# On Turedo Hierarchies and Intrinsic Universality

- Samuel Nalin 2
- Univ. Orléans, INSA Centre Val de Loire, LIFO EA 4022, Orléans, France 3
- **Guillaume Theyssier** 4
- I2M, CNRS, Université Aix-Marseille, France 5

### Abstract -

6

This paper is about turedos, which are Turing machine whose head can move in the plane (or in a 7 higher-dimensional space) but only in a self-avoiding way, by putting marks (letters) on visited positions and moving only to unmarked, therefore unvisited, positions. The key parameter of turedos is their lookup radius: the distance up to which the head can look around in order to make 10 its decision of where to move to and what mark to write. In this paper we study the hierarchy of 11 turedos according to their lookup radius and the dimension of space using notions of simulation up 12 to spatio-temporal rescaling (a standard approach in cellular automata or self-assembly systems). 13 We establish that there is a rich interplay between the turedo parameters and the notion of 14 simulation considered. We show in particular, for the most liberal simulations, the existence of 3D 15 turedos of radius 1 that are intrinsically universal for all radii, but that this is impossible in 16 dimension 2, where some radius 2 turedo are impossible to simulate at radius 1. Using stricter 17 notions of simulation, intrinsic universality becomes impossible, even in dimension 3, and there is a 18 strict radius hierarchy. Finally, when restricting to radius 1, universality is again possible in 19 dimension 3, but not in dimension 2, where we show however that a radius 3 turedo can simulate all 20

radius 1 turedos. 21

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Models of computation; Theory of 22

computation  $\rightarrow$  Formal languages and automata theory 23

Keywords and phrases Turedos; intrinsic universality; Higher-dimensional Turing machines; 24 Oritatami systems 25

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23 26

#### 23:2 On Turedo Hierarchies and Intrinsic Universality

# 27 **1** Introduction

The field of biomolecular computing has given rise to several theoretical models that 28 describe growing process of (molecular) assemblies governed by local interaction or 29 gluing rules. One of the most studied one, the abstract Tile Assembly model (aTAM) [19], 30 describes a process where the growth can happen anywhere asynchronously. It was 31 successfully implemented in vitro as DNA self-assembly [18]. In oritatami systems, 32 introduced more recently [9, 8] and inspired from RNA origami [11, 7], the growth 33 happens at a unique given point of the assembly and in a sequential manner. Despite their 34 obvious differences as models of biomolecular systems, they share common features as 35 computational models. Both are limited by the fact that they can only grow shapes (and 36 not erase them), but both were shown capable of embedding universal computations in 37 different ways [10, 12, 3, 14, 19]. Recently, it was also shown through new results on 38 oritatami systems that the infinite limit shapes both models can generate from finite seeds 39 have the same computational complexity and the same possible densities of occupied 40 position in the plane [17]. The key ingredients of these new results were, on one hand, the 41 introduction of a new model, called *turedos*, which abstracts away low level details of 42 oritatami systems and is easier to program, and on the other hand, a proof that oritatami 43 systems can simulate a large family of turedos. This underlines the interest of turedos but 44 also the surprising capabilities of models of sequential growth process in the plane. The 45 present work is entirely focused on the turedo model and aims at better understanding its 46 limitations and its expressive power as a growth process and computational model. 47

Turedos. Intuitively, a turedo is a Turing machine whose head can move in the plane 48 or in the 3-dimensional space but only in a self-avoiding way, *i.e.* without going back to 49 a previously visited position. More precisely, the turedo's head can only move to empty 50 or unmarked positions, and it must put a mark on each visited position when leaving it. 51 The key ingredient of a turedo, and what makes its main computational power, is its *lookup* 52 radius: when deciding a move and what mark to put on the visited position, the turedo has 53 access to the local configuration of marks around its position, up to some finite distance. 54 Like any Turing machine, a turedo also has a set of internal head states. Without entering 55 into details, an oritatami system consists of a "molecule" made of "beads" that can attract 56 each other. The molecule grows at each step following a periodic bead sequence and folds 57 as follows: the  $\delta$  most recently produced beads are free to move around to look for the 58 position that maximizes the number of bonds they can make with each other ; then the first 59 (oldest) bead among the  $\delta$  most recent ones is fixed according to that position, a new free 60 bead is added and the process iterates. This behavior can be realized in a turedo of radius 61  $\delta + 1$ . The main result of [17] is that oritatami of delay 3 can simulate turedos of radius 62 1. In fact, turedos also naturally capture variants of oritatami systems: for instance, one 63 can imagine negative (repulsive) bonds in the process of maximizing gluing strength, or 64 add local rules forbidding two bead type to be neighbor of each other. Therefore negative 65 results on turedos become negative results on oritatami systems, but also potential variants 66 of them. 67

**Simulations and universality.** The main question we address here is how the capabilities of turedos change with their radius and what is the role of the dimension of space. We are interested in qualitative differences and don't compare turedos move by move and cell by cell. We rather use a notion of simulation allowing spatio-temporal rescaling, similar to the one used in cellular automata [4, 1], in aTAM [6, 5, 16] or in the simulation result of turedo by oritatamis [17]. Intuitively a cell can be simulated by a block

of cells, and a step by a finite number of steps. We are naturally interested by hierarchy 74 results (existence of turedos that can't be simulated by lower radius ones), or on the 75 contrary by universality results (existence of turedos that can simulate all turedos or a 76 77 large class of them). For instance, the existence of intrinsically universal systems in the aTAM model was much studied and it was shown that it crucially depends on natural 78 parameters of the model [6, 16, 5, 13]. However, as natural as it might seem, a formal 79 notion of simulation is never a neutral choice when tackling these questions [1, 4, 2]. Thus, 80 in addition to the parameters of the turedo model we also study the influence of the 81 notion of simulation itself. For this we identify various ingredients, in particular the 82 possibility of fuzz, *i.e.* the tolerance allowed for the simulating turedo to visit some 83 regions of space close to the ongoing assembly that represent empty cells of the simulated 84 turedo and are therefore not yet been visited by it. Note that a similar notion of fuzz 85 appeared in the intrinsic universality results on aTAM [5]. We end up with three notions 86 of simulation: the rigorous, the fuzzless and the liberal ones. 87

Our contributions. After formalizing all the concepts mentioned so far (Section 2), we establish a surprisingly diverse general picture of the capabilities of turedos of simulating each other. Separating the negative results (Section 3) from the positive ones (Section 4), our results are the following:

- under fuzzless simulation, intrinsic universality is impossible whatever the dimension,
   there is a radius hierarchy, and actually the impossibility strikes at radius 2: no turedo
   can fuzzlessly simulate all radius 2 turedos (Theorem 5);
- when restricting to radius 1, rigorous intrinsic universality is possible in dimension 3 (Theorem 11), but not in dimension 2 (Theorem 6);
- 3. however, we built a 2D turedo of radius 3 which is able to rigorously simulate all 2D turedo of radius 1 (Theorem 10);
- 4. for liberal simulations, we establish intrinsic universality in dimension 3 and a complete
   hierarchy collapse at radius 1 (Corollary 13);
- finally we show that there is a 2D turedo of radius 2 which is impossible to simulate at radius 1, even under liberal simulations (Theorem 9).

Besides the above results, our contribution also lies in the constructions and proof 103 techniques. For instance, the negative result 5 is based on a general lemma for 2D turedos 104 of radius 1, which bounds the quantity of information (using Kolmogorov complexity) 105 that can be carried from the seed to another connected component of the plane when the 106 plane is divided by a 4-connected path. As another example on the constructive side, 107 result number 3 above uses a novel construction technique (called heat sink trick) that 108 fully exploits the potential of radius 3 and could be used in any place where two 109 unbounded streams of information have to be crossed. Finally, in Section 5, we discuss 110 various questions left open and present what we believe are promising future research 111 directions. 112

# 113 **2** Definitions

We denote by  $\mathbb{N}$  the set of natural numbers (including 0), by  $\mathbb{N}_+$  the positive ones, and by  $\mathbb{Z}$ the set of integers. We consider turedos on  $\mathbb{Z}^d$  for d = 2 or 3 (and in particular, we don't use the hexagonal lattice on the plane, mostly to simplify notations and dimension change). We fix a blank symbol  $\perp$  used to represent empty positions and common to all turedos. The ball of radius r in dimension d, denoted  $B_d(r)$ , is the set of positions reachable in r elementary moves (moves along vector of the canonical based of  $\mathbb{Z}^d$ ) from the origin.  $B_d(1)$  will always

#### 23:4 On Turedo Hierarchies and Intrinsic Universality

be the set of possible head moves in dimension d. We denote by c[z; S] the pattern of shape S around position z in configuration c, *i.e.* the map  $z' \in S \mapsto c_{z+z'}$ .

