

# Computational modeling of the kinetic Tile Assembly Model using a rule-based approach

Abdulumelik Mohammed<sup>1</sup>, Elena Czeizler<sup>2</sup>, Eugen Czeizler<sup>3</sup>

<sup>1</sup>Dept. of Computer Science, Aalto University, Espoo, Finland

<sup>2</sup>Citrus Solutions Oy, Helsinki, Finland

<sup>3</sup>The National Institute for Research and Development in Biological Sciences,  
Bucharest, Romania

<sup>3</sup>Dept. of Computer Science, Åbo Akademi University, Turku, Finland

---

## Abstract

The (abstract) Tile Assembly Model (aTAM), is a mathematical paradigm for the study and algorithmic design of DNA self-assembly systems. It employs the use of so-called DNA-tiles, which are abstractions of experimentally achievable DNA nanostructure complexes with similar inter-matching behaviours. To this day, there are about half-dozen different experimental implementations of DNA tiles and their sub-sequent algorithmic assembly into larger complexes, see e.g. Reif et al. 2012. In order to provide further insight into the assembly process, the aTAM model has been extended to a kinetic counterpart (kTAM). Although there is a wide abundance of different variants of the abstract model, e.g., stage, step, hierarchical, temperature-k, signal-passing, etc. (see e.g. Patitz 2012), numerical simulations of the kinetic counterpart have been performed only for a few types of these systems. This might be due to the fact that the numerical models and simulations of kTAM were almost exclusively implemented using classical stochastic simulation algorithms frameworks, which are not designed for capturing models with theoretically un-bounded number of species. In this paper we introduce an agent- and rule-based modeling approach for kTAM, and its implementation on NFsim, one of the available platforms for such type of modelling. We show not only how the modelling of kTAM can be implemented, but we also explore the advantages of this modelling framework for kinetic simulations of kTAM and the easy way such models can be updated and modified. We present numerical comparisons both with classical numerical simulations of kTAM, as well as comparison in between four different kinetic variant of the TAM model, all implemented in NFsim as stand-alone rule-based models.

---

## 1. Introduction

Recent advances in DNA-based nano-technology have opened the way towards the systematic engineering of inexpensive, nucleic-acid based nano-scale

devices for a multitude of purposes [23, 25, 26, 28]. The field evolved dramatically in the past 10-15 years from a stage where careful manual design and intimate knowledge of DNA’s atomic structure were needed for the design of simple structures [33], to the current algorithmic approaches employing the use of universally engineered elementary building blocks, e.g. DNA-tiles or DNA-Origami, that are further functionalized and driven to self-assemble into the desired complex shapes, see [1, 2, 18, 38]. The Tile Assembly Model (TAM), which is one of the theoretical design platforms exploiting such a modular assembly scheme, employs the use of so-called DNA-tiles, which can be seen as unit square blocks with active glues on their four edges (North, East, South and West). These active glues, implemented in experimental settings by single-stranded DNA sticky-ends, are driving the self-assembly process and determine the controlled aggregation of the tiles into the desired structures.

Since its introduction [30], the TAM formalism has been used successfully both in designing complex assembly nanostructures and in providing predictions regarding the possible experimental outcomes of certain designs [29]. To this extent, the aTAM model has been expanded to a kinetic counterpart, kTAM [37]. kTAM incorporates two types of reactions: association of tiles to an assembly (forward reaction) and dissociation (reverse reaction), see e.g. Figure 2 a). In a forward reaction, any tile can attach to the assembly at any position, even if only a weak bond is formed; the rate of this reaction,  $r_f$ , is proportional to the concentration of free tiles in the solution. In the second reaction, any tile can detach from the assembly with rate  $r_{r,b}$  which depends exponentially on the total number  $b$  of the bonds between the tile and the assembly, as well as the strength of these bonds. Thus, tiles which are connected to the assembly by fewer or weaker bonds are more prone to dissociation than those which are strongly connected.

Previous computational modeling of kTAM has been performed almost exclusively using a special tailored version of the Gillespie’s stochastic simulation algorithm [4, 16, 39]. The assembly starts at  $t = 0$  from a seed structure. Then, in discrete time steps, tiles are added or detached from the assembly, according to the corresponding association and dissociation rates. Other computational studies focuss on the design and optimisation processes of the sticky end sequences for these tiles [12, 13]. In these studies, the authors depart from the rather un-realistic assumption of previous kinetic models regarding “perfect” orthogonal sticky end sequences, and take these interactions into considerations when analyzing the dynamics of the system and the subsequent errors incorporated within.

In our work, we propose a rule-based modelling approach for predicting the time evolution of kTAM systems and their variants. *Rule-based* modelling is a discrete modeling technique [14] in which molecules are represented as *agents* with a finite number of free *sites*. The sites allow for agent-agent binding, thus generating molecular complexes. *Rules* are defined based on local *patterns* rather than by the full specification of the reactants, and thus provide a compact representation on how agents interact. Thus, rather than handling explicitly a large number of model variables, within this framework we only have a small

number of local interaction rules. This makes the rule-based paradigm well suited in handling the problem of the combinatorial explosion of the state space, as is the case of modelling self-assembly or polymerization systems, see e.g., [35].

In the next section we briefly introduce the abstract Tile Assembly Model, aTAM, and its kinetic counterpart, kTAM. In Section 3 we describe the concept of agent- and rule-based modeling, some general notions about this framework, as well as some of the advantages of agent- and rule-based computational modeling. In Section 4 we give a rule-based model for the kTAM and compare its predictions to other available documented simulations of the kTAM. Also, we detail on the advantages of this model, both from a computational point of view, and from the designer point of view. Inspired by the general principle of kinetic modeling of chemical reactions, in Section 5 we introduce some modifications in the kinetic modeling level in the kTAM model. Then, using the new rule-based computational model, we analyze how this impacts the time evolution of the assembly in terms of assembly growth and error-fraction. Finally, in Section 6 we conclude our investigation. The source code of all of our kTAM models is freely accessible from [27].

## 2. The abstract Tile Assembly Model, and its kinetic counterpart

The abstract Tile Assembly Model (aTAM) [30, 37] is a generalization of Wang tile systems, customarily designed for the study of self-assembly systems. The basic components of the aTAM are non-rotatable unit square tiles, uniquely defined by the sets of four glues placed on top of their edges. The glues are part of a finite alphabet and each pair of glues is associated a strength value, determining the stability of a link between two tiles having these glues on the abutting edges. In most cases, it is assumed that the strength of two distinct glues is zero, while a pair of matching glues has strength either 1 or 2.

Let  $\Sigma$  be the *alphabet of glues*. A *tile type*  $t$  is uniquely determined by the ordered set  $(g_1, g_2, g_3, g_4) \in \Sigma^4$  of glues placed on its North, East, South, and West edges, respectively. A *tile system*  $T \subseteq \Sigma^4$  is a finite collection of different tile types. Let  $\rho : \Sigma^2 \rightarrow \mathbb{N}$  be the *strength function*. Unless otherwise specified, we assume that for all  $g, g_1, g_2 \in \Sigma$ ,  $g_1 \neq g_2$ ,  $\rho(g_1, g_2) = 0$ , while  $\rho(g, g) \in \{1, 2\}$ .

