# TOPOLOGICAL ENTROPY OF TURING COMPLETE DYNAMICS

RENZO BRUERA, ROBERT CARDONA, EVA MIRANDA,
AND DANIEL PERALTA-SALAS

Abstract. We explore the relationship between Turing completeness and topological entropy of dynamical systems. We first prove that a natural class of Turing machines that we call "regular Turing machines" (which includes most of the examples of universal Turing machines) has positive topological entropy. We deduce that any Turing complete dynamics with a continuous encoding that simulates a universal machine in this class is chaotic. This applies to our previous constructions of Turing complete area-preserving diffeomorphisms of the disk and 3D stationary Euler flows. The article concludes with an appendix written by Ville Salo that introduces a method to construct universal Turing machines that are not regular and have zero topological entropy.

## 1. Introduction

Turing machines are one of the most popular models of computation and can be understood as dynamical systems on the space of bi-infinite sequences. In two breakthrough works [25, 26], Cris Moore introduced the idea of simulating a Turing machine using continuous dynamical systems in the context of classical mechanics. His rough idea was to embed both the space of sequences and the discrete dynamics of the Turing machine into the phase space of a vector field and its continuous dynamics, respectively, using suitable encoding functions. The well-known existence of Turing machines that are universal, i.e., that can simulate any other Turing machine, led him to introduce the definition of a Turing complete dynamical system.

Renzo Bruera is supported by the Spanish grants FPU20/07006 and PID2021-123903NB-I00, funded by MCIN/AEI/10.13039/501100011033, and by the Catalan grant 2021-SGR-00087.

Robert Cardona and Eva Miranda are partially supported by the AEI grant PID2019-103849GB-I00 of MCIN/ AEI /10.13039/501100011033, and AGAUR grant 2021 SGR 00603. Eva Miranda is supported by the Catalan Institution for Research and Advanced Studies via an ICREA Academia Prize 2021 and by the Spanish State Research Agency, through the Severo Ochoa and María de Maeztu Program for Centers and Units of Excellence in R&D (project CEX2020-001084-M).

Daniel Peralta-Salas is supported by the grants CEX2023-001347-S, RED2022-134301-T and PID2022-136795NB-I00 funded by MCIN/AEI/10.13039/501100011033. R.C., E.M. and D.P.-S. acknowledge partial support from the grant "Computational, dynamical and geometrical complexity in fluid dynamics", Ayudas Fundación BBVA a Proyectos de Investigación Científica 2021.

A striking corollary of his ideas was the discovery of a new type of complexity in dynamics, stemming from the fact that certain decision problems, such as the halting problem, cannot be decided by a universal Turing machine. This yields the conclusion that any Turing complete dynamical system exhibits undecidable behavior for the problem of determining whether the trajectories whose initial data belong to a certain computable set will enter a certain computable open set or not after some time. A priori this computational complexity is very different from the standard way one measures complexity in dynamics, specifically the positivity of the topological entropy, which leads to the usual chaotic behavior.

Since Moore's foundational work, many authors have revisited the problem of simulating Turing machines with dynamical systems in different contexts. This includes low dimensional dynamics [19], polynomial vector fields [13], closed-form analytic flows [20, 14], potential well dynamics [34] and fluid flows [7, 5, 6, 8]. As explained in the previous paragraph, these constructions yield complex dynamical systems with undecidable orbits, which are able to simulate any computer algorithm. It is then natural to ask about the relationship between computational complexity and topological entropy, or more precisely: does every universal Turing machine have positive topological entropy? Is every Turing complete dynamical system chaotic? The answer to this question is necessarily delicate, as illustrated by the recent construction of a Turing complete vector field of class $C^\infty$ on $\mathbb{S}^2$ with zero topological entropy [6].

Our goal in this article is to explore conditions under which a universal Turing machine and a Turing complete dynamics is also chaotic in the usual sense of positive topological entropy. The first step is to analyze the entropy of (universal) Turing machines directly understood as symbolic dynamical systems. The study of Turing machines from this perspective has been developed by several authors, see e.g. [21, 2, 18].

Under suitable conditions, we will be able to construct dynamical systems that exhibit both undecidable paths and chaotic invariant sets of horseshoe type. Our construction is based on the notion of regular Turing machine, which is presented in Section 4. Our main theorem is that regularity is a computable criterion that implies that the machine has positive topological entropy. Positivity of the entropy of a Turing machine was characterized by Jeandel [16]. However, unlike our criterion, this characterization is not computable because determining if a one-tape Turing machine has positive topological entropy is undecidable in general [10].

**Theorem 1.** *Any regular Turing machine has positive topological entropy.*

We did not find any examples in the literature of universal Turing machines that are not regular, although artificial examples can certainly be constructed (see Appendix A). There are also (reversible) universal Turing machines that are regular. It is important to emphasize that in Section 5 we relate the topological entropy of the symbolic system associated with a Turing machine with the topological entropy of not necessarily symbolic dynamical systems that are Turing complete. Other works that have analyzed the relationship between the topological entropy (and its computability) of symbolic systems and more general dynamical systems are [11, 12].

Combining Theorem 1 with the construction of Turing complete diffeomorphisms of the disk presented in [7] and the continuity of the encodings used there, we easily get the following corollary:

**Corollary 1.** *The Turing complete smooth area-preserving diffeomorphism of the disk $\varphi : D \to D$ constructed in* [7]*, which exhibits a compact invariant set $K$ homeomorphic to the square Cantor set, has positive topological entropy whenever the simulated universal Turing machine is regular.*

The technique introduced in [7], which is based on the suspension of Turing complete area-preserving diffeomorphisms of the disk, immediately yields the construction of Reeb flows on $\mathbb{S}^3$ with chaotic invariant sets that are Turing complete. As argued in [7] there are many compatible Riemannian metrics $g$ that make these Reeb flows stationary solutions of the Euler equations on $(\mathbb{S}^3, g)$. Incidentally, these metrics cannot be optimal (or critical) because the Reeb flows have positive topological entropy [24].

This article is organized as follows. In Section 2 we recall the usual interpretation of Turing machines as dynamical systems on compact metric spaces and prove some auxiliary results. In Section 3 we define the topological entropy of a Turing machine and show how it is related to the usual definition of topological entropy in dynamics. Finally, the main theorem is proved in Section 4 and its corollary in Section 5. For the sake of completeness we include Appendix A written by Ville Salo, where a universal Turing machine with zero topological entropy is constructed; in particular, such a UTM is not regular.

## 2. Turing machines as dynamical systems

In this section we explain how to define a continuous dynamical system on a compact metric space using a Turing machine. We also introduce a notion of universal Turing machine that is particularly convenient to study dynamical properties.

2.1. **The global transition function.** A Turing machine $T = (Q, q_0, q_{halt}, \Sigma, \delta)$ is defined by:

- A finite set $Q$ of "states" including an initial state $q_0$ and a halting state $q_{halt}$.
- A finite set $\Sigma$ which is the "alphabet" with cardinality at least two. It has a special symbol, denoted by 0, that is called the blank symbol.
- A transition function $\delta : Q \setminus \{q_{halt}\} \times \Sigma \longrightarrow Q \times \Sigma \times \{-1, 0, 1\}$.

The evolution of a Turing machine is described by an algorithm. At any given step, the configuration of the machine is determined by the current state $q \in Q$ and the current tape $t = (t_n)_{n\in\mathbb{Z}} \in \Sigma^{\mathbb{Z}}$. The pair $(q, t)$ is called a configuration of the machine. Any real computational process occurs throughout configurations such that every symbol in the tape $t$ is 0 except for finitely many symbols. A configuration of this type will be called compactly supported.