▶ **Definition 1** (Turedo). A turedo of dimension d and radius r is a triple  $T = (A, Q, \delta)$  where A is its finite alphabet with  $\bot \in A$ , Q is its finite set of head states and

$$\delta: Q \times A^{B_d(r)} \to Q \times A \setminus \{\bot\} \times B_d(1)$$

is its local transition map. A global state of T is a triple  $(c, z, q) \in \mathcal{G}_T = A^{\mathbb{Z}^d} \times \mathbb{Z}^d \times Q$ specifying a configuration, a position and a head state. The global transition map  $F_T : \mathcal{G}_T \to \mathcal{G}_T$ associated to T is defined by:

$$F_{T}(c, z, q) = \begin{cases} (c, z, q) & \text{if } c(z) \neq \bot \text{ or } c(z + \mu) \neq \bot \\ (c', z + \mu, q') & \text{else,} \end{cases}$$

where  $(q', a, \mu) = \delta(q, c[z; B_d(r)])$  and configuration c' is defined by c'(z) = a and c'(z') = c(z')for all  $z' \neq z$ .

The *domain* of a global state (c, z, q) is the set of non-blank positions of c plus the head position, formally:  $\mathcal{D}(c, z, q) = \{z\} \cup \bigcup \{z' : c(z') \neq \bot\}$  A global state (c, z, q) is finite if its domain is finite. We are interested in orbits starting from finite initial global states, called finite seeds.

▶ **Example 2** (The spiral-XOR turedo). Let  $A = \{\bot, 0, 1\}$  and  $Q = \{\leftarrow, \uparrow, \rightarrow, \downarrow\}$ . The spiral-135 XOR turedo has the following local rule. The head holds a direction  $d \in Q$  and tries to 136 move in that direction and let behind, as letter of A, the sum modulo 2 of the states of 137 neighboring (non  $\perp$ ) positions. When it can do the d move, it changes its internal state 138 (counter-clockwise), when it can't it takes the first available move (clockwise) and doesn't 139 change its state. Of course if there is no neighbor in state  $\perp$  then the turedo is blocked. The 140 following table shows the local transition map up to rotation of d (the red arrow can be 141 rotated and all blue arrows are defined relatively to the red arrow): 142

state	1st empty neighbor	new state	letter	move
1	<b>↑ (+0)</b>	← (-1)	$\sum mod2$	<b>↑ (+0)</b>
1	$\rightarrow$ (+1)	<b>↑ (+0)</b>	$\sum mod2$	$\rightarrow$ (+1)
1	↓ (+2)	<b>↑ (+0)</b>	$\sum mod2$	↓ (+2)
1	← (+3)	<b>↑ (+0)</b>	$\sum mod2$	← (+3)

143

<sup>144</sup> See Figure 1 for an example of orbit.

The main focus of this paper is to understand the role of radius and dimension in the computational complexity of turedos. We will denote by  $TUR_d(r)$  the set of turedos of dimension *d* and radius *r*, and  $TUR_d = \bigcup_{r>1} TUR_d(r)$ .

Before formalizing simulation, we first need to define *block encodings* which are ways to represent global states of a simulated turedo by blocks in the simulator. Any given  $b \in \mathbb{N}^d_+$ defines a rectangular block  $R_b = \{z \in \mathbb{N}^d : 0 \le z_i < b_i \text{ for } 1 \le i \le d\}$  and  $Z^d$  can be tiled by translated copies of  $R_b$  in a regular way by placing them on the sublattice  $b \otimes \mathbb{Z}^d$  where  $\otimes$ denotes the component-wise product. Each position  $z \in \mathbb{Z}^d$  can be uniquely decomposed into  $z = \rho_b(z) + \mu_b(z)$  where  $\rho_b(z) \in b \otimes \mathbb{Z}^d$  is the reference point of a block and  $\mu_b(z) \in R_b$ is an offset inside it. Initial global state Global state after 10 steps Global state after 307 steps **Global state after 307 steps** 

**Figure 1** Example of orbit of the spiral-XOR turedo (example 2) starting from a finite seed. Green represents 0, yellow represents 1 and white represents  $\perp$ . The head holds a direction (where the black triangle is pointing to) and its self-avoiding trajectory since the beginning is drawn as a black path.

▶ **Definition 3** (Block encoding). Let us fix a dimension d. Given two pairs of alphabets and state sets  $(A_1, Q_1)$  and  $(A_2, Q_2)$  with  $\bot \in A_1 \cap A_2$ , a block encoding of global states  $\mathcal{G}_1 = A_1^{\mathbb{Z}^d} \times \mathbb{Z}^d \times Q_1$  into  $\mathcal{G}_2 = A_2^{\mathbb{Z}^d} \times \mathbb{Z}^d \times Q_2$  is given by a block size  $b \in \mathbb{N}^d_+$  and two partial onto maps:

the headless block decoding map  $\alpha: D_{\alpha} \subseteq A_2^{R_b} \to A_1$  verifying  $\perp^{R_b} \in D_{\alpha}$  and  $\alpha(\perp^{R_b}) = \perp$ ,

161 *the* head block decoding map  $\beta : D_{\beta} \subseteq A_2^{R_b} \times R_b \times Q_2 \rightarrow Q_1 \times A_1$ .

<sup>162</sup> A global state  $(c, z, q) \in \mathcal{G}_2$  is valid for the encoding if it is made only of patterns from  $D_{\alpha}$  far from <sup>163</sup> the head and  $D_{\beta}$  around the head, precisely if:  $(c[\rho_b(z); R_b], \mu_b(z), q) \in D_{\beta}$  and  $c[\rho_b(z'); R_b] \in D_{\alpha}$ <sup>164</sup> for all  $z' \in \mathbb{Z}^d$  such that  $\rho_b(z') \neq \rho_b(z)$ .

Finally, the global decoding map  $\Gamma$  associates to any valid global state  $(c_2, z_2, q_2) \in \mathcal{G}_2$  a global state  $(c_1, z_1, q_1) \in \mathcal{G}_1$  defined by application of decoding maps  $\alpha$  or  $\beta$  on each block according to the presence of the head in the block, i.e. :

168 **b**  $\otimes z_1 = \rho_b(z_2)$ ,

169  $(q_1, c_1(z_1)) = \beta(c_2[\rho_b(z_1); R_b], \mu_b(z_2), q_2),$ 

170 •  $c_1(z) = \alpha(c_2[\rho_b(z_2); R_b])$  for all  $z \neq z_1$ .

The map  $\alpha$  and  $\beta$  being partial and onto intuitively means that not all global states are valid, and that any global state of  $A_1^{\mathbb{Z}^d} \times \mathbb{Z}^d \times Q_1$  can be encoded. Note that the headless block decoding map  $\alpha$  always decodes blank blocks  $\perp^{R_b}$  as blank state  $\perp$ . Denote by  $\mathcal{D}_b(c_2, z_2, q_2)$  the *block domain* of global state  $(c_2, z_2, q_2)$  which is the set of blocks that are not entirely blank, *i.e.*  $\mathcal{D}_b(c_2, z_2, q_2) = \{z : b \otimes z = \rho_b(z_2) \text{ or } c_2[b \otimes z; R_b] \neq \perp^{R_b}\}.$ 

We can now define simulations precisely using block encodings. Intuitively, we ask for the simulator to be able to reproduce any orbit of the simulated turedo starting from a finite seed, and using a (fixed) finite number of steps to simulate one step. Since we want the initial seed of the simulator to be neutral and without any pre-computed information about the future of the simulated orbit, we ask that its block domain correspond to the domain of the simulated seed.

**Definition 4** (Simulation). Let d be a fixed dimension. We say that a d-dimensional turedo  $T_2$ simulates a d-dimensional turedo  $T_1$  if there is:

<sup>184</sup> = a block encoding of  $\mathcal{G}_{T_1}$  into  $\mathcal{G}_{T_2}$  of bock size b and global decoding map  $\Gamma$ ,

185 a time scaling factor  $k \in \mathbb{N}_+$ ,

such that for each finite global state  $(c_1, z_1, q_1) \in \mathcal{G}_{T_1}$  and each global state  $(c_2, z_2, q_2) \in \mathcal{G}_{T_2}$ verifying:

- 188 corresponding block domain:  $D(c_1, z_1, q_1) = D_b(c_2, z_2, q_2)$ ,
- 189 *correct encoding:*  $(c_1, z_1, q_1) = \Gamma(c_2, z_2, q_2)$ ,



**Figure 2** Differences in allowed head movements in rigorous, fuzzless and liberal simulations with  $2 \times 2$  blocks. The colors have the following meaning: in red the positions or blocks which are not empty initially, in yellow the positions or blocks coding a non- $\perp$  letter during the orbit; in white the positions or blocks coding  $\perp$ ; in black the movement of the head.

then it holds  $\forall t \in \mathbb{N}, F_{T_1}^t(c_1, z_1, q_1) = \Gamma(F_{T_2}^{kt}(c_2, z_2, q_2))$ . Such simulations are the base upon which we define three variants (two restrictions and one generalization).