Given a tile system  $T$ , an *assembly*  $A$  is a partial mapping  $A : \mathbb{Z}^2 \rightarrow T$  assigning tiles to various elements from the two dimensional space. For each tile in an assembly, the *strength of its binding* is given by the sum of all strength values of the pairs of glues placed along the boundary between the tile and the assembly. A *Tile Assembly System* (TAS) is a structure  $(T, S, \rho, \tau)$ , where  $T$  is a tile system,  $S$  is an assembly called the seed structure,  $\rho$  is a strength function, and  $\tau$  is the temperature threshold for the assembly. Given an existing assembly  $A$ , such as the seed structure  $S$ , a tile can adjoin the assembly if its total strength of binding surpasses the temperature threshold  $\tau$ . In Figure 1 we present a TAS with 7 tile types and temperature  $\tau = 2$  which, starting from the seed tile, assembles a continuously growing structure corresponding to the evolution of the elementary cellular automaton (CA) “rule 90” (a.k.a. the XOR function) starting from the initial configuration ...000111... ; on this particular initial

configuration, the CA generates the well known *Sierpinski pattern*. Out of the 7 tile types in Figure 1, one can distinguish the tile used as seed (point d. in the figure), 2 tile types which assemble the zero- and one-boundary of the structure (point c. in the figure), and 4 rule-tile types, which fill the area in between the V shaped boundary (point d. in the figure). In a similar manner, one can consider for example any radius-half cellular automaton with  $n$  states and construct a TAS with  $O(n^2)$  tile types whose assembly simulates the evolution of the cellular automaton starting from a finite configuration (i.e., a configuration with finite non-zero entries). In particular, *rule 110* elementary cellular automaton, which is Turing universal [5], can also be assembled using a TAS with only 16 tile types (plus seed tile, plus tile types forming the boundary V-shape of the structure). In Appendix I we provide another example of a TAS with 21 tile types whose assembly simulates the evolution of the elementary cellular automaton “rule 30” starting from the initial configuration ...0001000... .

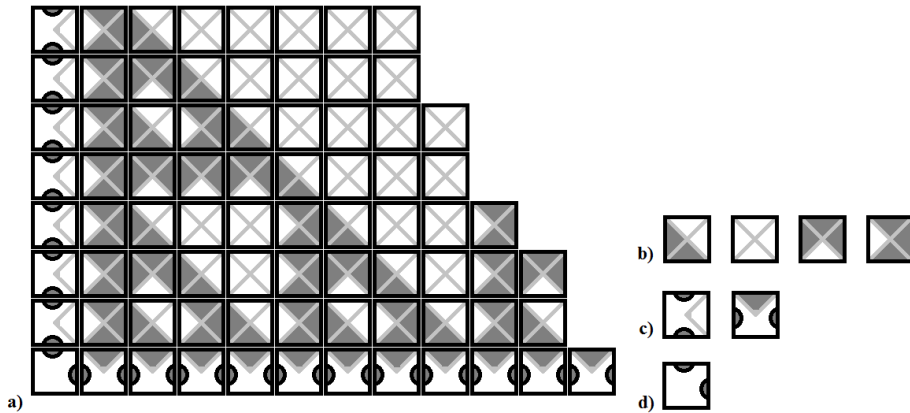


Figure 1: A 7-tile TAS simulating *rule 90* cellular automaton (a.k.a. the XOR function) on the input ...000111... (implemented as the continuously growing V-shaped boundary tiles), and assembling the Sierpinski pattern. a) Some initial part of the assembled structure. b)-d) the 7 tile system containing: b. 4 rule-tiles c. 2 boundary tiles, and d. one seed tile. The dark-circle glue has strengths 2, while the triangle glues have strength 1; the strength 0 glues on the seed and boundary tiles are not depicted.

The aTAM is a good formalization for crystal growth, incorporating the algorithmic assembly principles of these structures. Such theoretical framework is essential for the algorithmic design of assembling strategies for patterns and structures. The problem of finding minimal tile system for the unique assembly of a given pattern, PATS [9, 20], or the task of assessing the possible complexity of patterns assembled using specific tile types, [30], are only some of the important tasks which are most suitable to be addressed at this level. However, when analyzing, simulating, or assessing the experimental assembly of nano-structures, using e.g. DNA double-crossover tiles [17, 38] or DNA origami tiles [8, 40, 42], the aTAM is not suitable any more. This is because of the simplistic way in which this model reproduces the assembly system, not taking

into consideration the strength of the bonds, the various erroneous assemblies of tiles due to partial matching of the "sticky-end" strands, the reversibility of the reactions, i.e., detaching of tiles, etc. For dealing with all these aspects, the kinetic Tile Assembly Model (kTAM) has been proposed, [37], as a kinetic counterpart of the aTAM. Several variants of the kTAM exist, see e.g., [19, 32], however the main elements are similar. We present in the following the initial kinetic model introduced in [37], while other variations of this model are discussed later in the paper.

Within kTAM there are two types of reactions, each involving a tile and an assembly: i) addition of tiles (to the assembly), and ii) detaching of tiles (from the assembly), see e.g. Figure 2 a). Interactions between two tiles forming a new assembly, as well as interactions between two assemblies, are not taken into consideration in this kinetic model. In the first type of reactions, any tile can attach to the assembly on any position (up to the assumption that the alignment of the tiles is preserved), even if only a weak bond is formed. The rate of this reaction is proportional to the concentration of the free-floating tiles in the solution (assuming that all tile types are provided in similar concentrations). In the second type of reactions, any tile can detach from the assembly, with a rate which is exponentially correlated with the total strength of the bond between the tile and the assembly. Thus, tiles which are connected to the assembly by weaker or fewer bonds, are more prone to detaching than those which are strongly connected by several bonds.

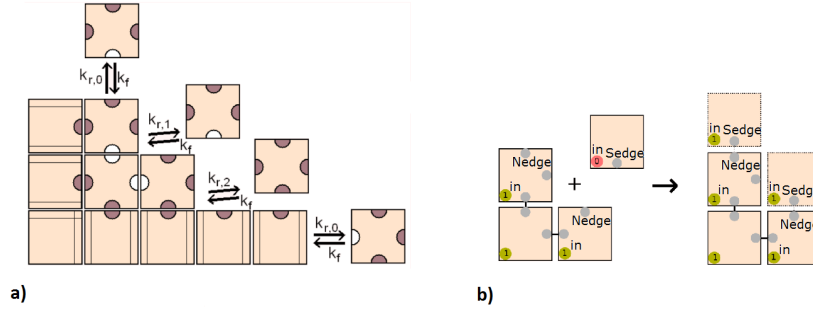


Figure 2: a) The dynamics of kTAM; b) Two possible North site association reaction in a rule-based implementation of TAM

Given a tile  $t \in \Sigma^4$ , the association (forward) rate constant  $r_f$  corresponding to the addition reaction of  $t$  to an existing assembly is

$$r_f = k_f [t] / \text{sec},$$

where  $[t]$  is the in-solution concentration of tile-type  $t$ , and  $k_f$  is a temperature dependent parameter. In case of DNA double-crossover (DX) tiles, this parameter is given by

$$k_f = A_f e^{-E_f/RT},$$

where  $A_f = 5 \cdot 10^8$  /M/sec,  $E_f = 4000$  cal/mol,  $R = 2$  cal/mol/K, and  $T$  is the temperature (in K), [37].

In the case of dissociation type (reverse) reactions, for a tile  $t$  which is connected to the assembly by a summed bond strength  $b$ , the rate constant  $r_{r,b}$  is given by

$$r_{r,b} = k_f e^{\Delta G_b^0 / RT},$$

where  $\frac{\Delta G_b^0}{RT}$  is the standard free energy needed in order to break  $b$  bonds. For DX tiles whose sticky-end glues are 5 base-long single-stranded DNA molecules,  $\Delta G_b^0$  can be estimated to

$$\Delta G_b^0 = e^{5b(11 - \frac{4000}{T}) + 3} \text{ cal/mol},$$

by using the nearest-neighbor model [31, 37]. The integer-base  $b$  parameter is ranging from 0 to 4, corresponding to cases ranging from a totally erroneous placement of the tile (no bonds connect it to the assembly) to fully integration of the tile into the assembly (all four sticky-ends are correctly matched).

In order to easily represent and scale the system, the free parameters involved in the formulas of  $r_f$  and  $r_{r,b}$  rate constants are re-distributed into just two parameters,  $G_{mc}$  and  $G_{se}$ , which are dimensionless parameters, and their values range in similar scale intervals:

$$r_f = \hat{k}_f e^{-G_{mc}},$$

$$r_{r,b} = \hat{k}_f e^{-bG_{se}},$$

where, in case of DX tiles,  $\hat{k}_f = e^3 k_f$  is adjusted in order to take into consideration possible entropic factors, such as orientation or location of the tiles.

The previous parameter re-distribution is made possible for the kinetic model introduced in [37] due to the assumption that all tile types are provided into the solution in similar concentrations, and that the consumption of the free monomers is negligible compared to the initial concentration. Because of this, the (free) tile concentration remains constant in time. Thus, the free parameters of kTAM are:  $G_{mc}$ , which is concentration dependent, and  $G_{se}$ , which is temperature dependent.