The algorithm is initialized by setting the current configuration to be $(q_0, s)$, where $s = (s_n)_{n\in\mathbb{Z}} \in \Sigma^{\mathbb{Z}}$ is the input tape. Then the algorithm runs as follows:

(1) Set the current state $q$ as the initial state and the current tape $t$ as the input tape.
(2) If the current state is $q_{halt}$, then halt the algorithm and return $t$ as output. Otherwise, compute $\delta(q, t_0) = (q', t'_0, \varepsilon)$, with $\varepsilon \in \{-1, 0, 1\}$.
(3) Replace $q$ with $q'$, and change the symbol $t_0$ by $t'_0$, obtaining the tape $\tilde{t} = ...t_{-1}.t'_0t_1...$ (as usual, we write a point to denote that the symbol at the right of that point is the symbol at position zero).
(4) Shift $\tilde{t}$ by $\varepsilon$ obtaining a new tape $t'$, then return to step (2) with the current configuration $(q', t')$. Our convention is that $\varepsilon = 1$ (resp. $\varepsilon = -1$) corresponds to the left shift (resp. the right shift).

Given a Turing machine $T$, its transition function can be decomposed as

$$\delta = (\delta_Q, \delta_\Sigma, \delta_\varepsilon) : Q \times \Sigma \to Q \times \Sigma \times \{-1, 0, 1\}\,.$$

Here the maps $\delta_Q, \delta_\Sigma$ and $\delta_\varepsilon$ denote the composition of $\delta$ with the natural projections of $Q \times \Sigma \times \{-1, 0, 1\}$ onto the corresponding factors. The Turing machine can be understood as a dynamical system $(R_T, X_T)$, where the phase space is

$$X_T := Q \times \Sigma^{\mathbb{Z}}$$

and the action

$$R_T : X_T \to X_T$$

is the *global transition function*, which is given by

$$R_T(q, (\dots t_{-1}.t_0t_1 \dots)) := \begin{cases} (q', (\dots t_{-1}t'_0.t_1 \dots))\,, & \text{if } \delta_\varepsilon(q, t_0) = 1\,, \\ (q', (\dots t_{-2}.t_{-1}t'_0 \dots))\,, & \text{if } \delta_\varepsilon(q, t_0) = -1\,, \\ (q', (\dots, t_{-1}.t'_0t_1 \dots))\,, & \text{if } \delta_\varepsilon(q, t_0) = 0\,, \end{cases} \qquad (1)$$

for $q \neq q_{halt}$, with $t'_0 = \delta_\Sigma(q, t_0)$ and $q' = \delta_Q(q, t_0)$. For $q = q_{halt}$, several extensions of the global transition function exist, the simplest and most natural being

$$R_T(q_{halt}, (\dots t_{-1}.t_0t_1 \dots)) := (q_{halt}, (\dots t_{-1}.t_0t_1 \dots))\,. \qquad (2)$$

This is equivalent to extending the transition function on halting configurations as $\delta(q_{halt}, t_0) := (q_{halt}, t_0, 0)$; all along this article we shall assume that the transition function is extended this way. We will use the notation

$$X^c_T = \{\text{set of compactly supported configurations}\}\,,$$

that is, the set of configurations of $T$ with a tape that has only finitely many non-blank symbols.

2.2. **Properties of $X_T$ and $R_T$.** Given a Turing machine $T$, for each $x = (q,t) \in X_T$ let us set $x_q := q$ and $x_i := t_i$. If we endow the finite sets $Q$ and $\Sigma$ with the discrete topology, $X_T$ becomes a compact metric space endowed with the complete metric

$$d\left(x, x'\right) := \begin{cases} 1, & \text{if } x_q \neq x'_q\,, \\ 2^{-n}, & \text{if } x_q = x'_q \text{ and } n = \sup\{k : x_i = x'_i \ \forall |i| < k\}\,. \end{cases} \tag{3}$$

It is elementary to check the continuity of the global transition function for this metric:

**Lemma 1.** *The global transition function $R_T : X_T \to X_T$ is continuous for the metric $d$.*

The metric $d$ defines a topology in the compact space $X_T$. Then, several natural sets and functions are open and continuous for this topology. Particularly important are the halting domain and the halting time:

**Definition 1** (Halting domain and halting time)**.** Let $T$ be a Turing machine with corresponding global transition function $R_T$ acting on the phase space $X_T$. As usual, the *halting domain* of $T$ is defined as the set

$$X_T^H := \{x \in \{q_0\} \times \Sigma^{\mathbb{Z}} : \exists N \in \mathbb{N} \text{ such that } T^N\left(x\right)_q = q_{halt}\}\,. \tag{4}$$

The number $N \in \mathbb{N}$ such that $T^N(x)_q = q_{halt}$ is called the *halting time* of $x$.

The following important lemma shows that the halting domain $X_T^H$ is open in $X_T$ and the halting time $N$ is continuous (both properties with respect to the natural topology on $X_T$ introduced before).

**Lemma 2.** *The halting domain of a Turing machine $T$ is open in $X_T$, and the halting time function $N : X_T^H \to \mathbb{N}$ is continuous.*

*Proof.* To see that $X_T^H$ is open we simply notice that

$$X_T^H = \left(\bigcup_{n\geq 1} \{x \in X_T : R_T^n\left(x\right)_q = q_{halt}\}\right) \cap \left(\{q_0\} \times \Sigma^{\mathbb{Z}}\right)\,.$$

Since the first term is the union of open sets, and the second term is also open, the claim follows.

To see the continuity of $N$, we observe that for a fixed $x \in X_T$, since $n := N(x)$ is finite, the machine can read (at most) the cells of the sequence $x$ in positions $[-n, n]$. Therefore, any other input coinciding with $x$ in the range $[-n, n]$ will halt after the same number of steps. We then conclude that for each natural number $n$, the set $N^{-1}(n) \subseteq X_T$ contains an open ball of radius $2^{-n}$ and center $x$, which implies the continuity of $N$. □

It is also convenient to introduce the notion of output function of a Turing machine, which assigns to the halting domain $X_T^H$ the set of elements in $X_T$ that are reached when halting. Next, we introduce the precise definition and the key property that the output function is continuous.

**Definition 2** (Output function)**.** For each Turing machine $T$, we define the output function $\Psi_T : X_T^H \to X_T$ as

$$\Psi_T(x) := R_T^{N(x)}(x) \quad \forall x \in X_T^H\,, \tag{5}$$

where $N(x)$ is the halting time function.

**Lemma 3.** *The output function is continuous.*

*Proof.* Fix $z \in X_T^H$. Since the halting time $N$ is continuous, then it is locally constant, so there is a ball $B(z) \subset X_T^H$ around $z$ where $N(x) = N(z)$ for all $x \in B(z)$. Therefore, locally we can write $\Psi_T(x) = R_T^n(x)$ for all $x \in B(z)$, with $n := N(z)$. Since $R_T$ is continuous (cf. Lemma 1), it follows that $\Psi_T$ is continuous in $B(z)$ for all $z$, so $\Psi_T$ is a continuous function. □

2.3. **Universal Turing machines.** Finally, we introduce a definition of *universal Turing machine (UTM)* following Morita in [27] (which is based on [30]). We remark that this definition is more general than the classical ones used by Shannon [31] and Minsky [22] in their foundational works.

Let $\{T_n\}_{n=1}^\infty$ be an enumeration of all Turing machines and define the space

$$\mathcal{X} := \bigcup_{n\geq 1} X_{T_n}^c$$

of all compactly supported configurations of $T_n$.