We say that a simulation is fuzzless if the block domain in the simulator orbit remains identical to the domain of the simulated orbit, precisely:  $\forall t \in \mathbb{N} : \mathcal{D}_b(F_{T_2}^{kt}(c_2, z_2, q_2)) = \mathcal{D}(F_{T_1}^t(c_1, z_1, q_1)).$ 

We say that a simulation is rigorous if the movements of the head of  $T_2$  in simulating orbits strictly remains inside blocks corresponding to the simulated head position of  $T_1$ , even at intermediate steps, precisely: if  $z_1^t$  denotes the head position of  $F_{T_1}^t(c_1, z_1, q_1)$  and  $z_2^t$  that of  $F_{T_2}^t(c_2, z_2, q_2)$ , it holds for all t' with  $kt \le t' \le k(t+1)$ :  $z_2^{t'} \in (b \otimes z_1^t + R_b) \cup (b \otimes z_1^{t+1} + R_b)$ . Finally, a liberal simulation is a generalized simulation where we only ask that for each finite global state  $(c_1, z_1, q_1) \in \mathcal{G}_{T_1}$  there exists a global state  $(c_2, z_2, q_2) \in \mathcal{G}_{T_2}$  with corresponding block domain and correct encoding such that it holds  $\forall t \in \mathbb{N}, F_{T_1}^t(c_1, z_1, q_1) = \Gamma(F_{T_2}^{kt}(c_2, z_2, q_2))$ .

Liberal simulations can have fuzz and make non-rigorous head movements. Note that 201 fuzzless simulations are equivalent to simulations where the headless block decoding map 202 is such that  $\alpha(u) = \bot \iff u = \bot^{R_b}$ , *i.e.* that the only block coding  $\bot$  is  $\bot^{R_b}$ . Note also 203 that a rigorous simulation is necessarily fuzzless because the head of the simulator has 204 no opportunity, even at intermediate time steps, to visit blocks not corresponding to the 205 domain of the simulated configuration. The power of fuzzless simulations compared to 206 rigorous ones is to allow the head to go back to blocks that were previously visited. Thus 207 the head can potentially retrieve information from non adjacent blocks that were written 208 a long time ago. With rigorous simulation on the contrary, the head has only access to 209 adjacent blocks during a simulation cycle. 210

We denote by  $\leq$  the liberal simulation, by  $\leq_{FL}$  the fuzzless simulation and by  $\leq_R$  the rigorous simulation. Among many properties of these simulation relations, we are particularly interested in universality: the capacity of a single turedo to simulate a whole set of turedos. Given a dimension d and a radius r, we denote by  $\mathcal{U}_d^{\leq}(r)$  the set of turedos  $T \in \text{TUR}_d$  such that for any  $T' \in \text{TUR}_d(r)$  it holds  $T' \leq T$ . We denote by  $\mathcal{U}_d^{\leq}$  the set of turedos  $T \in \text{TUR}_d$  such that for any  $T' \in \text{TUR}_d$  it holds  $T' \leq T$ . We use similar notations for simulation relations  $\leq_R$  and  $\leq_{FL}$ .

# 218 **3** Separation Results

# 219 3.1 No Fuzz, no Fun

The fuzzless condition gives much importance to larger radii, simply because a turedo's head surrounded by blocks coding  $\perp$  cannot read information far away without moving inside these blocks and thus creating fuzz. The following theorem exploits this obvious



**Figure 3** Example of a copy-and-move operation (on the left) and its rigorous simulation by a turedo of radius 1 with  $3 \times 3$  block size (on the right). The color convention is as follows: in red the letters present in the seed, in dark yellow the initial position of the head, and in light yellow, the positions visited by the head during the orbit. The only position on the south side of the middle block that can depend on *a* is the lower right corner, marked with a *X*.

limitation to show two immediate consequences: first, there is a radius hierarchy (new behaviors appear at radius r + 1 that cannot be simulated at radius r) and thus no general fuzzless universality; second, universality is impossible even at radius 2: any turedo (whatever its radius) will fail to simulate some radius-2 turedo. The dimension plays no role in these results.

**Theorem 5.** For any  $d \ge 2$  and  $r \ge 1$ , we have the following:

there is  $T_{r+1} \in \text{TUR}_d(r+1)$  such that for all  $T_r \in \text{TUR}_d(r)$ ,  $T_{r+1} \not\leq_{FL} T_r$ ; in particular,  $\mathcal{U}_d^{\leq_{FL}} = \emptyset$ .

for any  $T_r \in \text{TUR}_d(r)$  there exists  $T_2 \in \text{TUR}_d(2)$  such that  $T_2 \not\leq_{FL} T_r$ ; in particular,  $\mathcal{U}_d^{\leq_{FL}}(r) = \emptyset$  for any  $r \geq 2$ .

# **3.2** Dimension 2 and Radius 1: the Jordan Curve Burden

A turedo's head in dimension 2 always moves drawing a 4-connected path. When the turedo has radius 1, it has no way to read information across such a path (while it could with a larger radius). Therefore head movements for turedos of radius 1 turns into potential information barriers. The precise way in which this simple observation affects the simulation power of such turedos depends on the type of simulation considered.

Let us first consider rigorous simulations. Any turedo T (whatever its radius) can obviously do the following elementary copy-and-move operation (see Figure 3):

- move to the right to some position z;
- read the letter *a* present at position z + (1, 0);
- then move to position z + (0, 1) and leave behind letter *a* at position *z*.

In particular, if the head continues its way and later arrives at position z - (0, 1) from the south, it can read the information *a* copied at position *z*.

However, if we suppose that some  $T_1 \in \text{TUR}_2(1)$  simulates T under rigorous simulations with block size b, the movement of its head inside block  $b \otimes z$  to simulate the above copy-and-move step must be the following (see Figure 3):

coming from the left side of the block, it draws some path inside it until it reaches the
 right border (if not it cannot read any information from the adjacent block to the right);

then it must move north, otherwise it would be trapped in the south part of the blocks

by the 4-connected path drawn so far that connects the left and right sides of the block;
it must finally escape through the north side.

This head movement is such that at most 1 letter of  $T_1$  is written on the south side of block  $b \otimes z$  after having had the opportunity to read some information from the adjacent block that encodes letter *a*. In particular, if  $T_1$  has smaller alphabet than *T* this is not enough to completely encode *a* on the south side of block  $b \otimes z$ . So, this is a limitation that has to be dealt with if later in the simulation the head arrives from the south and has to read from

#### 23:8 On Turedo Hierarchies and Intrinsic Universality



**Figure 4** Behavior of turedo T' on the seed  $\sigma(\vec{a}, 2, a)$ . The red and blue colors indicate letters present in the seed. The dark yellow color indicates the initial position of the head and light yellow cells represent the path of the head until the last step of the orbit. The orange cell correspond to the position of the head before moving left or right according to the result of the test  $a \stackrel{?}{=} a_2$ .

the south side of block  $b \otimes z$ . A single copy-and-move is not enough to get a contradiction because the simulation of  $T_1$  could be organized so as to transport the complete information about *a* along the way and have it on hands already when arriving at the south of block  $b \otimes z$ . However, by repeating such copy-and-move steps, one can saturate the simulator and show the following theorem that states that there is no universal turedo of radius 1 among turedos of radius 1 for rigorous simulations.

▶ **Theorem 6.** For any  $T \in \text{TUR}_2(1)$  there is  $T' \in \text{TUR}_2(1)$  such that  $T' \not\leq_R T$ . In particular  $\mathcal{U}_2^{\leq_R}(1) \cap \text{TUR}_2(1) = \emptyset$ .