The ratio  $\frac{G_{mc}}{G_{se}}$  plays in some sense the role of the temperature  $\tau$  from the aTAM. Let  $\frac{G_{mc}}{G_{se}} = b - \epsilon$  for some bond strength  $b \in \{1, 2, 3, 4\}$  and a small  $0 < \epsilon < 1$ . Then, for a tile  $t$  attached to the assembly by bond strength  $b$  (or similarly any bond larger than  $b$ ), since  $\frac{r_f}{r_{r,b}} = e^{bG_{se} - G_{mc}} = e^{\epsilon G_{se}} > 1$ , the (forward) tile addition reaction on a neighborhood of  $t$  is favored, in comparison to the tile  $t$  detaching (reverse) reaction. Similarly, for a bond  $b' \in \{0, 1, 2, 3, 4\}$ ,  $b' < b$ , we have  $\frac{r_f}{r_{r,b}} = e^{b'G_{se} - G_{mc}} = e^{(b' - b + \epsilon)G_{se}} < 1$ , and thus, the tile  $t$  detaching reaction is favored in comparison to the addition reaction of new tiles abutting  $t$ . In conclusion, the growth of the assembly is favored only in the vicinity of those tiles which are attached to the crystal by a summed bond strength larger than or equal to  $b$ .

A major difference between aTAM and kTAM is that while in the first case, attachment of tiles is not allowed if the total bond strength is below temperature  $\tau$ , in the second case such erroneous attachments can happen. Indeed, although it may be unfavorable, a tile  $t$  may be locked in place inside an assembly even if its initial attachment to the crystal was done using a total bond strength below  $\frac{G_{mc}}{G_{se}}$ . This may happen if after attaching  $t$  a new tile  $t'$  is attached in such a way that the stability, i.e., total bond strength, of both  $t$  and  $t'$  is increased above the  $\frac{G_{mc}}{G_{se}}$  value. Thus, erroneous assemblies are possible in kTAM.

### 2.1. Computational modeling of the kTAM

Computational modeling of kTAM has been performed almost exclusively using a special tailored version of the Gillespie's Stochastic Simulation Algorithm (SSA), see e.g. [4, 16, 39]. The assembly starts at  $t = 0$  from a seed structure. Then, in discrete time steps, tiles are added or detached from the assembly, according to the corresponding association and dissociation rates. The structure of the assembly as well as the types of the tiles within are stored using a 2D array.

Let us assume that at some moment of time, the assembly consists of  $n$  tiles and there are a total of  $m$  empty sites around the crystal. During the next time step, there are two possible reaction types: a tile addition, or dissociation. The total tile association (on) rate is given by  $\sum_m r_f$ , while the total tile dissociation (off) rate is given by  $\sum_n r_{r,b}$  where, for each of the  $n$  tiles,  $r_{r,b}$  depends on the total bond strength  $b$  of that particular tile. Thus, the overall probability that the next event is an association reaction is  $Pr(on) = \frac{\sum_m r_f}{\sum_m r_f + \sum_n r_{r,b}}$ . The exact position, out of the  $m$  possible ones, at which a tile will be added, as well as the type of the tile added there, are chosen randomly, with equal probability. Regarding the dissociation reactions, the probability the next event is a tile removal reaction is  $Pr(off) = \frac{\sum_n r_{r,b}}{\sum_m r_f + \sum_n r_{r,b}}$ . Then, the tile to be removed from the assembly is chosen according to the dissociation rate distribution, namely, for a tile  $t$  with total bond strength  $b'$ , the dissociation probability is  $\frac{r_{r,b'}}{\sum_n r_{r,b}}$ . For determining the increment  $\Delta t$  of the time interval we use the probability distribution  $Pr(\Delta t) = (\sum_m r_f + \sum_n r_{r,b})e^{-(\sum_m r_f + \sum_n r_{r,b})\Delta t}$ . After an event is chosen and executed, the time is incremented with  $\Delta t$  and, for the next time step, the array is updated as well as all the reaction rates. (In practice, only part of the reaction rates are modified, while most of them remain unchanged.)

In the following section, we present a new computational model for the kTAM, based on a very different generic modeling framework using agent- and rule-based representations. As we show next, this modeling framework possesses significant advantages, particularly for modeling systems with potentially unbounded number of species, such as kTAM.

## 3. Modeling by agent- and rule-based representations

*Agent- and rule-based* modeling is a recent and promising discrete modeling approach [24], which has been used successfully in simulating biological sig-

nalling pathways [10, 11]. In this modeling paradigm molecules are represented as *agents* with a number of free *sites*. The sites, which may have several internal *states*, allow for agent-agent binding, thus generating molecular complexes. *Rules* are defined based on local *patterns* rather than full specification of the reactants, and thus provide a compact representation on how agents interact. In this way, rather than handling explicitly a high number of model variables, we only have a (often small) number of local interaction rules. This makes the agent- and rule-based paradigm well suited for handling the problem of the combinatorial explosion of the state space. The applicability of this approach for modeling protein self-assembly systems has also been investigated, see e.g. [35, 41] as in such systems, the number of different chemical species is potentially un-bounded.

From an algorithmic point of view, most of the rule-based modeling languages, as well as the corresponding computational modeling tools, represent agents (and multi-agent complexes) as graphs, while the reaction rules are implemented as graph rewriting rules. Thus, computational modeling of these systems is making use of well established and highly efficient graph algorithms.

*BioNetGen* [15] is one of the dedicated rule-based languages and computational model simulators. Several reaction simulators are based on the BioNet-Gen language (BNGL), e.g. Rule Bender [34] and NFsim [35]. In this study, the computational modeling of kTAM using agent- and rule-based representations has been performed using the NFsim modeling platform. This software suite augments the BNGL with several new features which proved to be useful for the simulation of the kTAM.

### 3.1. Computational modeling using BNGL and NFsim

In order to use the NFsim simulator, a model description has to be provided using the BNGL format. A BNGL file for NFsim contains six separate blocks: *parameters*, *molecule types*, *species*, *observables*, *functions*, and *reaction rules*. For a more detailed description of these blocks and of the syntax of the NFsim simulator we recommend [15] and [35]. The *parameters* section is used for introducing the constants used in the model. In the *molecule types* block, one defines the different agents used in the model, the number and identification name of its *sites*, as well as the different *states* each site can be placed in. For example, in the case of kTAM, besides a generator and a trash agent, we use only one generic agent called Tile containing several sites. Depending on the functionalization of this Tile agent, i.e., the particular state initialization of the various sites, it will represent a tile of one particular type or another. The *species* section introduces the initial particle numbers of various fully-instantiated agents, i.e., each site is in exactly one of its possible states, thus representing the initial species population.

In order to interrogate the current state of the system, we introduce *observables*. These are patterns of single or site-interacting multi-agents, with partial specifications of their site’s states or connectivity to other agents. During simulation, for each unit of time, the software reports the exact count of instances



fitting each of the these observables, in the current state of the system. For example, in the case of our Tile agents, we can define a site “in” with states 1 or 0, depending on whether that tile is in an assembly, or it is free-floating. Then, the observable `Molecules TilesInAssembly Tile(in~1)` will report (inside the observable `TilesInAssembly`) the number of tiles which are part of an assembled complex at each unit of time.

The *reaction rules* section describes the agent interaction rules based on which the system evolves in time. For a detailed description of the syntax of rules in NFsim see [35]. As in the case of observables, the reaction rules are defined based on patterns rather than full specifications of the reactant’s site’s states or connectivity. For example, in the modelling of DNA tile systems, the primary agent, *Tile*, may have a number of sites, including `Nedge`, `Eedge`, `Sedge`, and `Wedge`, each with internal states corresponding to possible glues of these tiles. Then, both tile-association and tile-dissociation reactions can be implemented by appropriate local rules. For example, the addition of a tile to a free North site, as depicted in Figure 2 b), can be implemented by the local rule,

```
Tile(Nedge,in~1) + Tile(Sedge, in~0) -> Tile(Nedge!1,in
~1).Tile(Sedge!1,in~1), kon}
```

where as before, the site `in` indicates whether the tile is in an assembly (`in~1`) or it is free-floating (`in~0`). The above rule can be interpreted as follows: a `Tile` with an unbounded `Nedge` site placed inside the assembly (i.e., `in~1`) interacts with a free `Tile` (i.e., `in~0`) with an unbounded `Sedge` site, and the two become bonded on the sites `Nedge` and `Sedge`; the reaction has a kinetic rate constant `kon`. Tile dissociation reactions, which are dependent on the total bond strength of the tiles, can be implemented in a similar manner.

