 1-CPU YEAR, called "*La plus longue demonstration mathematique de l'Histoire*."
- The proof breaks down once height (or width) of input patterns is fixed to some constant.

Fixed-height PATS and uniform PATS

These are two new variants of PATS to be considered.

Height-h PATS

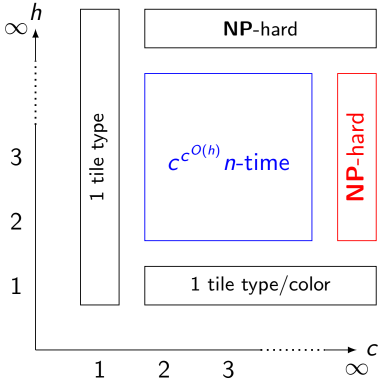
GIVEN: a pattern P of height h
FIND: a minimum RTAS that uniquely self-assembles P.

Uniform PATS

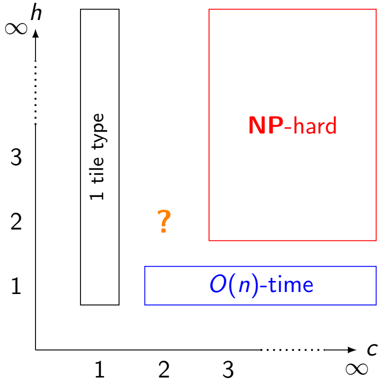
GIVEN: a pattern P
FIND: a minimum uniform RTAS that uniquely self-assembles P.

Complexity of height-h, c-PATS

Below, n is the width of an input pattern.



Non-uniform PATS



Uniform PATS

Min-state finite state transducer

Definition

A FST is a tuple (Σ, Q, s_0, δ), where

Σ, Q, s_0 : an alphabet, set of states, and initial state in Q.

$\delta \in Q \times \Sigma \to Q \times \Sigma$: a transition function. An input-output 4-tuple $\delta(p, a) = (q, b)$ is called a (a, b)*-transition* or *a-transition*

Encoding by FST

GIVEN: $S, S' \in \Sigma^*$ and $K \geq 1$

DECIDE: if $\exists$ a FST with at most K states that transduces S to S'.

Min-state finite state transducer

Hardness

The **NP**-hardness of ENCODING BY FST problem is summarized below.

	$\vert\Sigma\vert$	Proof
[Angluin, *Inform. Control*, 1978]	2	Complicated
[Vazirani & Vazirani, *TCS*, 1983]	3	Simple
This paper	2	Simple proof based on [Vazirani & Vazirani, *TCS*, 1983] for a more restricted problem

Min-state finite state transducer

Restricted variant

Encoding by FST

GIVEN: $S, S' \in \Sigma^*$ and $K \geq 1$

DECIDE: if $\exists$ a FST with at most K states that transduces S to S'

Min-state finite state transducer

Restricted variant

PROMISE ENCODING BY FST

GIVEN: $S, S' \in \Sigma^*$ and $K \geq 1$

DECIDE: if $\exists$ a FST with at most K states that transduces S to S' and satisfies the following promises:

- Each state has at most one incoming 0-transition and at most one incoming 1-transition.
- When transducing S to S':
  - $K-1$ (0, 0)-transitions, K (1, 1)-transitions, and 1 $(0, 1)$-transition is used.
  - The transitions are traversed in a unique specified order given as a part of the input.

Promise Encoding by FST
Proof

We will propose a reduction from 3-Partition to Promise Encoding by FST.

3-Partition

Given:	a multiset $A = \{a_1, a_2, \ldots, a_{3n}\}$ of integers with $\sum_{a_i \in A} a_i/n = p$ and $p/4 < a_i < p/2$,
Decide:	if $\exists$ a partition of A into n sets, each with sum p

Theorem [Garey & Johnson 1975]

3-Partition is strongly **NP**-hard.