**Definition 3** (Universal Turing machine)**.** A Turing machine $U$ is universal if there exist computable functions $c : \mathbb{N} \times \mathcal{X} \to X_U^c$ and $d : X_U^c \to \mathcal{X}$ such that for each $n \in \mathbb{N}$,

$$x \in X_{T_n}^H \cap X_{T_n}^c \iff c(n,x) \in X_U^H \cap X_U^c\,, \tag{6}$$

and

$$\Psi_{T_n}(x) = (d \circ \Psi_U \circ c(n,\cdot))(x) \quad \forall x \in X_{T_n}^H \cap X_{T_n}^c\,. \tag{7}$$

There is a technical detail that we have omitted, which is that we want to consider each element in $X_{T_n}^c$ as a *finite word* so that the boundary of the support of a configuration is explicitly given. We refer to the appendix for a detailed discussion of the definition of universal Turing machine. Most if not all examples of universal Turing machines satisfy this Definition 3; examples can be found in [31, 22, 27]. A particularly well-known property of universal Turing machines is that the halting problem for compactly supported inputs is undecidable for these machines.

## 3. Topological entropy of Turing machines

The topological entropy of Turing machines has been studied in [29] from a dynamical systems viewpoint, and its computability was analyzed by Jeandel in [16], by working with the entropy as is done with the speed of the machine. In this section, we recall Oprocha's formula to compute the topological entropy of a Turing machine and introduce Moore's generalized shifts as a model to describe the dynamics of any Turing machine. As we will see, the topological entropy of the generalized shift coincides with that of the Turing machine it simulates.

Let $T$ be a Turing machine. As argued in Section 2, it can be described using the global transition function $R_T$, which is a continuous dynamical system on the compact metric space $(X_T, d)$. In this setting, one can use the definition of topological entropy given by Bowen and Dinaburg [4, 9], which is equivalent to the original one by Adler, Konheim, and McAndrew [1].

In [29, Theorem 3.1] Oprocha obtained a remarkable formula showing that the topological entropy of $T$ can be computed as the following limit:

$$h(T) := \lim_{n\to\infty} \frac{1}{n} \log |S(n, R_T)| \,, \tag{8}$$

where $|\cdot|$ denotes the cardinality of the finite set $S(n, R_T)$, which is defined as

$$S(n, R_T) := \{u \in (Q\times\Sigma)^n : \exists x \in X_T \text{ s.t. } u_i = (R_T^{i-1}(x)_q, R_T^{i-1}(x)_0)\,\} \,, \tag{9}$$

where $u_i$ is defined for $i = 1, ..., n$ and denotes the $i^{\text{th}}$ component of $u$. Here, $R_T^j$ denotes the $j$-th iterate of the map $R_T$. Usually, the set $S(n, R_T)$ is called the set of $n$-words allowed for the Turing machine $T$.

In [26] Moore introduced a generalization of the shift map that he called a *generalized shift*, which is a class of dynamical systems that allows one to describe any Turing machine, and is different from the global transition function $R_T$. Let us introduce Moore's idea and how it connects with the dynamics and topological entropy of $(R_T, X_T)$.

**Definition 4** (Generalized shift)**.** Let $A$ be a finite set. For each $s \in A^{\mathbb{Z}}$ and $J, J' \in \mathbb{Z}$, we denote by $s_{[J,J']}$ the finite string containing the elements of $s$ in positions $J$ to $J'$, and we denote by $\oplus$ the operation of string concatenation. A generalized shift is a map $\Delta : A^{\mathbb{Z}} \to A^{\mathbb{Z}}$ that is given by

$$\Delta(s) = \sigma^{F\left(s_{[-r,r]}\right)}\left(\dots s_{(-\infty,-r-1]} \oplus G(s_{[-r,r]}) \oplus s_{[r+1,\infty)}\right) \,.$$

Here, $r$ is a natural number, $\sigma$ is the Bernoulli shift and $F$ and $G$ are maps

$$F : A^{2r+1} \to \mathbb{Z}\,,$$
$$G : A^{2r+1} \to A^{2r+1}\,.$$

As in Section 2.1, if we endow $A$ with the discrete topology then $A^{\mathbb{Z}}$ is a compact metric space and the generalized shift $\Delta$ is always continuous (independently of the choice of $r$, $F$, and $G$). We observe that the complete metric is defined as in the second formula of Equation (3).

**Remark 1.** Without any loss of generality one may assume that $A = \{0,1\}$, and thus $A^{\mathbb{Z}}$ is homeomorphic to the square ternary Cantor set $C^2 \subset \mathbb{R}^2$ via the homeomorphism given by

$$e(s) = \left( \sum_{k=1}^{\infty} s_{-k} \frac{2}{3^k}, \sum_{k=1}^{\infty} s_{k-1} \frac{2}{3^k} \right). \tag{10}$$

Accordingly, any generalized shift can be viewed as a dynamical system on the square Cantor set.

The connection between Turing machines and generalized shifts was established by Moore [26]. Let $T$ be a Turing machine with set of states $Q$ and set of symbols $\Sigma$, and define $A = Q \cup \Sigma$. With $X_T = Q \times \Sigma^{\mathbb{Z}}$, we also define the injective map

$$\varphi : X_T \to A^{\mathbb{Z}} \tag{11}$$

$$(q,t) \mapsto (\dots t_{-1}.qt_0 \dots). \tag{12}$$

Then:

**Theorem 2** (Moore). *Given a Turing machine $T$, there exists a generalized shift $\Delta$ on $A^{\mathbb{Z}}$ such that its restriction to $\varphi(X_T)$ satisfies*

$$\Delta|_{\varphi(X_T)} = \varphi \circ R_T \circ \varphi^{-1}|_{\varphi(X_T)}. \tag{13}$$

We claim that, in fact, the map $\varphi : X_T \to A^{\mathbb{Z}}$ is more than injective, it is a topological conjugation between the dynamical systems $\Delta$ and $R_T$. This will allow us to relate both dynamics.

**Lemma 4.** *The map $\varphi$ is a homeomorphism onto its image.*

*Proof.* Let us see that $\varphi$ is continuous. Indeed, for any $\varepsilon > 0$ choose $k$ such that $\varepsilon > 2^{-k}$. Let $x, x' \in X_T$ be such that $d(x,x') < 2^{-k}$. This means that

$$x_q = x'_q \text{ and } x_i = x'_i \text{ for all } |i| < k\,.$$

Then, clearly

$$d\left(\varphi(x), \varphi(x')\right) < 2^{-k} < \varepsilon\,,$$

thus implying continuity.

Similarly we may show that $\varphi^{-1}|_{\varphi(X_T)}$ is continuous. Fix $\varepsilon > 0$, let $k$ be such that $\varepsilon > 2^{-k}$ and let $y, y' \in \varphi(X_T) \subset A^{\mathbb{Z}}$ such that $d(y,y') < 2^{-(k+1)}$. That is, $y_i = y'_i$ for all $|i| < k+1$. Note that since $y$ and $y'$ are in $\varphi(X_T)$, they are of the form

$$y = y_{(-\infty,-1]} \oplus y_0 \oplus y_{[1,\infty)}$$

with $y_0 \in Q$ and $y_i \in \Sigma$ for all $i \neq 0$. Then, clearly

$$d\left(\varphi^{-1}(y), \varphi^{-1}(y')\right) < 2^{-k}\,,$$

and the lemma follows. □

For the purposes of this article, the main consequence of the aforementioned conjugation between Turing machines and generalized shifts is the following result, which shows the connection between the topological entropy of both systems. Since a generalized shift $\Delta$ is a map on a compact metric space, its topological entropy $h(\Delta)$ can be computed using Bowen-Dinaburg's definition, as before.

**Proposition 1.** *Let $T$ be a Turing machine and $\Delta$ its associated generalized shift. Then $h(\Delta) \geq h(T)$. In particular, if $T$ has positive topological entropy, then so does its associated $\Delta$.*

*Proof.* Since $\Delta|_{\varphi(X_T)} = \varphi \circ R_T \circ \varphi^{-1}|_{\varphi(X_T)}$, cf. Theorem 2, it is clear that the subset $\varphi(X_T) \subset A^{\mathbb{Z}}$ is forward invariant under the iterations of the generalized shift $\Delta$. Moreover, this property and Lemma 4 also show that the maps $R_T$ and $\Delta$ are topologically conjugate via the homeomorphism $\varphi : X_T \to \varphi(X_T)$. The invariance of the topological entropy under homeomorphisms and the fact that

$$h(\Delta) \geq h(\Delta|_{\varphi(X_T)}) ,$$

see e.g. [17, Section 3.1.b], complete the proof of the proposition. $\square$

## 4. A criterion for positive topological entropy

In this section we prove the main result of this work, which shows that a special type of Turing machine (what we call a regular Turing machine) has positive topological entropy. To this end, we exploit the fact that a dynamical system has positive topological entropy if it exhibits an invariant subset where the entropy is positive.