### 3.2. General advantages and disadvantages of rule-based modelling in simulating Tile Assembly Systems

There are a number of advantages in using a rule-based modelling approach for kTAM. Since this is a coarse-grain modelling framework, it allows the examination of a very diverse family of *observables*. Thus, the system can be analyzed extensively and both final and intermediate states can be inquired with detailed precision. Also, due to the current availability of appropriate software frameworks, numerical simulations of rule-based models are easy to run. Such simulations can be written in pseudocode, using e.g. BNGL [14] or  $\kappa$  [21]. Thus, the emphasis is placed on describing the system’s reaction rules, and not in dealing with the numerical simulation algorithm. Hence, custom simulations are easy to create, update, and modify.

Another advantage strongly related to the current availability of numerical simulator softwares is that within these numerical simulators one is not restricted to have a pre-initialized number of different species in the system. Namely, new species (i.e. complexes of linked agents) are created and simulated on the go in these numerical simulators. This is extremely important when simulating self-assembly systems, as the assembly grows in an un-ordered and

non-deterministic fashion. Thus, there is potentially an un-bounded number of different species that the system will encompass during its evolution, each corresponding to a different possible state of the assembly.

A major drawback of previous modelling approaches was that they could express only those systems where there exists a single growing assembly, and all the reactions (addition and dissociation) were between this unique assembly and the free floating tiles. This situation however, does not cover the case in which two partial assemblies, each consisting of more than one tile, are interacting. However, using the rule-based modelling framework one can implement such reactions too. This opens the possibility of modelling various variants of TAM (which are closer to experimental implementations), accounting for both assembly–tile interactions and assembly–assembly interactions, e.g., staged or hierarchical tile assembly models [22].

On the down side, modelling Tile Assembly Systems dynamics within a rule-based modelling framework does not solve all modelling problems. For example, the spatial characterisation/modelling of the emerging assemblies, although sometimes very important, can not be directly captured, and only in some very restricted cases can be taken into consideration. Indeed, rule-based models operate by use of local partial configurations. If such a locality constraint can be expressed as a local arrangements of the agents, such as a tile attaching horizontally to an assembly and detecting also an immediate vertical border neighboring (or overlapping), then local rules can be implemented to take this into considerations. However, if the generated neighboring (or overlapping) situation emerges out of non-contiguous arrangements of the existing assembly, then such situations can not be detected, thus generating modelling errors.

Another related problem represents the visualizations and the spatial characterisations of the emerged assemblies. Currently, although some of the rule-based modelling software, such as RuleBender [34] or RuleStudio [3], have incorporated some form of visualisation, these become easily obsolete for visualizing large complexes, such as the ones emerging from tile self-assembly systems. On the other hand, such models preserve within all the information needed for reconstructing the conformation of the emerging structures. Moreover, in the case of Tile system, since each agent/tile has the same shape, e.g., square, and same conformation of the linking sites, e.g., North, East, South, and West, the task of reconstructing the emerging assemblies becomes a matter of implementation, and will be considered in future studies. However, as we will show next, although a visual inspection of the emerging structures is not currently available, the output of such rule-based models, NFsim in particular - as the modelling environment of choice in this study-, is still enough for performing an informative analyses of the emerging assemblies.

#### 4. Rule based modeling of kTAM

In this section we will provide further details on our rule-based implementation of kTAM, as well as a comparison between its numerical simulation and the numerical simulation of a similar assembly process modelled in Xgrow [39].

Also, we will consider two small alterations of the model and analyze how these modifications in the modeling paradigm affect the assembly growth and the frequency of errors emerging in the assembly process. The source code of our kTAM models is freely accessible from [27].

#### 4.1. Model description

The rule-based model introduced here is lightly customized for the case when we need only 4 glues, namely,  $\{0, 1, 2, 3\}$ , on the tile edges, including the null glue; a more general model for  $n$  glues can be similarly created, even using an automated approach in case of large values of  $n$ . The model will be used to simulate a tile assembly system generating the Sierpinski pattern according to the kinetics described in [37]. The Sierpinski tile system consists of a seed tile, two boundary tiles and four rule tiles. The seed tile has two strength-2 glues (north and east edges) and two strength-0 glues (south and west edges). The aggregate first grows along the direction of the strength-2 edges by the addition of boundary tiles. Moreover, the boundary tiles have each a strength-0 edge that restricts the growth of the aggregate to other directions. The rule tiles on the other hand encode the XOR operation that underlies the Sierpinski triangle pattern. The tile types and aggregate growth of the Sierpinski tile system are depicted in Figure 1.

In our BNGL model of kTAM we used a generic tile agent whose site's states will determine the desired tile type. The specification of a tile agent is given by,

```
Tile (N1, E1, S1, W1, Ng~0~1~2~3, Eg~0~1~2~3, Sg~0~1~2~3, Wg
      ~0~1~2~3, Nm~0~1, Em~0~1, Sm~0~1, Wm~0~1, in~0~1, bnd
      ~0~1~2~3~4~5, act~none~ckglue~del~inc2~dec2~incl~decl)
```

where sites are separated by commas (,) and possible states of a certain site are separated by tildes (~). The first group of 4 sites, N1, E1, S1 and W1, encode the binding edges of a tile agent. For a square tile, these correspond to the northern, eastern, southern and western links of a tile. All reaction rules in the BNGL model where a bond is either established or broken utilize these sites. All other sites of a tile agent provide only the context where such bond formation occurs. The second group of sites, Ng, Eg, Sg, Wg, encode the glues of the four edges.<sup>1</sup> Each of these sites has four possible states corresponding to the four edge labels defined in the Sierpinski tile system. The encoding is as follows: 0 for the strength-0 glue, 1 for the strength-2 glue, 2 for the strength-1 black glue and 3 for the strength-1 white glue. The third set of sites, Nm, Em, Sm, Wm, is used to indicate whether a tile has matching glues with its neighbors. If such a site is in state 1, the tile has a matching glue with its neighbor on that edge. If it is in state 0, there is a mismatch on that edge.

---

<sup>1</sup>Although in theory a site can have various states and be used for connecting with other sites, from our implementation in NFsim we saw that such cases may generate undesired numerical errors in the simulation. Thus, we use sites N1, E1, S1, and W1 for implementing the physical bond, and sites Ng, Eg, Sg, and Wg for capturing the glue state of the corresponding edge.

In addition, a tile has three more sites unrelated to the four edges. The first of these, the site `in`, differentiates between the tiles within the assembly, i.e., `in~1`, and the free floating ones, i.e., `in~0`. The second site, `bnd`, determines the rate of dissociation of a tile. The states of this site encode the total strength by which the tile is bound to the assembly. Lastly, the site `act` signals the next action to be implemented on this tile.

For the reaction rules, we can distinguish several types of rules, with different purposes. Naturally, there are groups of reaction rules which model tile association and dissociation. However, there are also complementary rules which check glue matching, link a newly added tile to its neighbors, or update the bond strength of a tile, etc.

In Appendix I we provide detailed descriptions of these rules and their mechanistic behaviours. The complete source code of our kTAM model is freely accessible from [27].

#### 4.2. Simulation results, comparison, and analysis

To test the correctness of our rule-based implementation, we compared the simulated assembly growth of the Sierpinski TAS in both NFsim and Xgrow. Figure 3 a) shows the resulting assembly sizes (from both numerical simulation frameworks) after 100 simulated seconds for various values of the parameter  $G_{se}$  and a fixed value of 16 for the parameter  $G_{mc}$ . Each plotted point is an average over 30 simulation runs.