Promise Encoding by FST

Reduction sketch

Example ($n = 2, p = 3, A = \{0, 0, 1, 1, 2, 2\}$)
Set $K = (3p + 1)n + 1 = 21$.

$$S \quad =$$

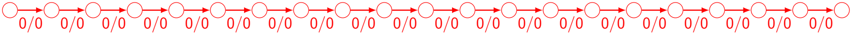
$$S' \quad =$$

Promise Encoding by FST
Reduction sketch

Example $(n = 2, p = 3, A = \{0, 0, 1, 1, 2, 2\})$
Set $K = (3p + 1)n + 1 = 21$.

$$S = 0^{K-1}0$$

$$S' = 0^{K-1}1$$

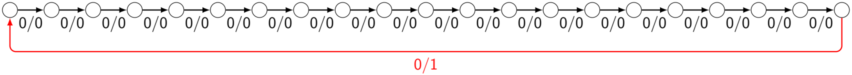


PROMISE ENCODING BY FST
Reduction sketch

Example ($n = 2, p = 3, A = \{0, 0, 1, 1, 2, 2\}$)
Set $K = (3p + 1)n + 1 = 21$.

$$S = 0^{K-1}00^{K-1}$$

$$S' = 0^{K-1}10^{K-1}$$



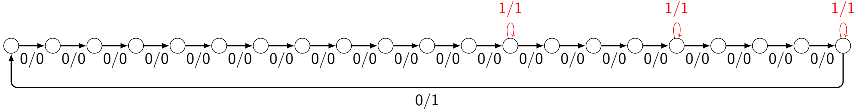
Promise Encoding by FST

Reduction sketch

Example $(n = 2, p = 3, A = \{0, 0, 1, 1, 2, 2\})$
Set $K = (3p + 1)n + 1 = 21$.

$$S \;=\; 0^{K-1}00^{K-1}\prod_{i=0}^{n}(0^{2pn+(p+1)i}10^{K-1-(2pn+(p+1)i)}0)$$

$$S' \;=\; 0^{K-1}10^{K-1}\prod_{i=0}^{n}(0^{2pn+(p+1)i}10^{K-1-(2pn+(p+1)i)}1)$$



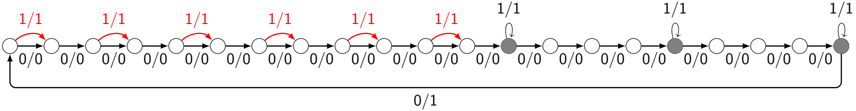
Promise Encoding by FST

Reduction sketch

Example ($n = 2, p = 3, A = \{0, 0, 1, 1, 2, 2\}$)
Set $K = (3p + 1)n + 1 = 21$.

$$S = 0^{K-1}00^{K-1}\prod_{i=0}^{n}(0^{2pn+(p+1)i}10^{K-1-(2pn+(p+1)i)}0)\prod_{j=0}^{np-1}(0^{2j}10^{K-1-(2j+1)}0)$$

$$S' = 0^{K-1}10^{K-1}\prod_{i=0}^{n}(0^{2pn+(p+1)i}10^{K-1-(2pn+(p+1)i)}1)\prod_{j=0}^{np-1}(0^{2j}10^{K-1-(2j+1)}1)$$



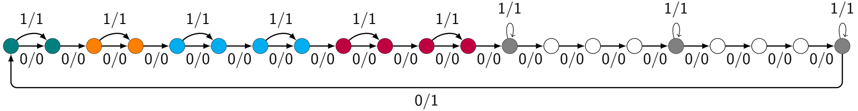
Promise Encoding by FST

Reduction sketch

Example $(n = 2, p = 3, A = \{0, 0, 1, 1, 2, 2\})$
Set $K = (3p + 1)n + 1 = 21$.

$$S = 0^{K-1}00^{K-1}\prod_{i=0}^{n}(0^{2pn+(p+1)i}10^{K-1-(2pn+(p+1)i)}0)\prod_{j=0}^{np-1}(0^{2j}10^{K-1-(2j+1)}0)$$

$$S' = 0^{K-1}10^{K-1}\prod_{i=0}^{n}(0^{2pn+(p+1)i}10^{K-1-(2pn+(p+1)i)}1)\prod_{j=0}^{np-1}(0^{2j}10^{K-1-(2j+1)}1)$$