### 4.1. **Positive entropy for strongly regular Turing machines.**

To illustrate the method of proof, we first consider the class of strongly regular Turing machines:

**Definition 5** (Strongly regular Turing machine)**.** A Turing machine $T$ is *strongly regular* if for some $\varepsilon \in \{-1, 1\}$ there exists a subset $Q' \times \Sigma' \subset (Q \setminus \{q_{halt}\}) \times \Sigma$, with $|\Sigma'| \geq 2$, such that $\delta_Q(Q' \times \Sigma') \subseteq Q'$ and $\delta_\varepsilon|_{Q' \times \Sigma'} = \varepsilon$.

Given a strongly regular Turing machine $T$ with $\varepsilon = 1$ (the case $\varepsilon = -1$ is analogous), we claim that the subset

$$Y_T := \{x \in X_T : x_q \in Q',\ x_i \in \Sigma' \text{ for all } i \geq 0\} = Q' \times \Sigma^{\mathbb{N}_0} \times \Sigma'^{\mathbb{N}} \subset X_T$$

is forward invariant under the global transition function $R_T$. Here $\mathbb{N}_0$ is the set of natural numbers without $\{0\}$.

**Lemma 5.** *$Y_T$ is forward invariant under $R_T$.*

*Proof.* Indeed, let $x = (q, (t_i)) \in Y_T$. We have

$$R_T(x) = (q', (\dots t_{-1}t'_0.t_1 \dots)) = (q', (\dots s_{-1}.s_0 s_1 \dots))$$

with $s_i := t_{i+1} \in \Sigma'$ for all $i \geq 0$ and $q' = \delta_Q(q, t_0) \in Q'$ by hypothesis. Therefore, $R_T(x) \in Y_T$ as claimed. $\square$

This lemma allows us to prove the following sufficient condition for positive topological entropy.

**Theorem 3.** *Let $T$ be a strongly regular Turing machine. Then*

$$h(T) \geq \log |\Sigma'| > 0\,.$$

*Proof.* As before, let us consider that $T$ is strongly regular with $\varepsilon = 1$, the other case being completely analogous. To estimate the topological entropy of $T$ we use Oprocha's formula in Equation (8). We claim that for each $n \geq 1$, $|S(n, R_T)| \geq |\Sigma'|^n$. To see this, fix $n$, let $q^0 \in Q'$ and consider any finite sequence $\{a'_0, a'_1, \dots, a'_{n-1}\} \subset \Sigma'$. Choose any $x \in X_T$ such that $x_q = q^0$ and $x_i = a'_i$ for $i = 0, \dots, n-1$. We define $q^i = R_T^i(x)_q$ for $i = 1, \dots, n-1$ and finally we set $u = \big((q^0, a'_0), \dots, (q^{n-1}, a'_{n-1})\big) \in (Q' \times \Sigma')^n$. Since $\delta_\varepsilon|_{Q' \times \Sigma'} = 1$, from (1) we infer that

$$\begin{aligned}
(R_T^0(x)_q, R_T^0(x)_0) &= (q^0, t_0) = (q^0, a'_0) = u_0\,, \\
(R_T^1(x)_q, R_T^1(x)_0) &= (q^1, t_1) = (q^1, a'_1) = u_1\,, \\
&\vdots \\
(R_T^{n-1}(x)_q, R_T^{n-1}(x)_0) &= (q^{n-1}, t_{n-1}) = (q^{n-1}, a'_{n-1}) = u_{n-1}\,,
\end{aligned}$$

so $u \in S(n, R_T)$. Since this holds for all finite sequences of length $n$ in $\Sigma'$, we conclude that $|S(n, R_T)| \geq |\Sigma'|^n$. Hence

$$h(T) \geq \lim_{n \to \infty} \frac{1}{n} \log |\Sigma'|^n = \log |\Sigma'|\,,$$

as we wanted to prove. ☐

This theorem can be readily applied to show that some particular examples of universal Turing machines exhibit positive topological entropy. For instance, the machine $T$ denoted as $UTM(6,4)$ in [28] has a transition function $\delta$ specified by the following table. The horizontal axis contains the states and the vertical axis contains the symbols. Here, $L$ and $R$ stand for $\delta_\varepsilon = -1$ and $\delta_\varepsilon = 1$ in our notation.

| $U_{6,4}$ | $u_1$ | $u_2$ | $u_3$ | $u_4$ | $u_5$ | $u_6$ |
|---|---|---|---|---|---|---|
| $g$ | $u_1bL$ | $u_1gR$ | $u_3bL$ | $u_2bR$ | $u_6bL$ | $u_4bL$ |
| $b$ | $u_1gL$ | $u_2gR$ | $u_5bL$ | $u_4gR$ | $u_6gR$ | $u_5gR$ |
| $\delta$ | $u_2cR$ | $u_2cR$ | $u_5\delta L$ | $u_4cR$ | $u_5\delta R$ | $u_1gR$ |
| $c$ | $u_1\delta L$ | $u_5gR$ | $u_3\delta L$ | $u_5cR$ | $u_3bL$ | halt |

FIGURE 1. Transition table of $UTM(6,4)$.

It is clear that $Q' \times \Sigma' := \{u_2\} \times \{b, \delta\}$ satisfies the hypotheses in the definition of a strongly regular Turing machine with $\varepsilon = 1$, and hence a straightforward application of Theorem 3 yields $h(T) \geq \log 2 > 0$, that is:

**Corollary 2.** *The universal Turing machine $UTM(6,4)$ has positive topological entropy.*

The same argument works when considering, for example, the reversible universal Turing machine $URTM(10,8)$ as introduced in [27, Section 7.3.2]:

**Corollary 3.** *The universal Turing machine $URTM(10,8)$ has positive topological entropy.*

Other examples of universal Turing machines that are strongly regular are $UTM(5,5), UTM(4,6)$ and $UTM(10,3)$ in [30], or $UTM(5,5), UTM(9,3)$ and $UTM(6,4)$ in [28]. The weakly universal Turing machines $WUTM(3,3)$ and $WUTM(2,4)$ in [35], or the famous Wolfram's weakly universal $(2,3)$ Turing machine are also strongly regular. We will later show that the universal Turing machine $UTM(15,2)$ in [28] or the weakly universal Turing machine $(6,2)$ in [35] are not strongly regular but are regular according to the definition in the following subsection (so, in particular, they have positive topological entropy).

4.2. **A generalized criterion for regular Turing machines.** In this subsection we establish a more general version of Theorem 3 that is also computable and implies that the Turing machine has positive topological entropy.