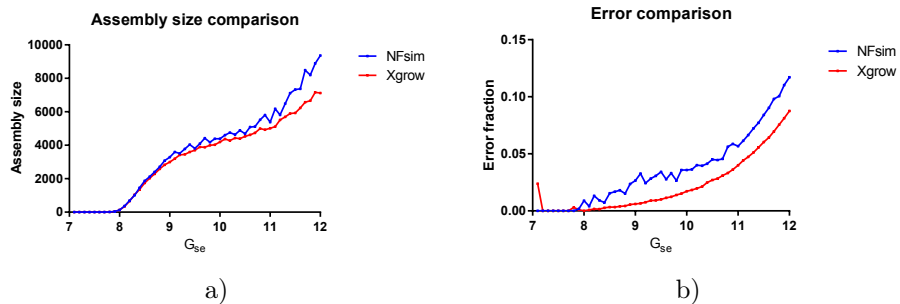


Figure 3: Comparison between the numerical predictions of a) aggregate growths and b) error fraction (calculated as the number of mismatched bonds over the number of tiles in the assembly) performed with NFsim, for rule-based implementation, and Xgrow, a dedicated simulator for kTAM. Both numerical simulators are implementing the same assembly system forming the Sierpinski pattern; the data is collected after simulating 100 seconds of model time and averaged over 30 runs. The  $G_{mc}$  parameter is set to 16 in both comparisons.

As it can be seen from the Figure 3 a), the NFsim simulation results on aggregate growth closely follow the corresponding Xgrow simulations, thus supporting the validity of the rule-based implementation. In both simulations, there is almost zero growth when the temperature is greater than two (i.e.  $G_{se} < 8$ ), as expected in the TAS, since the border tiles can only bind to the seed with

strength two. When the temperature decreases larger assemblies are formed, as a lower level of cooperation for the incoming tile becomes sufficient for growth.

Next, we study how well the rule-based model captures assembly errors by comparing the error fractions of NFsim simulations of the Sierpinski TAS with its Xgrow simulations. Figure 3 b) shows the error fraction, measured as the number of mismatched bonds per the assembly size, for the same parameter values as in Figure 3 a). Once again, the curve of the error fraction obtained from the rule-based model closely matches its counterpart from Xgrow, as both NFsim and Xgrow simulations confirm that larger aggregates with more mismatched bonds form at lower temperatures. This is explained by the fact that as the temperature is decreased (by increasing the value of  $G_{se}$ ), incoming tiles which are matching only one of their glues with some un-covered edge in the assembly are more likely to survive within the aggregate, because of the exponentially less likelihood of the dissociation reactions.

While the growth rates of both NFsim and Xgrow simulations are almost perfectly aligned, in terms of error fraction, there seem to be a systematic (approx. 0.01) increase for the error fraction reported by NFsim simulations. At this moment it is difficult to further analyze how substantial and/or significant these differences are, for example with respect to the effects on the resulting assemblies and/or the errors propagated in their patterns. This is mainly due to several aspects. First, currently there is no appropriate visualisation capability of the structural conformation of the assembly generated by the NFsim modelling framework.<sup>2</sup> Second, both Xgrow and NFsim simulators use a stochastic modelling approach. This implies that for analyzing the trends of the model dynamics, one has to collect data over a large-enough family of simulations, and average the results. However, currently, there are no methods to automatically collecting data regarding shape and patten dynamics of the resulting simulations, either in NFsim or in Xgrow. Finally, even if such data collection could be overcome, we found it hard to conceptualize a numerical measure of what could be described as an average pattern alteration due to erroneous assembly. All these aspects are considered by us as a motivation for further work.

## 5. Variations of the kTAM model

As previously mentioned, one of the advantages of the rule-based modelling methodology is the ease by which one can modify a model and re-run the simulations. To demonstrate this we consider two possible (and well justified) modifications of the kTAM model, which will thus generate four variant models, depending on whether we implement one, the other, both, or neither of these modifications.

The first modification refers to the kTAM assumption that each tile can attach to an assembly on each free position, with the argument that if no com-

---

<sup>2</sup>We expect to be able to address this in further developments, as we are currently working at a Tile-System visualisation tool for NFsim simulations.

| Alternatives | Slot | vs | Site |
|--------------|------|----|------|
| Un-Matching  | S1   |    | S2   |
| vs           |      |    |      |
| Matching     | S3   |    | S4   |

Table 1: Four possible scenarios based on two modelling alternatives: i) Slot vs Site interactions: Tiles can associate to the assembly on a free position (slot) basis, or on a free edge (site) basis; ii) Reactivity of Un-Matching vs Matching glues: Tiles can attach to the assembly in any circumstance or if at least one matching glue exists

mon glues would keep this tile in place, the detachment reaction has a very high reaction rate. In the alternative we consider here, a certain tile can attach to an assembly on some position only if there exists at least one common glue on two opposite edges of a tile in the assembly and the free-floating tile. We will denote this alternative choice as the “Un-Matching vs Matching” alternative.

The above considerations lead also to a second possible modification within the kTAM model. In the classical assumption, a tile attachment reaction assumes one free-floating tile is inserted on an empty position (or *slot*) of the assembly, independent on whether it is matching one, two, or zero glues. An alternative to this is that each individual edge of a free-floating tile can interact with a corresponding (i.e., with cardinal opposite orientation) free edge within the assembly. This modification has the potential to generate different behaviours, as the number of free edges is different than the number of empty positions. Moreover, in the framework of the first modification above, if a free-floating tile is matching the glues of both edges fitting a free position in the assembly, it has twice the chance of attaching to that position in comparison with a tile which is matching only one glue from the two neighboring edges. We will denote this second alternative choice as the “Slot vs Site” alternative.

Based on the above two possible modifications we constructed and explored the dynamics of four possible scenarios, for the cases when each modification is either implemented or not, see Table 1 for the description of these scenarios. Note that according to the partitioning in Table 1, scenario S1 stands for the classical kTAM model. We implemented rule-based models for each of the 3 new scenarios, and run parallel simulations for 100 (model-time) seconds, using the parameter  $G_{mc} = 16$  and  $G_{se} = 8.2$ . The comparative time evolution of the aggregate size and of the error fraction, computed as the ratio between the number of mismatched bonds over the total number of tiles in the aggregate, are presented in Figure 4.

Based on the numerical simulations we can conclude that while the “Un-Matching vs Matching” alternative had little effect over aggregate growth, i.e. scenarios S1, S3 and respectively S2, S4 present very similar growth behaviours, see Figure 4 a), it had some impact over the accumulated errors. Indeed, the numerical simulations show that allowing only matching interactions (i.e., scenarios S3 and S4) reduces to some extent the error fraction, see Figure 4 b), albeit these error fractions are relatively small for this particular choice of parameters  $G_{mc}$  and  $G_{se}$  for all 4 scenarios. An opposite dynamics behaviour can be seen for the “Slot vs Site” alternative, which has very little influence on the

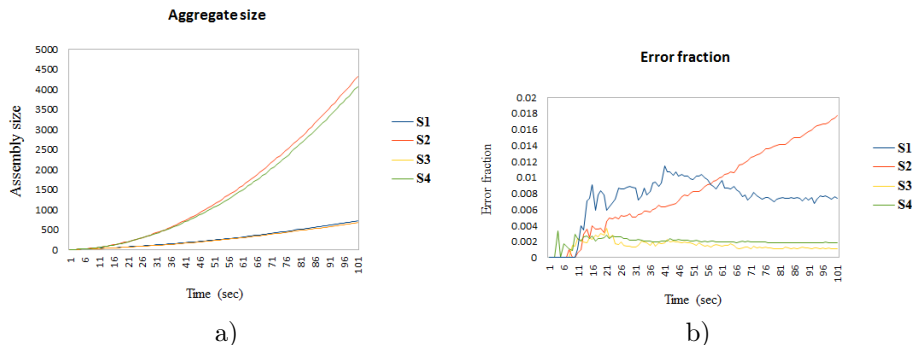


Figure 4: a) Aggregate growth for scenarios S1-S4. Larger aggregates form with site interactions (scenarios S2, S4) as there are more possible association reactions that can be triggered. On the other hand, aggregate growth is less affected by the distinction based on matching glues. b) Error fraction of the assembly in scenarios S1-S4, calculated as the number of mismatched bonds over the number of tiles in the assembly. Allowing only matching interactions (scenarios S3, S4) significantly reduces the error fraction, albeit low overall error rates. For each scenario, the data is collected from simulations covering 100 seconds of model time, and averaged over 30 runs; we use parameters  $G_{mc} = 16$  and  $G_{se} = 8.2$ .

error fraction, but highly impacts the aggregate growth.