**Proof.** Let *Q* be the state set of *T*, *A* be the alphabet of *T* and consider any alphabet  $A_+$ with  $m = |A| < |A_+| = m_+$ . Then it is straightforward to construct a turedo  $T' \in \text{TUR}_2(1)$ of alphabet  $A' = A_+ \cup \{\downarrow, \leftarrow, \uparrow\}$  that has the following behavior (see Figure 4):

for any  $n \in \mathbb{N}$ , any  $\vec{a} = (a_0, \ldots, a_n) \in A^{n+1}_+$ ,  $a' \in A_+$  and  $0 \le i \le n$ , consider the finite seed  $\sigma(\vec{a}, i, a')$  with head in position (0, 0),  $a_j$  in position (3j + 2, 0) for  $0 \le j \le n$ , a'in position (3i + 1, -3) and  $\downarrow$  in positions (3(n + 1) + 2, 0) and (3(n + 1) + 2, -1),  $\leftarrow$  in position (3(n + 1) + 2, -2) and finally  $\uparrow$  at position (3i, -2);

from such a seed, T' starts a sequence of n + 1 copy-and-move steps that results in having a copy of  $a_j$  at position (3j + 1, 0) for  $0 \le j \le n$ ; the end of this phase occurs at time step 5(n + 1) and the head reaches position (3(n + 1), 0);

 $_{277}$  = then T' reaches the first  $\downarrow$  and follows the move indications of arrows (down, down,

left), until it reaches the up arrow, and moves from position (3i + 1, -2) to (3i + 1, -1); finally, at position (3i + 1, -1) it moves right if  $a_i = a'$  and left otherwise (it can do so

because it has copied the value of a' when leaving position (3i + 1, -2)).

Let's call  $t_{n,i}$  the time step at which occurs this final left or right move  $(t_{n,i}$  only depends 281 on *i* and *n*): at time  $t_{n,i}$ , the head of T' must be either at position (3i, -1) or (3i + 2, -1). 282 Thus T' implements on seed  $\sigma(\vec{a}, i, a')$  the test of whether  $a_i = a'$ . We are going to show 283 that T cannot simulate T' under rigorous simulations. Suppose by contradiction that 284  $T' \leq_R T$  with block size b and time scaling factor k. Given  $n \in \mathbb{N}$  and  $0 \leq i \leq n$ , denote by 285  $A_n \subseteq \mathbb{Z}^2$  the set of positions that are on the right side of block  $b \otimes (3(n+1), 0)$  (the block 286 corresponding to the position reached by T' at the end of the copy-and-move sequence as 287 detailed above). Denote by  $B_{i,n} \subseteq \mathbb{Z}^2$  the set of positions that are on the south side of 288 block  $b \otimes (3i + 1, 0)$ . Finally, denote by  $C_{i,n} \subseteq \mathbb{Z}^2$  the set of positions made of the union of 289 blocks (3i+1,-3), (3i,-2), (3(n+1)+2,0), (3(n+1)+2,-1), (3(n+1)+2,-2) (i.e. 290 those corresponding to position a' or an arrow  $\{\downarrow, \leftarrow, \uparrow\}$  in the seed  $\sigma(\vec{a}, i, a')$ ). Consider 291 now  $n \in \mathbb{N}$ ,  $\vec{a}, \vec{c} \in A_+^{n+1}$   $a' \in A_+$  and  $0 \le i \le n$ , and take any two global states  $g_1$  and  $g_2$  of 292 T that correctly simulate the orbits of T' on seed  $\sigma(\vec{a}, i, a')$  and  $\sigma(\vec{c}, i, a')$  respectively and 293

that are identical on  $C_{i,n}$  Considering time step  $t_0(n) = 5k(n+1)$  corresponding to the 294 end of the copy-and-move sequence, if global states  $T^{t_0(n)}(g_1)$  and  $T^{t_0(n)}(g_2)$  are identical 295 on domains  $A_n$  and  $B_{i,n}$  and have the same head state, then both orbits must make the 296 same final decision to move to the left block or the right block at the final time step  $kt_{n,i}$ , 297 precisely: the head in global state  $T^{kt_{n,i}}(g_1)$  is in the same block as the head in global state 298  $T^{kt_{n,i}}(q_2)$  (and it must be either  $b \otimes (3i, -1)$  or  $b \otimes (3i+2, -1)$ ). Indeed, by the property of 299 rigorous simulations and the behavior of T', the only positions with content written before 300  $t_0(n)$  that the head of T can possibly read between time step  $t_0(n)$  and  $kt_{n,i}$  are positions 301 in  $A_n \cup B_{i,n} \cup C_{i,n}$ , so the orbit starting from step  $t_0(n)$  is completely determined by the 302 content of the configuration in that domain and the internal state of T at time  $t_0(n)$ . 303

<sup>304</sup>  $\triangleright$  Claim 7. There must exist  $n \in \mathbb{N}$ ,  $0 \le i \le n$ ,  $\vec{a} \in A_+^{n+1}$ ,  $a' \in A_+$  and  $\vec{c} \in A_+^{n+1}$  with  $a_j = c_j$ <sup>305</sup> for all j < i and  $a_i \ne c_i$ , and two global states  $g_1$  and  $g_2$  of T that correctly simulate seeds <sup>306</sup>  $\sigma(\vec{a}, i, a')$  and  $\sigma(\vec{c}, i, a')$  respectively, and also such that  $T^{t_0(n)}(g_1)$  and  $T^{t_0(n)}(g_2)$  have same <sup>307</sup> head state and are identical on domain  $A_n \cup B_{i,n} \cup C_{i,n}$ .

**Proof of the claim.** In this proof, we fix for each  $a \in A_+$  a unique block of  $A^{R_b}$  that encodes it, and for any seed of T' we only consider a unique global state of T that simulates it. First, there are only a bounded number (bound in n) of possible content of a configuration on domain  $A_n$  and state of T, so for each n there must exist  $u \in A^{A_n}$  and  $q \in Q$ , a set  $X_n \subseteq A_+^n$  of size  $\Omega(m_+^n)$  such that for each  $0 \le i \le n$  and each  $\vec{a} \in X_n$ , the corresponding global state g of T simulating T' on seed  $\sigma(\vec{a}, i, n)$ , is such that  $T^{t_0(n)}(g)$  is equal to u on domain  $A_n$  and with head state q.

Second, we claim that for large enough n there must be some i and a prefix  $a_0, \ldots, a_{i-1} \in A^i_+$  such that there are at least m + 1 choices of  $a_i \in A_+$  such that  $a_0, \ldots, a_i$  $a_{17}$  can be completed into an element  $\vec{a} \in X_n$ . Indeed, otherwise we would have  $|X| \leq m^n$ which would contradict the fact that  $|X| \in \Omega(m^n_+)$  for large enough n since  $m < m_+$ .

Now consider the set of global states that simulates the seeds  $\sigma(\vec{a}, i, a')$  where  $\vec{a} \in X_n$ 319 are the m + 1 completed vectors from the common prefix  $a_0, \ldots, a_{i-1}$ , and  $a' \in A_+$ . They 320 are identical on the blocks corresponding to the common prefix  $a_0, \ldots, a_{i-1}$  of the seed 321 they simulate. As already said, these global states at step  $t_0(n)$  are also identical on domain 322  $A_n$  and have same head state. Moreover, on domain  $B_{i,n}$  and still at step  $t_0(n)$ , they agree 323 because of the common prefix  $a_0, \ldots, a_{i-1}$ , except possibly on the lower-right corner where 324 they can take at most m different values (see Figure 3 and discussion at the beginning of 325 this section). We deduce that among the m + 1 choices for  $a_i$ , at least 2 must correspond to global states that completely agree on  $B_{i,n}$ . Denote by a' and c' these two choices and 327 consider  $\vec{a}$  and  $\vec{c}$  to be the vectors completing the prefixes  $a_0, \ldots, a_{i-1}, a'$  and  $a_0, \ldots, a_{i-1}, c'$ 328 respectively. The claim follows by choosing seeds  $\sigma(\vec{a}, i, a')$  and  $\sigma(\vec{c}, i, a')$ . 329

The theorem follows from the claim by contradiction: as shown above, global states  $g_1$  and  $g_2$  force the same behavior of T starting from time  $t_0(n)$ , but at the same time their orbits should not end up in the same block because they simulate seeds of T' that do not have the same answer to the final equality test.

We will now establish a strong separation between  $TUR_2(1)$  and  $TUR_2(2)$  even under liberal simulations. We first establish a lemma expressing bounds on information leakage between too regions separated by a 4-connected path. It is formulated using Kolmogorov complexity. Recall that the (plain) Kolmogorov complexity of a string  $u \in \{0, 1\}^*$  is the length of the shortest program that outputs u, more precisely the length of the shortest  $v \in \{0, 1\}^*$  such that a suitable fixed universal Turing machine outputs u on input v (see

### 23:10 On Turedo Hierarchies and Intrinsic Universality



**Figure 5** Orbit of turedo  $T_2$  starting from a seed u of length n (in red) with the head initially in the position shown in dark yellow. The last part of the orbit is shown in orange.

[15]). For any  $X \subseteq \mathbb{Z}^2$  and any (partial) configuration  $c \in Q^X$  of finite domain, we denote by K(c) its Kolmogorov complexity, which is the Kolmogorov complexity of the finite binary string u that encodes c as a list of pairs (z, c(z)) such that  $c(z) \neq \bot$  given in lexicographical order.