For this criterion we need to introduce some notation. As before, we denote tapes in $\Sigma^{\mathbb{Z}}$ using $t$ and $t_i$ denotes the $i^{\text{th}}$ symbol of $t$. We will construct a computable function

$$\phi : Q \times \Sigma \longrightarrow \{H, P\} \sqcup \{\pm 1\} \times Q,$$

which tells us whether the machine, with current state $q$ and reading the symbol $s$ at the zero position, will eventually (after perhaps some steps without shifting) shift to the right, to the left, or not shift at all before halting (H) or becoming periodic (P). More precisely, given a pair $(q,s) \in Q \times \Sigma$, if $\delta_\varepsilon((q,s)) = \pm 1$ and $\delta_Q((q,s)) \neq q_{halt}$, then

$$\phi(q,s) := (\delta_\varepsilon(q,s), \delta_Q(q,s)) \, .$$

If $\delta_Q(q,s) = q_{halt}$ then

$$\phi(q,s) := H \, .$$

Otherwise, setting $\delta_Q(q,s) = q_1$ and $\delta_\Sigma(q,s) = s_1$, if $\delta_Q(q_1,s_1) = q_{halt}$, then we define

$$\phi(q,s) := H \, ,$$

and we iterate this process. It is easy to check that for any pair $(q,s)$, only the following possibilities can occur for the aforementioned iteration:

(1) After $k$ steps of the machine without any shifting, we reach a configuration $(\tilde{q}, \tilde{t})$ such that $\delta_Q(\tilde{q}, \tilde{t}_0) = q_{halt}$. In this case $\phi(q,s) := H$.

(2) The iterates of the global transition function applied to $(q,t)$ never shift nor reach a halting configuration. Then for some $k$ we have $R_T^k(q,t)=(q,t)$ and the orbit becomes periodic. We define in this case $\phi(q,s):=P$.
(3) After $k$ steps without any shift the machine shifts to the left without halting. That is, a configuration of the form $(q_+,\tilde{t})$ with $\delta(q_+,\tilde{t}_0)=(q',s',1)$, $q'\neq q_{halt}$, is reached. In this case, we define $\phi(q,s):=(1,q')$.
(4) After $k$ steps without shifting the machine shifts to the right without halting. That is, a configuration of the form $(q_-,\tilde{t})$ with $\delta(q_-,\tilde{t}_0)=(q',s',-1)$ and $q'\neq q_{halt}$ is reached. In this case, we define $\phi(q,s):=(-1,q')$.

Of course, the integer $k$ depends on the pair $(q,s)$. We can define the function $\tau:Q\times\Sigma\to\mathbb{Z}$ as the function giving such an integer $k$.

**Definition 6** (Regular Turing machine)**.** A Turing machine $T$ is *regular* if, for some $\varepsilon\in\{-1,1\}$, there exist two different sequences

$$(q_1,s_1),...,(q_{m_1},s_{m_1}) \text{ and } (q_1',s_1'),...,(q_{m_2}',s_{m_2}')$$

of pairs in $Q\backslash\{q_{halt}\}\times\Sigma$, with $m_1\geq 2$, $m_2\geq 2$, such that $q_1=q_1'$, $\phi(q_i,s_i)=(\varepsilon,q_{i+1})$, $\phi(q_j',s_j')=(\varepsilon,q_{j+1}')$ for all $1\leq i\leq m_1-1$, $1\leq j\leq m_2-1$, and $\phi(q_{m_1},s_{m_1})=\phi(q_{m_2}',s_{m_2}')=(\varepsilon,q_1)$. We also require that none of the sequences is a concatenation of copies of the other sequence.

**Graph interpretation**. The regularity of a Turing machine can be easily understood in terms of two graphs that we can associate to $T$ using the function $\phi$. These graphs are different, although somewhat related, from the classical state diagram of the transition function of a Turing machine, see e.g. [33, Section 3.1]. For each $\varepsilon\in\{-1,1\}$ we can associate to $T$ a graph as follows: the vertices of the graph are the set of states of the machine. Given two vertices $q$ and $q'$, we define an edge oriented from $q$ to $q'$ for each $s\in\Sigma$ such that $\phi(q,s)=(\varepsilon,q')$. It is then obvious from the definition, that a Turing machine is regular if and only if for some $\varepsilon$ the corresponding graph contains two different oriented cycles with at least one common vertex.

The following examples show that there are universal Turing machines that are regular, but not strongly regular. The first example also illustrates the graph interpretation of a regular Turing machine.

**Example 1.** An example of a (weakly) universal Turing machine that is regular but not strongly regular is given by the $(6,2)$ machine in [35].

| $WU_{6,2}$ | $u_1$ | $u_2$ | $u_3$ | $u_4$ | $u_5$ | $u_6$ |
|---|---|---|---|---|---|---|
| $g$ | $u_10L$ | $u_60L$ | $u_20R$ | $u_51R$ | $u_41L$ | $u_11L$ |
| $b$ | $u_21L$ | $u_30L$ | $u_31L$ | $u_60R$ | $u_41R$ | $u_40R$ |

FIGURE 2. Transition table of $WUTM(6,2)$.

It is easy to check that it is not strongly regular. On the other hand, the sequences $(4,g),(5,b),(4,g)$ and $(4,b),(6,g),(4,b)$ satisfy the required properties for the machine to be regular. Figure 3 pictures the graph (as defined before) of the machine for $\varepsilon=1$. Notice how $u_4$ belongs to two different cycles.

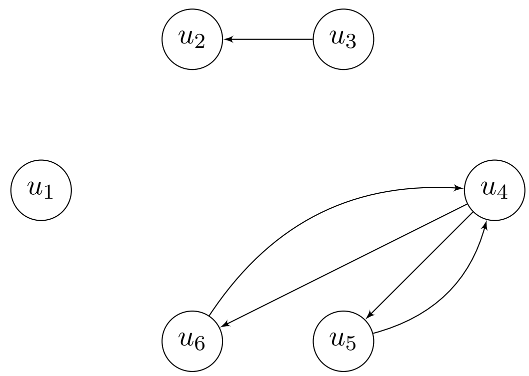


FIGURE 3. Graph of $WU_{6,2}$ for $\varepsilon=+1$.

**Example 2.** The universal Turing machine $UTM(15,2)$ in [28] is another example of universal Turing machine that is not strongly regular, but it is regular. Looking at the transition table [28, Table 16], one notices that the sequences $(u_4,c),(u_6,c),(u_4,c)$ and $(u_4,c),(u_6,b),(u_4,c)$ satisfy the necessary conditions in Definition 6.

As suggested by the name, any strongly regular Turing machine is regular. The graph interpretation is crucial to prove this property.

**Proposition 2.** *A strongly regular Turing machine is regular.*

*Proof.* Let $T$ be a strongly regular Turing machine with $\varepsilon=1$ (the other case is analogous), and let $Q'\subset Q\backslash\{q_{halt}\}$ and $\Sigma'\subset\Sigma$ be the subsets such that $\delta_\varepsilon|_{Q'\times\Sigma'}=1$ and $\delta_Q(Q'\times\Sigma')\subseteq Q'$ with $|\Sigma'|\geq 2$. Consider the associated graph defined above (with $\varepsilon=1$) and the subgraph $G$ given by the vertices $Q'$. It is clear that each vertex of $G$ is the origin of at least two edges, since any pair $(q,s)\in Q'\times\Sigma'$ has $\delta_\varepsilon(q,s)=1$ and $|\Sigma'|\geq 2$. Starting with a vertex $q_1\in Q'$ of $G$, we can iteratively move along an edge (without ever repeating that edge), following a sequence of vertices $q_i\in Q'$ and stop whenever we reach a $k$ such that $q_k=q_j$ for some $j<k$. This will necessarily happen, since after we have moved $|Q'|-1$ times, we will repeat a vertex. This way we find a cycle $C_1$, and we denote its set of vertices by $V_1$.

Consider the graph $G_1$ obtained by removing from $G$ the edges of $C_1$, and take a vertex $q\in Q'\setminus V_1$. Again, iteratively move along the edges of the graph starting at $q$. Notice that each vertex in $V_1$ is the origin of some edge, hence after at most $|Q|-1$ steps we will find another cycle $C_2$ with vertices $V_2$. If $V_1\cap V_2\neq\emptyset$, we are done. Otherwise $V_1$ and $V_2$ are disjoint, and we consider the graph $G_2$ obtained by removing from $G_1$ the edges of the cycle $C_2$. Repeating this process, if we do not find two cycles sharing a vertex, we end up with disjoint cycles $C_1, ..., C_N$ containing every vertex of $G$. If we remove all the edges of the cycles $C_1, ..., C_N$ from $G$, we obtain a graph $G'$ such that every vertex is the origin of at least one edge. We can apply the argument once more to $G'$, finding another cycle $C_0$, which necessarily intersects one of the $C_1, ..., C_N$, which completes the proof of the proposition. □

Finally, we are ready to prove the main theorem of this article.