The ease by which typical rule-based models can be modified and/or updated is an important generic advantage of this modelling framework. In this study, we considered only two such alterations of the classical kTAM model. However, taking further advantage of this modelling aspect, for the particular study of Tile Assembly Systems, or even for general DNA-based algorithmic assembly systems, remains to be further explored in more depth in future studies.

## 6. Conclusions

DNA self-assembly is nowadays an established branch of nanotechnology. The advantages of DNA as a medium for nano-fabrications does not come from being a super-material, e.g., in terms of strength, conductivity, or stiffness. Rather, it comes from its highly predictable, and thus programmable, assembly properties, and its intrinsic bonding interactions with a large class of organic and anorganic matter. In this paper we report on a new approach for modelling the kinetics of DNA-based Tile Assembly Systems via a rule-based modelling methodology; we concentrate on the use of NFsin [35] as our de-facto choice for a rule-based modelling software, and on kTAM as our target model to be captured.

Using a rule-based modelling approach for capturing the dynamics of the kTAM model brings a number of advantages. Since this is a coarse-grain modelling framework, it allows us to trace a very diverse family of *observables*, both during the system evolution and in its final state. Also, the availability of several rule-based software frameworks makes the development of these models as well as their numerical simulations simple to perform. Such models can thus be

written in pseudocode using specific protocols, e.g. BNGL [14] or  $\kappa$  [21], and run on the appropriate platforms. Thus, the emphasis is placed on describing the system’s reaction rules, and not in dealing with the numerical simulation algorithm. Hence, custom simulation are easy to create, update, and modify.

One of the down-sides of previous modelling approaches was the lack of tools for modelling multiple-seeded assemblies, each evolving either independently or interacting with one-another. This however restricts us for modelling a number of TAS variants, such as staged or hierarchical TAS. Using a rule-based modelling framework we are not anymore bound to this constraint, as now we can implement reactions between two partial assemblies, each consisting of more than one tile, interacting and changing their local configurations, e.g., binding to one-another, or just exchanging signals. This opens the possibility of modelling different variants of TAM (which are closer to experimental implementations), accounting for both assembly–tile interactions and assembly–assembly interactions, see e.g. [22].

In our study, we first created a rule-based model of kTAM. Using the same kinetic parameters as in previous numerical implementations of kTAM, we compared the prediction of those models (regarding time-growths and error-fraction of the assembly) with that of our model. This comparison shows that the rule based modeling framework is at least as expressive as the previous state-of-the-art modelling methodologies for kTAM. Then, we introduced some small variations into the model and studied how this affects the overall behaviour of the system. In particular, while the “Un-Matching vs Matching” alternative had little effect over aggregate growth, it had considerable impact over the accumulated errors. At the same time, the “Slot vs Site” alternative exhibits an opposite behaviour, that is, it has very little influence on the error fraction, but highly impacts the aggregate growth.

Future studies will concentrate on the implementation and analysis of various other types of TAM formalisms, such as stage, hierarchical, signal-passing, etc. A milestone for this approach will be overcome when one would succeed in capturing both the dynamic behaviour of a general DNA assembly process and its 3D structural details, e.g. capturing the space- and time-dynamics of a DNA Origami assembly process. Partially related to this aspect, an implementation of a tile-assembly visualizer for the output of these models is also to be considered in future studies.

Another future direction for improvement is the incorporation of more accurate sequence interactions, as implemented for example by Evans et al. [12, 13]. Currently, all glue-glue interactions are assumed to be implemented using orthogonal DNA sequences; in practice, such thing is almost impossible to accomplish. However, the rule-based modelling framework has the capability (in principle) of differentiating between different energetic levels of the binding affinities.



## References

- [1] Erik Benson, Abdulmelik Mohammed, Alessandro Bosco, Ana I. Teixeira, Pekka Orponen, and Björn Högberg. Computer-aided production of scaffolded DNA nanostructures from flat sheet meshes. *Angewandte Chemie International Edition*, 55(31):8869–8872, 2016.
- [2] Erik Benson, Abdulmelik Mohammed, Johan Gardell, Sergej Masich, Eugen Czeizler, Pekka Orponen, and Björn Högberg. DNA rendering of polyhedral meshes at the nanoscale. *Nature*, 523(7561):441–444, 2015.
- [3] Bob Buffone, Bill Masek, and Monte Brown. The RuleStudio Simulator. <http://kappamodeler.github.io/rulestudio/>.
- [4] Anshul Chaurasia, Sudhanshu Dwivedi, Prateek Jain, and Manish K. Gupta. Xtile: An error correction package for DNA self-assembly. In *Proc. 6th Annual Conference on Foundations of Nanoscience (FNANO 09)*, 2009.
- [5] Matthew Cook. Universality in Elementary Cellular Automata. *Complex Systems*, 15(1):1–40, 2004.
- [6] A. Credi, M. Garavelli, C. Laneve, S. Pradalier, S. Silvi, and G. Zavattaro. **nanok**: A calculus for the modeling and simulation of nano devices. *Theoretical Computer Science*, 408(1):17 – 30, 2008.
- [7] Alberto Credi, Marco Garavelli, Cosimo Laneve, Sylvain Pradalier, Serena Silvi, and Gianluigi Zavattaro. Modelization and simulation of nano devices in **nanok** calculus. In *Computational Methods in Systems Biology, International Conference, CMSB 2007, Edinburgh, Scotland, September 20-21, 2007, Proceedings*, pages 168–183, 2007.
- [8] Eugen Czeizler and Pekka Orponen. Fault tolerant design and analysis of carbon nanotube circuits affixed on DNA origami tiles. *IEEE Transactions on Nanotechnology*, 14(5):817–877, 2015.
- [9] Eugen Czeizler and Alexandru Popa. Synthesizing minimal tile sets for complex patterns in the framework of patterned DNA self-assembly. *Theor. Comput. Sci.*, 499:23–37, 2013.
- [10] Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Rule-based modelling, symmetries, refinements. In Jasmin Fisher, editor, *FMSB*, volume 5054 of *Lecture Notes in Computer Science*, pages 103–122. Springer, 2008.
- [11] Vincent Danos, Jérôme Feret, Walter Fontana, and Jean Krivine. Scalable simulation of cellular signaling networks. In *Proceedings of the 5th Asian Conference on Programming Languages and Systems*, Lect. Not. in Comp. Sci., pages 139–157. Springer-Verlag, 2007.

- [12] Constantine G. Evans and Erik Winfree. *DNA Sticky End Design and Assignment for Robust Algorithmic Self-assembly*, pages 61–75. Springer International Publishing, 2013.
- [13] Constantine Glen Evans. *Crystals that count! Physical principles and experimental investigations of DNA tile self-assembly*. PhD thesis, California Institute of Technology, 2014.
- [14] J. Faeder, M. Blinov, and W. Hlavacek. Rule-based modeling of biochemical systems with bionetgen. In *Systems Biology*, volume 500, pages 113–167. Humana Press, 2009.
- [15] James R. Faeder, Michael L. Blinov, Byron Goldstein, William, and S. Hlavacek. Rule-based modeling of biochemical networks. *Complexity*, 10:22–41, 2005.
- [16] T. Fochtman and M. Patitz. Tile assembly simulator: A software package for tile-based algorithmic self-assembly. In *Self-Adaptation and Self-Organizing Systems Workshops (SASOW), 2013 IEEE 7th International Conference on*, pages 27–28, Sept 2013.
- [17] Tsu-Ju Fu and Nadrian C. Seeman. DNA double-crossover molecules. *Biochemistry*, 32(13):3211–3220, Apr 1993.
- [18] Kenichi Fujibayashi, Rizal Hariadi, Sung Ha Park, Erik Winfree, and Satoshi Murata. Toward reliable algorithmic self-assembly of DNA tiles: A fixed-width cellular automaton pattern. *Nano Letters*, 8(7):1791–1797, 2008.
- [19] Kenichi Fujibayashi and Satoshi Murata. Precise simulation model for DNA tile self-assembly. *IEEE Transactions on Nanotechnology*, 8(3):361–368, May 2009.
- [20] Mika Göös, Tuomo Lempiäinen, Eugen Czeizler, and Pekka Orponen. Search methods for tile sets in patterned DNA self-assembly. *J. Comput. Syst. Sci.*, 80(1):297–319, 2014.
- [21] W. Hlavacek, J. Faeder, M. Blinov, R. Posner, M. Hucka, and W. Fontana. Rules for modeling signal-transduction systems. *Science's STKE*, 2006.
- [22] Patitz Matthew J. *An Introduction to Tile-Based Self-assembly*, pages 34–62. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [23] Y. Krishnan and F. Simmel. Nucleic acid based molecular devices. *Angew Chem Int Ed Engl*, 50(14):3124–56, 2011.
- [24] Lucas Laursen. Computational biology: Biological logic. *Nature*, 462:408–410, 2009.