▶ **Lemma 8.** Let  $C_0 \in \mathbb{N}$  be some constant and  $T \in \text{TUR}_2(1)$ . Then there is another constant  $C \in \mathbb{N}$  with the following property. Consider any 4-connected path  $\rho$  of  $\mathbb{Z}^2$  that divides  $\mathbb{Z}^2$  in 2 or more connected components, and any finite global state  $s \in \mathcal{G}_T$  with head at position (0,0), and whose domain  $\mathcal{D}(s)$  lies entirely in one of the connected components defined by  $\rho$ , denoted  $A_0$ . Suppose moreover that for some  $n \in \mathbb{N}$ , the orbit from global state s to global state  $(c, z, q) = F_T^n(s)$ is such that the head visits path  $\rho$  at most  $C_0$  times. Then, the restriction of c to the complement of  $A_0$  has 'small' kolmogorov complexity:  $K(c_{|\mathbb{Z}^2 \setminus A_0}) \leq C \log(n)$ .

<sup>351</sup> Note that if the seed *s* has 'large' kolmogorov complexity, for instance  $\Omega(n)$ , then *n* steps are <sup>352</sup> far from enough to transmit all the information about *s* to another connected component <sup>353</sup> under the hypothesis of the lemma. The power of this lemma lies in the fact that constant *C* <sup>354</sup> does not depend on the path  $\rho$  nor on the seed *s*. In particular, one can choose  $\rho$  depending <sup>355</sup> on *s* to apply the lemma. Turedos of radius 2 can overcome the limitation of Lemma 8 <sup>356</sup> because they can transmit information over a path without writing on it. It turns out that <sup>357</sup> this is enough to separate TUR<sub>2</sub>(2) from TUR<sub>2</sub>(1) even under liberal simulations.

**Theorem 9.** There is  $T_2 \in \text{TUR}_2(2)$  such that for any  $T_1 \in \text{TUR}_2(1)$ :  $T_2 \not\leq T_1$ .

- Proof. Let's consider the turedo  $T_2 \in \text{TUR}_2(2)$  that behaves as follows on a seed made of a vertical word u of length n (see Figure 5):
- it copies u to the right by making zigzags and does this n times (it implements a unary counter initialized to the length of u while making copies);
- it then moves one cell to the right without making copies (and thus leaving an empty column above);
- it then do again n copies of u by zigzag while moving to the right (note that the first copy can be done because T has radius 2);
- at the end of the last copy it goes around the last bloc of n copies by the north side until it encounters the empty column and then goes down into it until it is blocked.

Denote by  $z_N$  and  $z_S$  the northmost and southmost positions of the last sequence of n south moves of the head (see Figure 5), and by  $t_N$  and  $t_S$  the respective time steps at which the head is at position  $z_N$  and  $z_S$ . Note that  $t_S$  is  $O(n^2)$  and it is the final step of the orbit considered here.

Now suppose by contradiction that there is some  $T_1 \in TUR_2(1)$  such that  $T_2 \leq T_1$ . 373 Denote by k the time rescaling factor and b the block size involved in this simulation. Let's 374 suppose that n is large enough (to be precised later) and that u has large kolmogorov 375 complexity, let's say  $\Omega(n)$ . Consider a global initial state  $s_1$  for  $T_1$  from which starts a 376 correct simulation of the run of  $T_2$ . When the simulation reaches step  $kt_N$ , the head of  $T_1$ 377 is inside bloc  $b \otimes z_N$  and there must be a finite 4-connected path  $p_1, \ldots, p_m$  of empty 378 positions from this position to some position inside bloc  $b \otimes z_S$  because  $T_2$  has to simulate 379 the state changes made by  $T_1$  between steps  $t_N$  and  $t_S$  along the vertical segment of 380 positions from  $z_N$  to  $z_S$ . Let  $\rho$  denote the infinite path that extends  $p_1, \ldots, p_m$  infinitely to 38 the north from  $p_1$  and infinitely to the south from  $p_m$ . 382

We claim that there is a bound  $C_0$  depending only on  $T_1$ , b and k, but not on n and 383 neither on u, such that the run of  $T_1$  starting from global state  $s_1$  until time step  $kt_N$ 384 crosses at most  $C_0$  times path  $\rho$ . First, by choice of  $p_1, \dots, p_m$ , such crossings can only 385 happen at positions of  $\rho$  that are either at the north of  $p_1$  or at the south of  $p_m$ . The 386 simulation is liberal, so the head of  $T_1$  has some freedom of move but it must always 387 remain at a bounded distance from the block corresponding to the simulated head 388 position of  $T_2$  during intermediate steps, precisely: if the head of  $T_2$  is at position z at time step t, then the head of  $T_1$  must be inside block  $b \otimes z$  at time step kt and therefore at 390 distance at most k of block  $b \otimes z$  during time steps between kt and k(t+1). A position is 391 therefore potentially reachable by  $T_1$  before time step  $kt_N$  only if it is at distance at most k 392 from a block  $b \otimes z$  such that position z in the run of  $T_2$  is visited before time step  $t_N$ . The 393 key observation is that in the run of  $T_2$ , there are only finitely many positions that are 394 visited before time  $t_N$  and at distance less than k from either  $z_N$  or any position at the 395 north of it, or from  $z_s$  or any position at the south of it. From this we deduce that 396  $\rho \setminus \{p_1, \ldots, p_m\}$  is crossed a bounded number of times  $C_0$  by the head of  $T_1$  before time 397 step  $kt_N$ . The claim that  $\rho$  is crossed at most  $C_0$  times before time step  $kt_N$  follows since 398  $\{p_1, \ldots, p_m\}$  are by definition empty before this time step. 399

Finally, note that path  $p_1, \ldots, p_m$  cannot move away more than distance k from blocks  $b \otimes z_N$  to  $b \otimes z_S$ , so if n is large enough, the domain of  $s_2$  is guaranteed to lie entirely inside the left connected component  $A_0$  of  $\mathbb{Z}^2 \setminus \rho$ . Similarly, the blocks containing the encoding of the rightmost copy of u must lie entirely inside  $\mathbb{Z}^2 \setminus A_0$ . In particular, the configuration c of  $T_1$  reached at step  $kt_N$  must be such that  $K(c_{|\mathbb{Z}^2 \setminus A_0}) \in \Omega(n)$  by choice of u. However, Lemma 8 applied at step  $kt_N$  to  $T_1$  and  $\rho$  gives:  $K(c_{|\mathbb{Z}^2 \setminus A_0}) \leq C \log(kt_S) \in O(\log(n))$  which is a contradiction for large enough n.

# 407 **4** Universality Results

#### 4.1 Radius 3 in 2D under Rigorous Simulations: the Heat Sink Trick

Theorem 6 shows that no turedo of radius 1 can be universal for  $TUR_2(1)$  under rigorous 409 simulations. We show that this is however possible with radius 3. In order to achieve 410 universality, four key behaviours must be performed by our turedo at each simulation 411 step. It needs to read all necessary information of neighbouring blocks, compute the next 412 simulation step, write the computed letter and exit the current block (entering the correct 413 next one). Figure 6 illustrate those behaviours. To rigorously simulate a turedo of radius 414 1, it seems natural to perform the 3 behaviours interacting with neighbours on the edge of 415 the block, each on its layer, motivating a radius 3. But as we are dealing with universality, 416 the so called necessary information is not only the letters of neighbouring blocks but also 417 the transition table defining the simulated turedo. Carrying this information has a direct 418

#### 23:12 On Turedo Hierarchies and Intrinsic Universality



**Figure 6** Behaviour, in one block, of the presented 2D radius 3 turedo  $T_3 \in \mathcal{U}_2^{\leq R}(1)$ . The light blue rectangles represent the information to read of the neighbouring cells. In this example, a letter of  $A_1$  is encoded by 3 letters of  $A_3$ . We can see the 2 padding cells, the transition table (here of arbitrary small size for readability of the figure), the buffer of size 12 (to contain 4 letters of  $A_1$  encoded), the padding, the letter of the block encoded by 3 letters of  $A_3$  repeated 4 times each and again the the padding, the buffer, the table and the last 2 padding cells. The arrows represent the path followed by the turedo, entering the block in the bottom left of the figure. The blue part reads the content of the adjacent blocks, the grey one allows for turning, the orange one is where the computing takes place, the red one fetchs the computed letters (and writes them on the faces that will not be visited again before exiting the block) and the green one finishes to write the letters and exits to the next block.

<sup>419</sup> impact on the width of the reading and writing layer, we propose an intertwined zigzag to
<sup>420</sup> merge the space occupied by those two, keeping the radius 3 and taking full advantage of
<sup>421</sup> it.