**Theorem 4.** *A regular Turing machine has positive topological entropy.*

*Proof.* Let $T$ be a regular Turing machine, and let us assume that $\varepsilon = 1$, the other case being analogous. Define the integers

$$a_1 := 1 + \sum_{i=1}^{m_1} \tau(q_i, s_i), \quad a_2 := 1 + \sum_{i=1}^{m_2} \tau(q_i', s_i'),$$

and assume without any loss of generality that $a := a_1 \geq a_2$. Obviously, $a_1 > m_1$ and $a_2 > m_2$, so $a > \max\{m_1, m_2\}$. Consider integers of the form $n = ra$ with $r \in \mathbb{N}_0$. We claim that

$$|S(n, R_T)| \geq 2^r. \tag{14}$$

It is then easy to check that the topological entropy of $T$ is positive. Indeed, using Oprocha's formula we have:

$$\begin{aligned} h(T) &= \lim_{n\to\infty} \frac{1}{n} \log |S(n, R_T)| \\ &= \lim_{r\to\infty} \frac{1}{n} \log |S(n, R_T)| \\ &\geq \lim_{r\to\infty} \frac{1}{ra} r \log 2 \\ &= \frac{\log 2}{a} > 0\,. \end{aligned}$$

To see that the estimate (14) holds, we need to define some sequences of pairs in $Q \times \Sigma$. For each $(q_i, s_i)$, we define $(q_{i,1}, s_{i,1}) = \big(\delta_Q(q_i, s_i), \delta_\Sigma(q_i, s_i)\big)$ and then iteratively

$$(q_{i,j}, s_{i,j}) = \big(\delta_Q(q_{i,j-1}, s_{i,j-1}), \delta_\Sigma(q_{i,j-1}, s_{i,j-1})), \text{ for } j \in \{2, ..., \tau(q_i, s_i) - 1\}.$$

We define analogously $(q'_{i,j}, s'_{i,j})$. Consider the sequences

$$u_1 = \Big( (q_1, s_1), (q_{1,1}, s_{1,1}), (q_{1,2}, s_{1,2}), ..., (q_{1,\tau(q_1,s_1)-1}, s_{1,\tau(q_1,s_1)}), (q_2, s_2), \\ (q_{2,1}, s_{2,1}), ..., (q_{m_1-1,\tau(q_{m_1-1},s_{m_1-1})}), (q_{m_1}, s_{m_1}) \Big),$$

$$u_2 = \Big( (q_1', s_1'), (q_{1,1}', s_{1,1}'), (q_{1,2}', s_{1,2}'), ..., (q'_{1,\tau(q_1,s_1)-1}, s'_{1,\tau(q_1,s_1)}), (q_2', s_2'),$$
$$(q_{2,1}', s_{2,1}'), ..., (q'_{m_1-1,\tau(q_{m_1-1},s_{m_1-1})}, s'_{m_1-1,\tau(q_{m_1-1},s_{m_1-1})}), (q'_{m_1}, s'_{m_1}) \Big).$$

and any sequence of the form

$$u = v_1 \oplus v_2 \oplus ... \oplus v_r,$$

where $v_i$ is equal to either $u_1$ or $u_2$. This sequence has size at most $ra$, and there are $2^r$ possible choices which are all different thanks to the property that the two sequences of pairs $(q_i, s_i)$ and $(q_i', s_i')$ in $Q \times \Sigma$ satisfy that each one is not a concatenation of copies of the other one. For each possible $u$, consider the initial tape

$$t_u = ...00.t_1 t_2 ... t_{ra} 00...$$

constructed as follows. If $v_1 = u_1$, then the first $m_1$ symbols of the tape are $s_1, ..., s_{m_1}$. If $v_1 = u_2$, then instead the first $m_2$ symbols of the tape are $s_1', ..., s_{m_2}'$. The next group of symbols is determined by $v_2$, if $v_2 = u_1$ then the next symbols are $s_1, ..., s_{m_1}$, and if $v_2 = u_2$ then the next symbols are $s_1', ..., s_{m_2}'$. We do this up to $v_r$, and this determines at most $ra$ symbols. We can fill the rest of symbols up to $t_{ra}$ with zeroes, for instance. By construction, initializing the machine with the configuration $x = (q_1, t_u)$, it is easy to check that $(R_T^i(x)_q, R_T^i(x)_0)$ follows sequentially the pairs in $u$, and after that it possibly runs through some other pairs in $Q \times \Sigma$. This shows that for each $u$ there is (at least) a distinct element in $S(n, R_T)$, which proves that the bound (14) holds, as we wanted to show. □

Notice that, using the graph interpretation of a regular Turing machine, the proof of Theorem 4 yields that the number of paths from the vertex that belong to two different cycles grows exponentially with respect to the length of the path. Heuristically, this might be interpreted as a hint that the topological entropy is positive, as rigorously established in Theorem 4

We believe that the hypothesis that implies positive topological entropy, in our case the definition of regular Turing machine, can probably be relaxed a bit at the cost of adding quite some more technical details and definitions. We have not found any example in the literature of a universal Turing machine that is not regular, although artificial examples can certainly be constructed, as explained in Appendix A.

## 5. Topological entropy of Turing complete area-preserving diffeomorphisms and Euler flows

In this section we use our main theorem to construct Turing complete dynamical systems with positive topological entropy. We first recall the definition of a Turing complete dynamics $X$ on a topological space $M$:

**Definition 7** (Turing complete dynamical system)**.** A dynamical system $X$ on $M$ is *Turing complete* if there is a universal Turing machine $T_u$ such that for any input $t_{in}$ of $T_u$, there is a computable point $p \in M$ and a computable open set

$U \subset M$ such that $\mathrm{Orb}_X(p) \cap U \neq \emptyset$ if and only if the machine $T_u$ halts with input $t_{in}$.

In several examples of Turing complete systems in the literature, however, stronger properties are satisfied. For instance, for a diffeomorphism $f : M \to M$ of a manifold, one might require that there is a compact invariant subset $C \subset M$ such that $f|_C$ is conjugate or semiconjugate to the global transition function of the Turing machine $T_u$. The following lemma follows from a combination of [7, Lemma 4.5] and [7, Proposition 5.1]:

**Lemma 6.** *Let $T$ be a reversible Turing machine whose global transition function has been extended to halting configurations via the extension of the transition function $\delta(q_{halt}, t) = (q_0, t, 0)$ for all $t \in \Sigma$. Then there exists a bijective generalized shift $\Delta$ that is conjugate to $R_T$, and a smooth area-preserving diffeomorphism $\varphi : D \longrightarrow D$ of a disk $D$ (of radius larger than one) that is the identity near the boundary and whose restriction to the square Cantor set $C^2 \subset D$ is conjugate to $\Delta$ by the homeomorphism $e$ in Equation* (10)*.*

It was shown in [7, Corollary 3.2] that the diffeomorphism $\varphi$ can be realized as the first-return map on a disk-like transverse section of a stationary solution to the Euler equations for some metric on any compact three-manifold $M$, which yields a Turing complete stationary fluid flow [7, Theorem 6.1]. The main application of Theorem 1 is that the Turing complete Euler flows constructed in [7] always have positive topological entropy whenever the simulated universal Turing machine is regular.