- [25] Jiang Li, Hao Pei, Bing Zhu, Le Liang, Min Wei, Yao He, Nan Chen, Di Li, Qing Huang, and Chunhai Fan. Self-assembled multivalent DNA nanostructures for noninvasive intracellular delivery of immunostimulatory cpg oligonucleotides. *ACS Nano*, 5(11):8783–8789, 2011.
- [26] Kyle Lund, Anthony J. Manzo, Nadine Dabby, Nicole Michelotti, Alexander Johnson-Buck, Jeanette Nangreave, Steven Taylor, Renjun Pei, Milan N. Stojanovic, Nils G. Walter, Erik Winfree, and Hao Yan. Molecular robots guided by prescriptive landscapes. *Nature*, 465:206–210, May 2010.
- [27] Abdulmelik Mohammed and Eugen Czeizler. NFsim modelling of kTAM. <https://github.com/eugencz/NFsim-modelling-of-kTAM>, 2017.
- [28] Lulu Qian, Erik Winfree, and Jehoshua Bruck. Neural network computation with DNA strand displacement cascades. *Nature*, 475:368–372, Jul 2011.
- [29] J.H. Reif, H. Chandran, N. Gopalkrishnan, and T. LaBean. *Self-Assembled DNA Nanostructures and DNA Devices in Nanofabrication Handbook*. CRC Press, 2012.
- [30] Paul W. K. Rothmund and Erik Winfree. The program-size complexity of self-assembled squares. In *Proc. 32nd Annual ACM Symposium on Theory of Computing (STOC 2000)*, pages 459–468, New York, NY, USA, 2000. ACM.
- [31] John SantaLucia, Jr., Hatim T. Allawi, and P. Ananda Seneviratne. Improved nearest-neighbor parameters for predicting DNA duplex stability. *Biochemistry*, 35(11):3555–3562, 1996.
- [32] Rebecca Schulman and Erik Winfree. Programmable control of nucleation for algorithmic self-assembly. *SIAM Journal of Computing*, 39(4):1581–1616, 2009.
- [33] N. Seeman. Nucleic acid junctions and lattices. *Journal of Theoretical Biology*, 99(2):237–247, 1982.
- [34] A.M. Smith, W. Xu, Y. Sun, J.R. Faeder, and G.E. Marai. Rulebender: Integrated modeling, simulation and visualization for rule-based intracellular biochemistry. *BMC Journal of Bioinformatics*, 13:1–16, Jun 2012.
- [35] M. Sneddon, J. Faeder, and T. Emonet. Efficient modeling, simulation and coarse-graining of biological complexity with nfsim. *Nature methods*, 8(2):177–183, 2011.
- [36] Cristian Versari and Gianluigi Zavattaro. Complex functional rates in the modeling of nano devices (extended abstract). *Electronic Notes in Theoretical Computer Science*, 293:3 – 15, 2013.

- [37] Erik Winfree. Simulations of computing by self-assembly. Technical Report CaltechCSTR:1998.22, California Institute of Technology, May 1998.
- [38] Erik Winfree, Furong Liu, Lisa A. Wenzler, and Nadrian C. Seeman. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394:539–544, Aug 1998.
- [39] Erik Winfree, Rebecca Schulman, and Constantine Evans. The Xgrow Simulator. <http://www.dna.caltech.edu/Xgrow/>.
- [40] Na Wu and Itamar Willner. DNAzyme-controlled cleavage of dimer and trimer origami tiles. *Nano Letters*, 16(4):2867–2872, 2016.
- [41] Jin Yang and William S Hlavacek. The efficiency of reactant site sampling in network-free simulation of rule-based models for biochemical systems. *Physical Biology*, 8(5), 2011.
- [42] Zhao Zhao, Yan Liu, and Hao Yan. Organizing DNA origami tiles into larger structures using preformed scaffold frames. *Nano Letters*, 11(7):2997–3002, 2011.

## Appendix I

*A TAS implementing the evolution of the elementary cellular automaton “rule 30”*

In Figure 5 we present a TAS with 21 tile types and temperature  $\tau = 2$  which, starting from the seed tile, assembles a continuously growing structure corresponding to the evolution of the elementary cellular automaton “rule 30” starting from the single-1 initial configuration, i.e., the configuration ...0001000... . Out of the 21 tile types in Figure 5, one can distinguish the tile used as seed (point d. in the figure), 4 tile types which assemble the zero boundary of the structure (point c. in the figure), and 16 rule-tile types, which fill the area in between the V shaped boundary (point d. in the figure).

*Detailed description of a rule-based implementation of the kTAM model*

The rule-based model introduced here is lightly customized for the case when we need only 4 glues, namely,  $\{0, 1, 2, 3\}$ , on the tile edges, including the null glue; a more general model for  $n$  glues can be similarly created, even using an automated approach in case of large values of  $n$ . The model is used to simulate a tile assembly system generating the Sierpinski pattern according to the kinetics described in [37]. The Sierpinski tile system consists of a seed tile, two boundary tiles and four rule tiles. The seed tile has two strength-2 glues (north and east edges) and two strength-0 glues (south and west edges). The aggregate first grows along the direction of the strength-2 edges by the addition of the boundary tiles. Moreover, the boundary tiles each have a strength-0 edge that restrict the growth of the aggregate to two directions. The rule tiles on

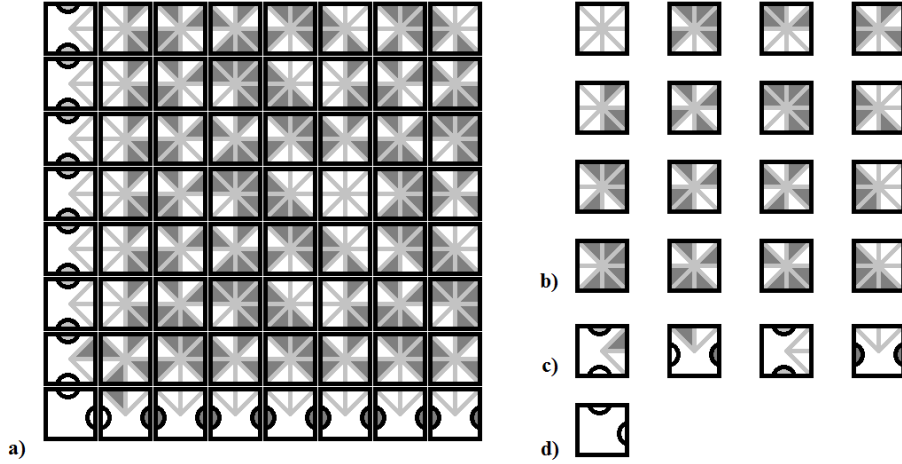


Figure 5: A 21-tile TAS simulating rule 30 cellular automaton. a) Some initial part of the assembled structure. b)-d) the 21 tile system containing: b. 16 rule-tiles c. 4 boundary tiles, and d. one seed tile. The strengths of the glues for the seed and boundary tiles are 2, while the strengths of the glues of the rule tiles are 1. Each glue of a rule-tile in b. is formed from a binary pair. The input of the rule-tile is given by the left and bottom glues, i.e., four bits. The output is given by the two-bit pair corresponding to the top side of the tile which is always identical to the bit-pair on the right side of the tile. For example, for the top right tile in b) the input is given by the pairs (1,1),(0,0), and the output is the pair (0,1). This coding is derived from the rule 30 elementary cellular automaton.

the other hand encode the XOR operation that underlies the Sierpinski triangle pattern. The tile types and aggregate growth of the Sierpinski tile system are depicted in Figure 1.