# Large **Theorem 10.** $\mathcal{U}_2^{\leq_R}(1) \cap TUR_2(3) \neq \emptyset$ .

**Proof.** We show that there is  $T_3 \in TUR_2(3)$  such that for all  $T_1 \in TUR_2(1)$ :  $T_1 \leq_R T_3$ . 423 Denote  $T_3 = (A_3, Q_3, \delta_3)$ . Let's take  $T_1 \in TUR_2(1)$  some 2D turedo,  $T_1 = (A_1, Q_1, \delta_1)$ , a 424 configuration  $c_1 \in A_1^{\mathbb{Z}^2}$  and describe how  $T_3$  simulates it with square blocks  $R_{(n,n)}$  and 425  $n = 0 \mod 4$ . We first focus on the organisation of transmittable information in a given 426 block *B*, *i.e.* the transition table  $\delta_1$  and the letter  $a_1 \in A_1$  in this position in  $c_1$ . To be 427 accessible to the neighbouring blocks, this information is present on the outside edges of 428 Β, repeated that on each edge such 429 B(0,i) = B(i, n-1) = B(n-1, n-1-i) = B(n-1-i, 0). Considering a partial onto 430 letter decoding map  $\gamma: A_3^m \to A_1$  with  $m \in \mathbb{N}$ , the organisation on one edge of B is the 431 following. The first two positions B(0,0) and B(0,1) are empty or irrelevant, then the next 432  $3m|A_1|$  positions from B(0,2) to  $B(0,3m|A_1|+1)$  are the encoding of the transition table 433 with  $\gamma$  (which we assume to be a multiple of 4 without loss of generality). Positions 434  $B(0, 3m|A_1| + 2)$  to  $B(0, m(3|A_1| + 4) + 1)$  are reserved for a buffer in which 4 letters will 435 be encoded (the ones contained in the 4 neighbouring blocks). Then  $m(3|A_1| + 4) + 2$ 436

positions are empty or irrelevant, from  $B(0, m(3|A_1| + 4) + 2)$  to  $B(0, 2m(3|A_1| + 4) + 3)$ , 437 to allow the block to be spacious enough for the computation. Following that is written 438  $u \in A_3^{4m}$  such that  $u(4i+k) = \gamma(a_1)(i)$  for all  $0 \le k < 4$  (the redundancy is present to 439 ensure proper reading later). Then again, from  $B(0, 2m(3|A_1| + 6) + 4)$  to 440  $B(0, 3m(3|A_1| + 4) + 2m + 5)$  is some irrelevant padding followed by the 4m sized buffer, 441 the  $3m|A_1|$  sized transition table and 2 irrelevant position, finishing 442  $B(0, m(12|A_1|+20)+7)$ . An illustration of this distribution is represented in figure 6. 443

Let's now describe the behaviour of  $T_3$  in a block to achieve universality. As all 444 necessary information to compute is contained in a  $m(3|A_1| + 4)$  letters long word on  $A_3$ , 445 we base our construction on two types zigzags of this size : the square and the heat sink. 446 A square gadget of even size k is a back and forth k/2 times of a k sized line during which 447 the content of the original line is replicated on each four sides of the created square. This 448 copy is possible thanks to the radius of  $T_3$  being greater than 2. Its main use is to keep and 449 spread information while cornering. A heat sink gadget is a zigzag with each back and 450 forth being spaced by 2. The radius 3 of  $T_3$  allows the heat sink to still copy information 451 from the previous zag during a zig. Its purpose is to acquire and transmit data laterally 452 while being intertwined with another heat sink. Both gadget and the simulation of a block 453 are illustrated in figure 6 to illustrate the following explanation. 454

 $T_3$  enters a block B at position B(0,2) (or B(2,n-1), B(n,n-3), B(n-3,0) up to 455 rotation), it first continues forward by 2 (until reaching B(2,2)) then turns 90° 456 counter-clockwise and starts a square gadget, initialised by the copy of the transition table 457 and empty buffer available at distance 3. Those squares will be performed at each corners 458 of the block with in between a heat sink which will both transmit the transition table 459 already collected earlier and approach the outside edge of the neighbouring block at 460 distance 3, reading its information and filling the buffer. (The heat sink has access to only 461 half of this information but it has been taken care of by the redundancy detailed earlier). 462 Once the information of all four neighbouring blocks collected, right after the last letter 463 has been read,  $T_3$  turns toward the center of the block, using the space left by the padding, 464 performing one more square gadget. With a big enough transition table and a well chosen 465 encoding, computing the next transition of  $T_1$  can be performed in a square the size of the buffer and transition table encoded. As the only way to leave the center of the block is 467 now a path of width 1,  $T_3$  must transmit its computation before rejoining the edge of the 468 block. It does so with a zigzag that follows along the the inside of the reading heat sink. 469 This information is then retrieved by the writing process, with a heat sink intertwined 470 with the reading one. The writing process writes on the edge of the block on the faces 47 before the exit one and at distance 1 after. When following a square gadget, it copies the 472 transition table and writes an empty buffer (which is possible because the square not only 473 corners but copies information on all its sides), when following a reading gadget, it creates 474 a heat sink gadget of its own, fetching the computed information and writing it on the face 475 of the block. Lastly, once all the writing has been done,  $T_3$  finishes filling its block by going 476 back to the exit edge, following the writing path which had been shifted to the inside by 1 477 to allow for this, copying everything from the reading path to effectively write it on the 478 edge making it attainable and finally  $T_3$  exits the block at distance 2 from the corner. 479

# **440** 4.2 The Power of Third Dimension and Liberal Simulations

Having a third dimension available allows for a lot more freedom to simulate a turedo. Let us first focus on  $TUR_3(1)$  and rigorous simulations. In dimension 2 we had to use the heat sink trick to gather all required information and we used a radius of 3 to accommodate



**Figure 7** One block of the presented 3D radius 1 turedo  $T \in U_3^{\leq R}(1)$ . At this stage *T* comes from the block above, the front and left blocks are also non-empty and the others are empty. *T* will exit the block by entering the next block at its right. On the right is represented the disposition of information. The light blue cubes are information to read in adjacent blocks and the red ones are information written by *T* in the considered block. In this example, a letter of *A'* is encoded by 2 letters of *A*, the transition table is of arbitrary small size for readability and they are written in a cross on the faces of blocks. On the left is the behaviour of *T* in the same considered block, in blue is the reading phase (building also a skeleton), in orange the computation phase, in red the writing phase and in green the exit phase.

for a reading, a writing and an exiting layer. The third dimension allows us to shrink the radius to 1 thanks to its crossing capabilities. The trick to achieve this is to have a marker to indicate if the neighbouring block is empty or not, to prevent trying to read in spaces where the turedo will have to write later on.

# <sup>488</sup> ► Theorem 11. $\mathcal{U}_3^{\leq_R}(1) \cap TUR_3(1) \neq \emptyset$ .

**Proof.** We show that there is  $T \in TUR_3(1)$ ,  $T = (A, Q, \delta)$ , such that for all  $T' \in TUR_3(1)$ : 489  $T' \leq_R T$ . Denote  $T' = (A', Q', \delta')$ . For a block B, the transmittable information is organised 490 on the faces as follow. In the center of each faces, a marker is written, indicating that 491 information is present. On the face of the block facing the next block (the exit face), added 492 to the presence marker, is an empty cross of width 1 with only at its end a stop marker 493 indicating the end the block. Surrounding it is a cross of width 3 in which is written the 494 transition table and a buffer big enough to accommodate for the encoding of 6 letters of A'495 and one state from Q' (with the size of the transition table and the buffer plus 1 left before 496 writing). On the face of the block facing other empty blocks, the same crossing pattern is 497 used to only write the presence marker and the computed letter at distance 2 of the center 498 of the face. See figure 7 for this organisation on an example. 499

T has the following behaviour in a block, also illustrated in figure 7. First, T performs 500 a reading phase : entering a block by the middle of a face, it travels straight to the edge 501 of the face until reading the stop marker. We assume T entering by the middle of the 502 face as it truly enters at distance one of the middle but this shift of one position in one 503 direction can be remembered and corrected immediately after entering. This allows T to 504 turn and go back toward the center, reading the transition table and the buffer. All this 505 information gathered, T follows a fixed path composed of shrinking zigzags, staying in 506 the planes formed by the 3D cross centered in the block (hence the peculiar way we place 507 information, to reserve space for this path). Doing so, it reaches the center of each face 508 while carrying the transition table and filling the transition buffer when a presence marker 509

is read. As the order in which the faces are visited is fixed, T can store in its head state 510 which neighbouring blocks are empty. This reading phase has the added benefit to have 511 created a skeleton in the block, enabling easy travel for the following phases. Next T enters 512 513 the computing phase consisting of a square zigzag, using the transition table and the now filled buffer carried previously and so present in all the branches of the centered 3D cross. 514 Once the next step of the simulation computed, T can carry the letter to encode following 515 the skeleton and write it on all previously identified faces except the exit one. The last 516 phase, the exit one, consists in carrying the encoded letter to the exit face, retrieve on the 517 centered 3D cross the transition table and the buffer (reset with the new state and letter of 518 the computed transition) and writing all this information as previously presented thanks 519 to square and triangular zigzags. Finally T exits to the next block, at distance one of the 520 center of the face (to get around the reading skeleton). 4 521

The combination of 3D and liberal simulations allows to shrink the radius of any turedo to 1. In this construction, the computation of simulated transitions is done internally in the simulating turedo's head. The challenging part however is in acquiring the states of distant neighbors. Thankfully the liberal nature of the simulation allows travel through empty blocks and the 3D enables crossing paths without intersection. Still, a rigorous organisation is needed in order to prevent overlapping.