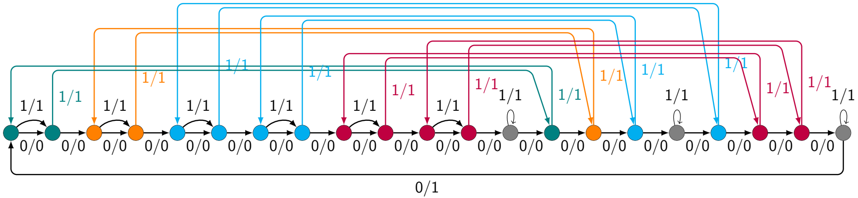
Promise Encoding by FST

Reduction sketch

Example $(n = 2, p = 3, A = \{0, 0, 1, 1, 2, 2\})$
Set $K = (3p + 1)n + 1 = 21$.

$$S = 0^{K-1}00^{K-1}\prod_{i=0}^{n}(0^{2pn+(p+1)i}10^{K-1-(2pn+(p+1)i)}0)\prod_{j=0}^{np-1}(0^{2j}10^{K-1-(2j+1)}0) \prod_{j=0}^{np-1}(0^{2j+1}110^{K-1-2j}0)$$

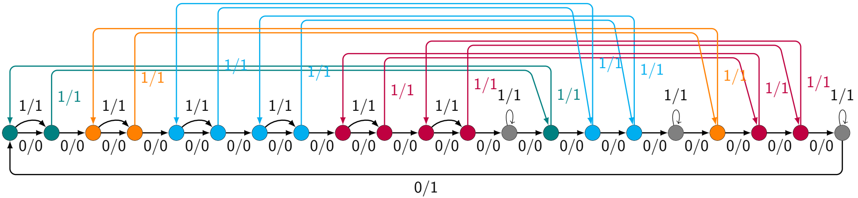
$$S' = 0^{K-1}10^{K-1}\prod_{i=0}^{n}(0^{2pn+(p+1)i}10^{K-1-(2pn+(p+1)i)}1)\prod_{j=0}^{np-1}(0^{2j}10^{K-1-(2j+1)}1) \prod_{j=0}^{np-1}(0^{2j+1}110^{K-1-2j}1)$$



Promise Encoding by FST
Reduction sketch

Example $(n = 2, p = 3, A = \{0, 0, 1, 1, 2, 2\})$
Set $K = (3p + 1)n + 1 = 21$.

$$
\begin{aligned}
S &= 0^{K-1}00^{K-1}\prod_{i=0}^{n}(0^{2pn+(p+1)i}10^{K-1-(2pn+(p+1)i)}0)\prod_{j=0}^{np-1}(0^{2j}10^{K-1-(2j+1)}0) \\
&\quad \prod_{j=0}^{np-1}(0^{2j+1}110^{K-1-2j}0) \\
S' &= 0^{K-1}10^{K-1}\prod_{i=0}^{n}(0^{2pn+(p+1)i}10^{K-1-(2pn+(p+1)i)}1)\prod_{j=0}^{np-1}(0^{2j}10^{K-1-(2j+1)}1) \\
&\quad \prod_{j=0}^{np-1}(0^{2j+1}110^{K-1-2j}1)
\end{aligned}
$$

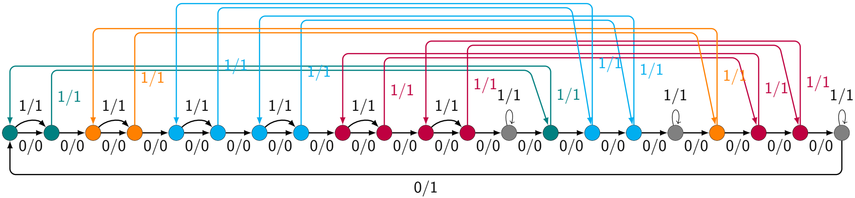


Promise Encoding by FST

Reduction sketch

Example ($n = 2, p = 3, A = \{0, 0, 1, 1, 2, 2\}$)
Set $K = (3p + 1)n + 1 = 21$.