In our BNGL model of the Sierpinski tile system, we used a generic tile agent whose site's states determine the desired tile type. The specification of a tile agent is given by,

```
Tile (N1, E1, S1, W1, Ng~0~1~2~3, Eg~0~1~2~3, Sg~0~1~2~3, Wg
~0~1~2~3, Nm~0~1, Em~0~1, Sm~0~1, Wm~0~1, in~0~1, bnd
~0~1~2~3~4~5, act~none~ckglue~del~inc2~dec2~inc1~dec1),
```

where sites are separated by commas (,) and possible states of a certain site are separated by tildes(~). The first set of sites, (N1, E1, S1, W1), encode the binding edges of a tile agent. For a square tile, these correspond to the northern, eastern, southern and western links of a tile. All reaction rules in the BNGL model where a bond is either established or broken utilize these sites. All other sites of a tile agent provide only the context where such bond formation occurs. The second set of sites, (Ng, Eg, Sg, Wg), encode the edge glues of the four edges.<sup>3</sup> Each of these sites have four possible states corresponding to the four

<sup>3</sup>Although in theory a site can have both various states and be used for connecting with

edge labels defined in the Sierpinski tile system. The encoding is as follows; 0 to strength-0 glue, 1 to strength-2 glue, 2 to strength-1 black glue and 3 to strength-1 white glue. The third set of sites, ( $N_m$ ,  $E_m$ ,  $S_m$ ,  $W_m$ ), are used to indicate whether a tile has matching glues with its neighbors. If such a site is in state 1, the tile has a matching glue with its neighbor on that edge. If it is in state 0, there is a mismatch on that edge.

In addition, a tile has three more sites unrelated to the four edges. The first of these, the site `in`, differentiates the tiles which are part of the assembly from the free floating ones. A tile has a state 1 in its `in` site if it is in the assembly, or a state 0 otherwise. The second site, the site `bnd`, determines the rate of dissociation of a tile. The states of the site `bnd` represent the total strength by which the tile is hold within the assembly. Finally, the site `act` signals the next action to undertake on its tile.

For the reaction rules, we can distinguish several types of rules, with different purposes. Naturally, there are groups of reaction rules which model tile association and dissociation. However, there are also accompanying and administrative reaction rules, e.g., rules which check glue matching, which link a newly added tile to its neighbors, which update the bond strength of a tile, etc.

In a tile association reaction, a free floating tile attaches to a tile in the assembly. The attachment can be between any two tile types in kTAM, as long as one is free and the other is in the assembly. Thus, in a reaction rule of tile association, one tile agent has state 1 in site `in`, while the other has state 0. For instance, a tile-association reaction for the addition of a tile on the east of the assembly was implemented by the local rule,

```
Tile(E1,in~1,act~none) + Tile(W1,in~0,act~none) -> Tile(
    E1!1,in~1,act~ckglue).Tile(W1!1,in~1,act~ckglue) kon.
```

This rule specifies that a free tile binds with a tile with an unbound eastern link inside the assembly at its western edge. Furthermore, the `in` site of the free floating tile has been updated to 1 to indicate it is now part of the assembly. Since tiles are added to the assembly at an equal rate independent of the tile type, the reaction proceeds with a reaction rate `kon`, a constant.

Naturally, we expect three more analogous rules for tile associations on free west, north and south edges. However, such a specification for north-south interactions would allow two different tiles to fill one concave corner slot. Thus, only east-west bindings are of the above form. North-south bindings are conditioned on the case that there are no tiles in the east and west sides of the slot. Moreover, a reaction rule ensures that a tile placed on a corner slot binds with its other neighbor (that is, if the tile was bound east/west, this rule binds it to north/south neighbour; and vice versa). In a sense, this binding establishes the slot solely for the newly added tile. The new binding may be between matching

---

other sites, from our implementation in NFsim we saw that such cases may generate errors in the numerical simulation. Thus, we use sites  $N1, E1, S1$ , and  $W1$  for implementing the physical bond, and sites  $N_g, E_g, S_g$ , and  $W_g$  for capturing the glue state of the corresponding edge.

glues - requiring a strength update of the interacting tiles. Thus, the `act` site of these tiles changes its state to `ckglue`.

In the association reaction rule and slot establishing reaction, the `act` site was changed from state `none` to `ckglue`. When this state change occurs, a set of reactions which check whether attached tiles have matching glues become activated. For instance, the reaction rule,

```
Tile (E1!1, Eg~1, Em~0, act~ckglue) . Tile (W1!1, Wg~1, Wm~0, act~
ckglue) -> Tile (E1!1, Eg~1, Em~1, act~inc2) . Tile (W1!1, Wg
~1, Wm~1, act~inc2) kmax,
```

declares that if the linked tiles both have a strength-2 glue (`Eg~1` and `Wg~1`), the summed strength of their matching glues needs to be incremented by two (`act~inc2`). For matching glues of strength 1, `act` site will be updated to state `inc1`. The actual incrementation is done via a rule,

```
Tile (bnd~0, act~inc2) -> Tile (bnd~2, act~none) kmax,
```

for the case where the tile initially had no bond strength and the increment size is 2. Similarly, rules are defined for checking strength-1 matching glues and for size one increments. The matching-glue checking reactions also updated the state of the `Em/Wm` sites to 1 enabling the identification of correct bonds. Finally, both reactions proceed at the maximum rate, occurring instantaneously so as not to affect the dynamics of `kTAM`, until all tiles which have participated in a binding (either due to initial tile attachment or due to slot establishment) have their binding strength incremented.

On the other hand, tile dissociation was achieved by a combination of a set of reactions. First, tiles in the assembly are marked to be deleted by rules of the form,

```
Tile (in~1, bnd~2, corner~0, act~none) -> Tile (in~1, bnd~2,
corner~0, act~del) kTonReal/exp(2*Gse)
```

at a rate determined by their total bond strength. Next, the bonds of the marked tiles are broken by rules of the form,

```
Tile (N1!1, in~1, act~none) . Tile (S1!1, in~1, act~del) -> Tile (
N1, in~1, act~none) + Tile (S1, in~1, act~del) kmax
```

When the tiles have detached, their former neighbors may need to update their bond strength if their glue was labeled as matching. Hence, local rules of the form,

```
Tile (E1, Eg~1, Em~1, act~none) -> Tile (E1, Eg~1, Em~0, act~dec2
) kmax.
```

signal the need to decrement the total bond strength of the tile. Finally, the bond strength is decremented using rules of the form,

```
Tile (bnd~2, act~dec2) -> Tile (bnd~0, act~none) kmax
```

*Note on implementation of instantaneous reaction rules*

One of the key techniques used in our modelling approach is the employment of instantaneous reaction rules, such as the reaction incrementing and decrementing the total bond strength or the set of reactions checking whether attached tiles have matching glues. Because NFsim does not natively have incorporated within such mechanism of implementing an un-interrupted sequential set of reactions without advancing the model time, a modelling artifice has to be employed. Namely such reactions are defined as having a very high reaction rate,  $kmax$  in comparison to all other reactions, such that these reactions are chosen preferentially, while their sequentiality is enforced by adding specific states to some of the sites; such is for example the site *act* of our *Tile* agent. Also, the high reaction rate enforces a very small, arguably insignificant, advancement of the model time, for running this reactions. Such modeling artifice is not new for the BNGL modelling community, see for example [6, 7, 36].

In the case of our systems, it has been observed experimentally that assigning  $kmax$  to  $10^6$  ensures such an instantaneous application of these rules, as this rate has a  $10^2$ -fold increase from the next (fast) model reaction (namely the detachment of those tiles which are bound to the assembly by a bond of strength 0).