**Theorem 12.** For any radius r and any  $T_r \in TUR_3(r)$  there is  $T_1 \in TUR_3(1)$  such that  $T_r \leq T_1$ .

**Proof.** Let  $r \in \mathbb{N}_+$  and  $T_r \in \text{TUR}_3(r)$ ,  $T_r = (A_r, Q_r, \delta_r)$ . We build  $T_1 = (A_1, Q_1, \delta_1)$ , 530 with  $Q_1$  big enough to encode in one state a position  $z \in \mathbb{Z}^3$  modulo 2r of each dimension 531 and the  $|B_3(r)|$  neighbouring letters. This allows for an instant computation of  $\delta_r$  once all 532 needed information is gathered. Let b be the block size, the critical aspect of this simulation 533 is for  $T_1$  to visit all blocks  $b \otimes z'$  with  $z' \in z + B_3(r)$  for each simulation step. Therefore 534 we have to assign non intersecting exploration paths for all positions at distance less than 535 2r. To achieve this, we define  $\mathcal{C} = \{0, ..., 8r^3 - 1\}$  a set of colors and we assign the color 536  $(z_1 \mod 2r) + 2r(z_2 \mod 2r) + 4r^2(z_3 \mod 2r)$  to the block  $b \otimes (z_1, z_2, z_3)$ . By taking b = 0537  $8r^{3}l + 7$  with  $l \in \mathbb{N}_{+}$ , in each block, for each color  $c \in C$ , we can reserve tubes of width 538 *l* following the edges of a centered cube of edges of length lc + 1 and the direct extension 539 of said edges to the face of the block (see figure 8b). This creates reserved spaces for each 540 color consisting of centered nested cubes. Those cubes fill a space of  $8r^3l$ , we add 1 to have 541 a proper center and 6 to have some padding near the faces of the blocks (we will discuss 542 its necessity later). Note that l actually doesn't need to be large, l = 10 is enough. On each 543 face of the block  $b \otimes z$ , the letter  $a \in A_r \subset A_1$  is repeated in a cross pattern (see 8a). 544

 $F_{T_r}(c_r, z_r, q_r)$  is simulated as follow. Assuming  $T_1$  knows the color  $c \in C$  of block  $b \otimes z_r$ 545 (which is possible as its position in all directions modulo 2r is stored in its head state), we 546 can define a reference starting position to explore the neighbouring blocks by ordering the 547 directions of  $B_3(1)$ . Moreover, this ordering allows to decide a depth-first exploration of 548 the blocks  $b \otimes z'$  with  $z' \in B_3(r)$  passing through each block at most seven times. Once 549 the exploration done, back in block  $b \otimes z$ , the head of  $T_1$  contains all necessary information 550 to compute  $\delta_r$  and all that remains to do is to write the computed letter in a cross pattern 551 on the faces of the block. This is possible following a eulerian path, crossing only on the 552 center of the faces, hence the padding of 2 defined earlier. The 1 padding left is for  $T_1$  to 553 align itself with its next color (which is possible since it knows its current color and has 554 computed to in which block to go next). 555

#### 23:16 On Turedo Hierarchies and Intrinsic Universality



(a) The letter  $a \in A_r \subset A_1$  (b) Reserved space for the (c) One face of a block. is repeated on all faces in a cross exploration path of color  $c \in C$ .

**Figure 8** Representation of the reserved space for color c in block  $b \otimes z$ . In red are the tubes of width l on the edges of the centered cube of edges of length lc + 1. In green are the extensions of said tubes, allowing to reach the reserved space for color c in the neighbouring blocks, completing the exploration path of color c (8b). Those extensions also allow the exploration path to access the blue cross containing the letter of position z in the configuration (to either read or write)(8c).

<sup>556</sup> By combining Theorem 11 and Theorem 12, we get the existence of an intrinsically <sup>557</sup> universal 3D turedo for liberal simulations as expressed in the following corollary.

**558 Corollary 13.**  $\mathcal{U}_3^{\leq} \cap TUR_3 \neq \emptyset$ .

# 559 **5** Discussion

The problem tackled in this work depends on three parameters (radius, dimension, simulation). Our results give a rather clear picture of the simulation hierarchies in 3D, but we left several open question in the 2D case, in particular: is there a turedo in  $TUR_2(2)$ which is universal for  $TUR_2(1)$  under rigorous simulations? what if we allow liberal simulations? Actually even Theorem 6 raises questions: the simulation impossibility makes a crucial use of non-connected seeds, does this impossibility remain if we just ask simulation of orbits starting from connected seeds?

In this work we chose the square lattice in 2D (to simplify and as we also consider the 567 3D case) whereas oritatami are mostly considered on the hexagonal lattice. We don't 568 expect any significant difference on the simulation hierarchy result by changing from 569 square to hexagonal lattice on a given model (either turedos or oritatami). However, it is 570 not clear that the delay hierarchy for oritatami behaves likes the radius hierarchy for 571 turedos. In particular, we don't know if an analog of Theorem 10 holds for oritatami. The 572 key difference between a large radius turedo and a large delay oritatami is that the turedo 573 can gather information locally across obstacles, while the oritatami can only probe 574 information around that can be reached by a path of empty positions (because it can only 575 probe by trying to position a small strand of beads). 576

<sup>577</sup> We end this paper by suggesting the following two directions in order to better <sup>578</sup> understand the gap between turedos and oritatami systems: what if we restrict turedos to <sup>579</sup> 'see' only neighboring positions that can be reached through a path of r empty positions? <sup>580</sup> and what if we enrich oritatami systems by a more general 'magnetic' attraction law <sup>581</sup> between beads where pairs of distant beads can still contribute to the total amount of <sup>582</sup> attraction that the free strand at the end of the molecule is trying to maximize (let's say by <sup>583</sup> a quadratic decrease with distance up to some radius)?