$$
\begin{aligned}
S &= 0^{K-1}00^{K-1}\prod_{i=0}^{n}(0^{2pn+(p+1)i}10^{K-1-(2pn+(p+1)i)}0)\prod_{j=0}^{np-1}(0^{2j}10^{K-1-(2j+1)}0) \\
&\quad \prod_{j=0}^{np-1}(0^{2j+1}110^{K-1-2j}0)0^4 11010^{14}00^8 11010^{10}0 \\
S' &= 0^{K-1}10^{K-1}\prod_{i=0}^{n}(0^{2pn+(p+1)i}10^{K-1-(2pn+(p+1)i)}1)\prod_{j=0}^{np-1}(0^{2j}10^{K-1-(2j+1)}1) \\
&\quad \prod_{j=0}^{np-1}(0^{2j+1}110^{K-1-2j}1)0^4 11010^{14}10^8 11010^{10}1
\end{aligned}
$$



Promise Encoding by FST

Application

Theorem

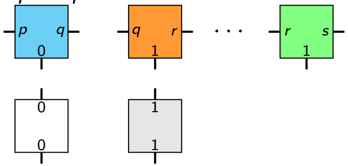
The non-uniform height-2 PATS is **NP**-hard.

Proof. Let $F = (S, S', K, S_\delta)$ be an instance of PROMISE ENCODING BY FST, where S_δ is a $2K$-ary sequence of length n to specify the order in which the available $2K$ transitions should be used. We convert F into the height-2, $(2K+2)$-colored pattern $P =$

$S_\delta[1]$	$S_\delta[2]$	$\cdots$	$S_\delta[n]$
$S[1]$	$S[2]$	$\cdots$	$S[n]$

.

for $\delta : p \to q$ on 0



Using $2K+2$ tile types, one can assemble P uniquely from a non-uniform seed $\Leftrightarrow$ F has a solution (see left). $\square$

FPT algorithm for non-uniform, height-h, c-PATS

Let P be a given c-colored pattern of height h and width n.

- Being of height h, P cannot involve more than c^h types of column.
- One type of height-h column can be uniquely self-assembled using h pairwise-distinct tile types (hard-coding).
- Column types can be encoded along the x-axis of a non-uniform seed.
- Identical columns are assembled in an identical way, while assemblies of columns of distinct type involve no tile type in common.

Upperbound (valid only for non-uniform seed)

$T = hc^h$ tile types are enough to uniquely self-assemble P from a non-uniform seed.

FPT algorithm for non-uniform, height-h, c-PATS

Recall $T = hc^h$.

- $4 \log T + \log c$ bits are enough to specify one tile type.
- Hence, $4T \log cT$ bits are enough to specify one set of at most T tile types.
- Thus, there are at most $2^{4T \log cT} = (cT)^{4T}$ sets of at most T tile types.

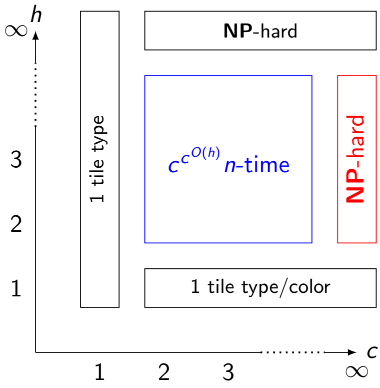
Dynamic programming

We can check in $O(hT^{h+2})n$ time if each such set of tile types can be employed to uniquely self-assemble P.

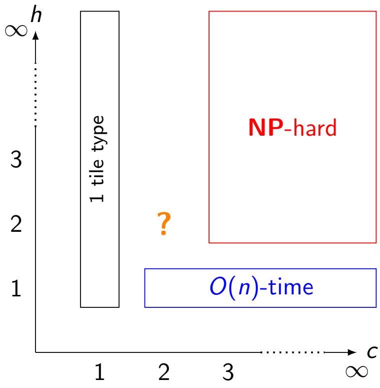
Consequently, $O((cT)^{4T} \times hT^{h+2}n) = T^{O(T)}n = c^{c^{O(h)}}n$ time is enough.

Complexity of height-h, c-PATS

Below, n is the width of an input pattern.



Non-uniform PATS



Uniform PATS

Uniform Pats

Theorem

Uniform, height-2, 3-Pats is **NP**-hard.

Proof.

A variant of Promise Encoding by FST

$\rightarrow$ Uniform, height-2, 3-Pats.