**Corollary 4.** *If $T$ is a regular reversible Turing machine then its associated diffeomorphism of the disk $\varphi$ and the steady Euler flows on a compact three-manifold $M$ constructed in* [7] *have positive topological entropy. If $T$ is universal, then $\varphi$ exhibits a compact chaotic invariant set $K$ homeomorphic to the square Cantor set so that $\varphi|_K$ is Turing complete.*

*Proof.* Given Theorem 4, the Turing machine $T$ has positive topological entropy. Its associated generalized shift $\Delta$ has positive topological entropy too by Proposition 1. The identification of the space of sequences of the generalized shift with a square Cantor set $C \subset [0, 1]^2$ via the homeomorphism (10) implies that $\Delta$ induces a map $\tilde{\Delta} : C \to C$. By [7, Proposition 5.1] there exists an area-preserving diffeomorphism $\varphi : D \to D$ of a disk $D$ strictly containing the unit square whose restriction to $C$, which is an invariant set, coincides with $\tilde{\Delta}$. Hence, the topological entropy of $\varphi$ is necessarily positive. The compact chaotic invariant set is $K \equiv C$, and $\varphi|_K$ is Turing complete whenever $T$ is universal. The stationary Euler flow in [7, Theorem 6.1] admits a transverse disk where the first-return map is conjugate to $\varphi$, and therefore it has positive topological entropy too. □

ACKNOWLEDGEMENTS

The authors are very grateful to Leonid Polterovich for suggesting to study the topological entropy of Turing complete dynamical systems. We also acknowledge

Ville Salo for the improvements and suggestions on a previous version of this work, and for writing Appendix A.

## References

[1] R.L. Adler, A.G. Konheim, M.H. McAndrew. *Topological entropy.* Trans. Amer. Math. Soc. 114 (1965), 309–319.
[2] V. D. Blondel, J. Cassaigne, C. Nichitiu. *On the presence of periodic configurations in Turing machines and in counter machines.* Theor. Comput. Sci. 289 (2002), 573–590.
[3] R. Berger. *The undecidability of the domino problem.* Mem. Amer. Math. Soc. 66 (1966), 1–72.
[4] R. Bowen. *Entropy for groups endomorphisms and homogeneous spaces.* Trans. Amer. Math. Soc. 153 (1971), 401–414.
[5] R. Cardona, E. Miranda, D. Peralta-Salas. *Turing universality of the incompressible Euler equations and a conjecture of Moore.* Int. Math. Res. Notices 2022 (2022), 18092–18109.
[6] R. Cardona, E. Miranda, D. Peralta-Salas. *Computability and Beltrami fields in Euclidean space.* J. Math. Pures Appl. 169 (2023), 50–81.
[7] R. Cardona, E. Miranda, D. Peralta-Salas, F. Presas. *Constructing Turing complete Euler flows in dimension 3.* Proc. Natl. Acad. Sci. 118 (2021), 19(1–9).
[8] R. Cardona, E. Miranda, D. Peralta-Salas, F. Presas. *Universality of Euler flows and flexibility of Reeb embeddings.* Adv. Math. 428 (2023), 109142.
[9] E. Dinaburg. *A connection between various entropy characterizations of dynamical systems.* Izv. Akad. Nauk SSSR 35 (1971), 324–366.
[10] A. Gajardo, N. Ollinger, R. Torres-Avilés. *Some undecidable problems about the trace-subshift associated to a Turing machine.* Discrete Mathematics & Theoretical Computer Science, 17: Automata, Logic and Semantics (2015).
[11] S. Gangloff, A. Herrera, C. Rojas, M. Sablik. *Computability of topological entropy: From general systems to transformations on Cantor sets and the interval.* Discrete Cont. Dyn. Sys. 40 (2020), 4259–4286.
[12] S. Gangloff, A. Herrera, C. Rojas, M. Sablik. *On the computability properties of topological entropy: a general approach.* Preprint (2019) arXiv:1906.01745.
[13] D.S. Graca, M.L. Campagnolo, J. Buescu. *Computability with polynomial differential equations.* Adv. Appl. Math. 40 (2008), 330–349.
[14] D.S. Graca, N. Zhong. *Analytic one-dimensional maps and two-dimensional ordinary differential equations can robustly simulate Turing machines.* Computability 12 (2023), 117–144.
[15] P. Hooper. *The undecidability of the Turing machine immortality problem.* J. Symbolic Logic 31 (1966), 219–234.
[16] E. Jeandel. *Computability of the entropy of one-tape Turing machines.* 31st International Symposium on Theoretical Aspects of Computer Science, 2014.
[17] A. Katok, B. Hasselblatt, *Introduction to the Modern Theory of Dynamical Systems*, Cambridge Univ. Press, New York, 1995.
[18] J. Kari, N. Ollinger. *Periodicity and immortality in reversible computing.* International Symposium on Mathematical Foundations of Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.
[19] P. Koiran, M. Cosnard, M. Garzon. *Computability with low-dimensional dynamical systems.* Theor. Comput. Sci. 132 (1994), 113–128.
[20] P. Koiran, C. Moore. *Closed-form analytic maps in one and two dimensions can simulate universal Turing machines.* Theor. Comput. Sci. 210 (1999), 217–223.
[21] P. Kůrka. *On topological dynamics of Turing machines.* Theor. Comput. Sci. 174 (1997), 203–216.

[22] M. Minsky. *A 6-symbol 7-state universal Turing machine*. MIT Lincoln Laboratory Report G-0027, 1960.
[23] M. Minsky. *Recursive unsolvability of Post's problem of "tag" and other topics in theory of Turing machines*. Ann. of Math. 74 (1961), 437–455.
[24] Y. Mitsumatsu, D. Peralta-Salas, R. Slobodeanu. *On the existence of critical compatible metrics on contact 3-manifolds*. Preprint (2023) arXiv:2311.15833.
[25] C. Moore. *Unpredictability and undecidability in dynamical systems*. Phys. Rev. Lett. 64 (1990), 2354–2357.
[26] C. Moore. *Generalized shifts: unpredictability and undecidability in dynamical systems*. Nonlinearity 4 (1991), 199–230.
[27] K. Morita. *Theory of reversible computing*. Springer Japan, 2017.
[28] T. Neary, D. Woods. *Four small universal Turing machines*. Fund. Inform. 91 (2009), 123–144.
[29] P. Oprocha. *On entropy and Turing machine with moving tape dynamical model*. Nonlinearity 19 (2006) 2475–2487.
[30] Y. Rogozhin. *Small universal Turing machines*. Theor. Comput. Sci. 168 (1996), 215–240.
[31] C.E. Shannon. *A universal Turing machine with two internal states*. Automata Studies, pp. 157–165, Ed. C.E. Shannon and J. McCarthy, Princeton Univ. Press, Princeton, 1956.
[32] S. Simpson. *Symbolic dynamics: entropy = dimension = complexity*. Theor. Comput. Sys. 56 (2015), 527–543.
[33] M. Sipser. *Introduction to the theory of computation* (3rd ed.). International Thomson Publishing, 2012.
[34] T. Tao. *On the universality of potential well dynamics*. Dyn. PDE 14 (2017), 219–238.
[35] D. Woods, T. Neary. *Small semi-weakly universal Turing machines*. In International Conference on Machines, Computations, and Universality, September 2007, pp. 303-315. Berlin, Heidelberg: Springer Berlin Heidelberg.

## Appendix A. A universal Turing machine with zero topological entropy (by Ville Salo, University of Turku, Finland)

The goal of this appendix is to construct a universal Turing machine with zero topological entropy. In particular, this machine is not regular in the sense of Definition 6. The construction is based directly on the universal counter machines of Minsky [23]. After this construction, we explain how to obtain another proof (of a stronger result) from a more involved construction of Hooper [15] (or Kari and Ollinger [20]).