References

584

#### Florent Becker, Diego Maldonado, Nicolas Ollinger, and Guillaume Theyssier. Universality in 1 585 freezing cellular automata. In Sailing Routes in the World of Computation - 14th Conference on 586 Computability in Europe, CiE 2018, Kiel, Germany, July 30 - August 3, 2018, Proceedings, pages 587 50-59,2018. doi:10.1007/978-3-319-94418-0\\_5. 588 2 Laurent Boyer and Guillaume Theyssier. On factor universality in symbolic spaces. In Mathematical Foundations of Computer Science 2010, pages 209–220. Springer Berlin Heidelberg, 590 2010. doi:10.1007/978-3-642-15155-2\_20. 591 3 Matthew Cook, Yunhui Fu, and Robert T. Schweller. Temperature 1 self-assembly: 592 Deterministic assembly in 3D and probabilistic assembly in 2D. In SODA2011: Proceedings 593 of the 22nd Annual ACM-SIAM Symposium on Discrete, pages 570–589, 2011. 594 Marianne Delorme, Jacques Mazoyer, Nicolas Ollinger, and Guillaume Theyssier. Bulking ii: 4 595 Classifications of cellular automata. Theor. Comput. Sci., 412(30):3881–3905, 2011. 596 5 David Doty, Jack H. Lutz, Matthew J. Patitz, Robert T. Schweller, Scott M. Summers, and 59 Damien Woods. The tile assembly model is intrinsically universal. In FOCS2012: Proceedings of 598 the 53rd Annual IEEE Symposium on Foundations of Computer Science, pages 302–310, 2012. 599 Pierre Étienne Meunier and Damien Woods. The non-cooperative tile assembly model is 6 600 not intrinsically universal or capable of bounded Turing machine simulation. In STOC 2017: 601 Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, pages 328–341, 602 2017603 7 Cody Geary, Guido Grossi, Ewan K. S. McRae, Paul W. K. Rothemund, and Ebbe S. Andersen. 604 RNA origami design tools enable cotranscriptional folding of kilobase-sized nanoscaffolds. 605 Nature Chemistry, 13:549–558, 2021. 606 Cody Geary, Pierre-Étienne Meunier, Nicolas Schabanel, and Shinnosuke Seki. Programming 607 8 biomolecules that fold greedily during transcription. In MFCS2016: Proceedings of the 41st 608 International Symposium on Mathematical Foundations of Computer Science, volume 58 of LIPIcs, 609 pages 43:1–43:14, 2016. 610 9 Cody Geary, Pierre-Étienne Meunier, Nicolas Schabanel, and Shinnosuke Seki. Oritatami: A 611 computational model for molecular co-transcriptional folding. International Journal of Molecular 612 Sciences, 9(2259), 2019. doi:https://doi.org/10.3390/ijms20092259. 613 10 Cody Geary, Pierre-Étienne Meunier, Nicolas Schabanel, and Shinonsuke Seki. Proving the 614 Turing universality of oritatami cotranscriptional folding. In ISAAC 2018: Proceedings of the 615 29th International Symposium on Algorithms and Computation, volume 123 of LIPIcs, pages 23:1-616 23:13, 2018. 617 11 Cody Geary, Paul W. K. Rothemund, and Ebbe S. Andersen. A single-stranded architecture for 618 cotranscriptional folding of RNA nanostructures. Science, 345:799-804, 2014. 619 12 Cody W. Geary, Pierre-Étienne Meunier, Nicolas Schabanel, and Shinnosuke Seki. Proving the 620 turing universality of oritatami co-transcriptional folding. In Wen-Lian Hsu, Der-Tsai Lee, and 621 Chung-Shou Liao, editors, 29th International Symposium on Algorithms and Computation, ISAAC 622 2018, December 16-19, 2018, Jiaoxi, Yilan, Taiwan, volume 123 of LIPIcs, pages 23:1–23:13. Schloss 623 Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ISAAC.2018.23. 624 13 Jacob Hendricks, Matthew J. Patitz, and Trent A. Rogers. Universal simulation of directed 625 systems in the abstract tile assembly model requires undirectedness. In 2016 IEEE 57th Annual 626 Symposium on Foundations of Computer Science (FOCS). IEEE, oct 2016. doi:10.1109/focs. 627 2016.90. 628 14 James I. Lathrop, Jack H. Lutz, Matthew J. Patitz, and Scott M. Summers. Computability 629 and complexity in self-assembly. Theory Comput. Syst., 48(3):617–647, 2011. doi:10.1007/ 630 s00224-010-9252-0. 631 15 Ming Li and Paul Vitányi. Algorithmic complexity. In *Texts in Computer Science*, pages 101–195. Springer New York, 2008. doi:10.1007/978-0-387-49820-1\_2. 633 Pierre-Etienne Meunier, Matthew J. Patitz, Scott M. Summers, Guillaume Theyssier, Andrew 16 634 Winslow, and Damien Woods. Intrinsic universality in tile self-assembly requires cooperation. 635

**CVIT 2016** 

# 23:18 On Turedo Hierarchies and Intrinsic Universality

- In Chandra Chekuri, editor, Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on
   Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014, pages 752–771. SIAM,
- 638 **2014.** doi:10.1137/1.9781611973402.56.
- Daria Pchelina, Nicolas Schabanel, Shinnosuke Seki, and Guillaume Theyssier. Oritatami
   Systems Assemble Shapes No Less Complex Than Tile Assembly Model (ATAM). In Petra
- Berenbrink and Benjamin Monmege, editors, 39th International Symposium on Theoretical Aspects
- of Computer Science (STACS 2022), volume 219 of Leibniz International Proceedings in Informatics
   (LIPIcs), pages 51:1–51:23, Dagstuhl, Germany, 2022. Schloss Dagstuhl Leibniz-Zentrum für
- 644 Informatik. doi:10.4230/LIPIcs.STACS.2022.51.
- Paul W. K. Rothemund, Nick Papadakis, and Erik Winfree. Algorithmic self-assembly of DNA
   Sierpinski triangles. *PLoS Biology*, 2:2041–2053, 2004.
- Erik Winfree. *Algorithmic Self-Assembly of DNA*. PhD thesis, California Institute of Technology,
   1998.

# **A** Proof of Theorem 5

**Proof.** We prove the result for d = 2, the argument can be generalized to higher dimension straightforwardly (by completing 2D configurations by  $\perp$  everywhere else).

For the first item, simply consider a turedo  $T_{r+1} \in TUR_2(r+1)$  that has the following 652 behavior when the head in position (0,0) has only  $\perp$  letters at the north and at the south 653 of its current position: read the letters at positions (r + 1, 0) and (0, r + 1) and move to the 654 north is they are equal and to the south otherwise. Consider any turedo  $T_r \in TUR_2(r)$  and 655 any block size  $b \in \mathbb{N}^2_+$ . To simulate  $T_{r+1}$  fuzzlessly,  $T_r$  has to move either to block  $b \otimes (0,1)$ 656 or block  $b \otimes (0, -1)$  depending on blocks  $b \otimes (r + 1, 0)$  and  $b \otimes (0, r + 1)$  without entering 657 into any other neighboring block: this is impossible, because with radius r turedo  $T_r$  can't 658 have any information about either  $b \otimes (r+1,0)$  or  $b \otimes (0,r+1)$  before making a decisive 659 move (by entering inside either  $b \otimes (0, 1)$  or  $b \otimes (0, -1)$ ) so it will fail to correctly simulate 660 661 the orbit of at least one seed.

The second item can be proved similarly, using a pumping trick on the alphabet: for any 662 fixed  $T_r \in \text{TUR}_2(r)$  with alphabet of cardinal k, choose  $T_2 \in \text{TUR}_2(2)$  with an alphabet 663 strictly larger than  $k^{r^2}$  so that at least on dimension of the block size b of any potential 664 simulation of  $T_2$  by  $T_r$  has to be at least r + 1 (otherwise there is simply no way to code 665 all letters of  $T_2$  on different blocks of size b). Then, choosing  $T_2$  to have the same behavior 666 as  $T_{r+1}$  above, we get the same contradiction: there is a direction of b, let's say the vertical 667 one, which overwhelms the radius r of  $T_r$  so  $T_r$  won't be able to read the content of block 668  $b \otimes (0,2)$  before making a decisive move and will therefore fail to correctly simulate at least one orbit. 670

# 671 **B** Proof of Lemma 8

<sup>672</sup> **Proof.** We show that  $c_{|\mathbb{Z}^2 \setminus A_0}$  can be computed from the following description  $\mathcal{D}$ :

the finite list of positions on  $\rho$  that are visited by the head during the *n* first steps of the run starting from *s*;

the list of events corresponding to each such position z given as a triple: time at which the head leaves position z, letter written at that step, and move made by the head at that step.

This description is of size  $O(\log(n))$  because both positions z and time steps occurring in the above lists are bounded by n by definition (recall that the head is initially at (0,0)).

Because *T* is of radius 1 and  $\rho$  is 4-connected, each time the head of the turedo is neither in  $A_0$  nor on  $\rho$ , the local transition does not depend on the current configuration on domain  $A_0$ . A Turing machine can therefore compute  $c_{|\mathbb{Z}^2 \setminus A_0}$  from this description by maintaining the following partial information step by step:

<sup>684</sup> • the current configuration restricted to domain  $\mathbb{Z}^2 \setminus A_0$ ,

the partial information on the head position z: the exact position if  $z \notin A_0$  or the state "undefined" else.

This information is straightforward at the initial step since  $\mathcal{D}(s) \subseteq A_0$  so the head is in  $A_0$ and the configuration is  $\perp$  everywhere outside  $A_0$ . The partial information at step n is enough to give  $c_{|\mathbb{Z}^2 \setminus A_0}$  and it is updated from one step i to the next i+ as follows:

<sup>690</sup> if the partial information on the head at step *i* is undefined and time step i + 2 does not <sup>691</sup> appear in the lists of  $\mathcal{D}$ , then don't change the partial information (the head is in  $A_0$  and <sup>692</sup> won't move to  $\rho$  at step i + 1);

# 23:20 On Turedo Hierarchies and Intrinsic Universality

- if the head information is undefined but step i + 2 appears in  $\mathcal{D}$ , then updates the head
- position to the position on  $\rho$  that corresponds to the item stamped by time steps i + 2 in  $\mathcal{D}$ ;
- <sup>696</sup> if the head position is on  $\rho$  then some item in the list of  $\mathcal{D}$  must be stamped by time step <sup>697</sup> i+1 and gives all the information to update both the head position and the configuration <sup>698</sup> on  $\rho$ ;
- <sup>699</sup> = finally if the head position is neither in  $A_0$  nor on  $\rho$ , then the knowledge of the current
- <sup>700</sup> configuration restricted to domain  $\mathbb{Z}^2 \setminus A_0$  is enough to update the partial information <sup>701</sup> (position and partial configuration).
- 702