Uniform PATS

Theorem

Uniform, height-2, 3-PATS is **NP**-hard.

Proof.

A variant of PROMISE ENCODING BY FST

→ Uniform, height-2, 3-PATS.

Remaining time $<$ |Proof|

Uniform PATS

Theorem

Uniform, height-2, 3-PATS is **NP**-hard.

Proof.
A variant of PROMISE ENCODING BY FST
$\rightarrow$ Uniform, height-2, 3-PATS.

$$\begin{aligned} \text{Remaining time} &< |\text{Proof}| \\ &\leq \text{Springer's patience } (\approx 12 \text{ pages}) \end{aligned}$$

Uniform PATS

Theorem

Uniform, height-2, 3-PATS is **NP**-hard.

Proof.

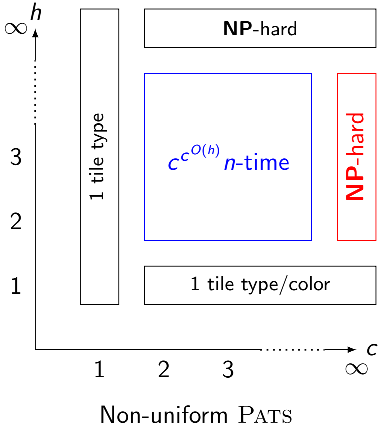
A variant of PROMISE ENCODING BY FST

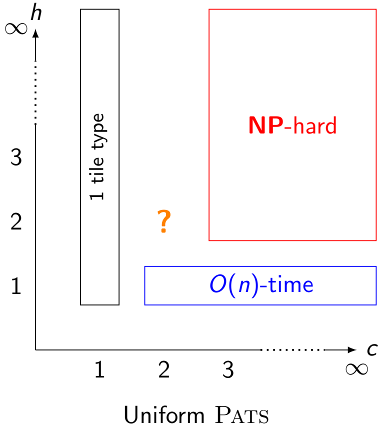
$\rightarrow$ Uniform, height-2, 3-PATS.

$$\begin{aligned} \text{Remaining time} &< |\text{Proof}| \\ &\leq \text{Springer's patience } (\approx 12 \text{ pages}) \\ &\ll 1 \text{ CPU year.} \end{aligned}$$

Complexity of height-h, c-PATS

Below, n is the width of an input pattern.





Thanks!!

This work is in part supported by

- JST Program to Disseminate Tenure-Tracking System, 6F36
- JSPS Grant-in-Aid for Young Scientists (A), 16H05854
- JSPS Grant-in-Aid for Research Activity Start-Up, 15H06212

Japan Science and Technology Agency

科研費 KAKENHI

References I

D. Angluin.
On the complexity of minimum inference of regular sets.
Inform. Control 39: 337-350, 1978

R. D. Barish, R. Schulman, P. W. K. Rothemund, and E. Winfree.
An information-bearing seed for nucleating algorithmic self-assembly.
PNAS 106(15): 6054-6059, 2009.

E. Czeizler and A. Popa.
Synthesizing minimal tile sets for complex patterns in the framework of patterned DNA self-assembly.
DNA 18, LNCS 7433, pp. 58-72, Springer, 2012.

References II

M. R. Garey and D. S. Johnson.
Complexity results for multiprocessor scheduling under resource constraints.
SIAM J. Comput. 4(4): 397-411, 1975.

L. Kari, S. Kopecki, P-E. Meunier, M. J. Patitz, and S. Seki.
Binary pattern tile set synthesis is NP-hard.
ICALP 2015, LNCS 9134, pp. 1022-1034, Springer, 2015.

P. W. K. Rothemund.
Folding DNA to create nanoscale shapes and patterns.
Nature 440: 297-302, 2006.

References III

U. V. Vazirani and V. V. Vazirani.
A natural encoding scheme proved probabilistic polynomial complete.
Theor. Comput. Sci. 24(3): 291-300, 1983.

E. Winfree.
Algorithmic Self-Assembly of DNA.
PhD thesis, California Institute of Technology, June 1998.

E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman.
Design and self-assembly of two-dimensional DNA crystals.
Nature 394: 539-544, 1998.