We start by recalling Rogozhin's definition [30] of a universal Turing machine; compare with Definition 3 in the main text. First, for a Turing machine $M$, let $B_M$ be the (computable) set of its finite-support configurations. An important technical point is that for partial computable functions (recall that a partial computable function is just any function that a computer, e.g. a Turing machine, can compute) operating on such configurations, we want $B_M$ to be represented so that the boundaries of the support are explicitly visible (although they are not explicitly visible to the Turing machine $M$); a natural way is to code $B_M$ as instantaneous descriptions, i.e., as finite words. Let $B$ be a disjoint union of all the sets $B_M$. If $a, b \in B_M$, write $a \Rightarrow_M b$ for individual computation steps of $M$, and write $\Rightarrow_M^*$ for the transitive closure. Define a partial function by letting

$\Phi_M(a) = b$ if $a \Rightarrow_M^* b$ and $b$ is a halting computation, and leave $\Phi_M(a)$ undefined if $M$ never halts when started from $a$.

It is well-known that there exists a universal partial computable function $\phi : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, in the sense that $\phi(n, x) = \psi_n(x)$ for all $n, x \in \mathbb{N}$, where $\psi_n$ is the $n$th partial computable function of type $\psi_n : \mathbb{N} \to \mathbb{N}$ (under some Gödel numbering).

**Definition 8.** The Turing machine $U$ is *universal* if there exist total computable functions $e : \mathbb{N} \times \mathbb{N} \to B_M$ and $d : B_M \to \mathbb{N}$ such that $d(\Phi_U(e(n, x))) = \phi(n, x)$ for all $n, x$.

The equality of two partial functions here means they are defined on precisely the same inputs. The function $e$ is the *encoding function* which is given the Gödel number of a partial computable function, and an input, and it encodes it as a Turing machine tape. The *decoding function* $d$ then decodes the halting configuration into a number.

**Remark 2.** This definition has the drawback that since universal Turing machines simulate arbitrary computation on natural numbers, it is difficult to sensibly discuss complexity-theoretic (let alone algorithmic) issues, such as how much of the computation is actually performed by $d, e$. This definition only requires that it is $U$ that passes the "halting problem barrier". There exist universal Turing machines $U$ which can simulate an arbitrary Turing machine $M$ in such a way that $d, e$ just output a substitutive encoding of a configurations of $M$ (with the transition table of $M$ output at regular intervals), and $U$ simulates $M$ with only linear slow-down. This implies positive entropy. However, it seems likely that there are interesting encoding schemes that are "more efficient" than ours, yet allow for zero entropy universality.

**Theorem 5.** *There exists a universal Turing machine which has zero topological entropy.*

*Proof.* We recall the definition of a 2-counter machine. This is $C = (Q, q_0, q_{\mathrm{halt}}, \delta)$ where $\delta : (Q \setminus \{q_{\mathrm{halt}}\}) \times \{0, +\}^2 \to Q \times \{-1, 0, +1\}^2$ is the transition relation. The machine defines a partial transition function on $Q \times \mathbb{N} \times \mathbb{N}$ (defined if and only if the state is not $q_{\mathrm{halt}}$. The interpretation of $(q, m, n)$ is that the machine is in state $q$, and the current counter values are $m, n$.

The interpretation of $\delta(q, (a_1, a_2)) = (q', b_1, b_2)$ is that if the current state is $q$ and $a_i = 0$ iff the $i$th counter has zero value, then we step into state $q'$ and add $(b_1, b_2)$ to the counter values (we require that $a_i = 0 \implies b_i \neq -1$). Write $t \Rightarrow_C t'$ if there is a one-step computation of $C$ from $t \in Q \times \mathbb{N} \times \mathbb{N}$ to $t' \in Q \times \mathbb{N} \times \mathbb{N}$, and as in the case of Turing machines define a partial function $\Phi_C(t) = t'$ if $t \Rightarrow_C^* t'$ and $t'$ is halting (i.e. $t' = (q_{\mathrm{halt}}, m, n)$ for some $m, n$).

It is a result of Minsky [23] that a 2-counter machine can simulate an arbitrary Turing machine, in the sense that for any Turing machine $M$ with states $Q$, we can find a 2-counter machine with states $Q' \supset Q$ and the same halting state $q_{\mathrm{halt}}$, and total computable functions $e : B_M \to Q \times \mathbb{N}$ and $d : Q \times \mathbb{N} \to B_M$ such that if started from state $(e(t), 0)$ with $t \in B_M$, the next time we are in a state $(q', m', 0) \in Q \times \mathbb{N} \times \mathbb{N}$ (i.e. the next time the second counter contains 0, and the state is in the subset $Q$ of $Q'$), we have $d(q', m') = t'$ such that $t \Rightarrow_M t'$).

Next, for any counter machine $C$ with states $Q$ and transitions $\delta$, it is again possible to construct a Turing machine $M$ with alphabet $\{@, 0, 1\}$ and states $Q' = Q \sqcup Q''$ which simulates the counter machine in the following sense: If started on the configuration

$$^{\omega}010^m@0^n10^{\omega}$$

with the head on the @-symbol in state $q \in Q$, then when in the sequence of $\Rightarrow_M$-steps we next enter a configuration of the form

$$^{\omega}010^{m'}@0^{n'}10^{\omega}$$

with the head on the @-symbol in some state $q' \in Q$, we have $(q, m, n) \Rightarrow_C (q', m', n')$. We call this the *simulation property.*

The way this is done is simply that the Turing machine performs a back and forth sweep both ways to check which of the counters have zero value, and then performs new sweeps in order to update them, according to the transition function of $C$.

We describe a naive concrete implementation. One can use a state set of the form $Q \sqcup (Q \times \{0, +\}^2 \times X) \sqcup \{\perp\}$, where $Q$ simulated the states of the counter machine, $\perp$ is a fail state, and the elements of $Q \times \{0, +\}^2 \times X$ are interpreted as follows: $Q$ remembers the counter machine state, $\{0, +\}^2$ is a finite amount of memory for storing values of counters, and $X$ is used to remember what we are doing.

Specifically we can pick

$$\begin{aligned} X = \{&\text{left check}, \text{left check return}, \text{right check}, \text{right check return}, \text{left update}, \\ &\text{left drop}, \text{left update return}, \text{right update}, \text{right drop}, \text{right update return}\}. \end{aligned}$$

Initially, when started in a state simulating a state of the counter machine, our machine does not know what the counter values are, and should perform two sweeps in order to calculate them. This can be done as follows:

$$\begin{aligned} \delta'(q, @) &= ((q, 0, 0, \text{left check}), @, -1) \\ \delta'((q, a, 0, \text{left check}), 0) &= ((q, +, 0, \text{left check}), 0, -1) \\ \delta'((q, a, 0, \text{left check}), 1) &= ((q, a, 0, \text{left check return}), 1, 1) \\ \delta'((q, a, 0, \text{left check return}), 0) &= ((q, a, 0, \text{left check return}), 0, 1) \\ \delta'((q, a, 0, \text{left check return}), @) &= ((q, a, 0, \text{right check}), @, 1) \\ \delta'((q, a, b, \text{right check}), 0) &= ((q, a, +, \text{right check}), 0, 1) \\ \delta'((q, a, b, \text{right check}), 1) &= ((q, a, b, \text{right check return}), 1, -1) \\ \delta'((q, a, b, \text{right check return}), 0) &= ((q, a, b, \text{right check return}), 0, -1) \\ \delta'((q, a, b, \text{right check return}), @) &= ((q, a, b, \text{left update}), @, -1) \end{aligned}$$

At this point, the state is expected to be $(q, a, b, \text{left update})$ where $q$ is the simulated counter machine state, and $a, b \in \{0, +\}$ are the information about whether the counters are zero or positive. Next, we should update the counters. Note that decrementing the left counter means simply changing the 1 on the left to 0, and rewriting it on the right (by using the drop state), and similarly for other counter updates; and symmetrically for the right counter.

Specifically, assume the counter machine transitions as $\delta(q, a, b) = (q', c, d)$. Then the added transitions can be taken to be

$$\begin{aligned}
\delta'((q, a, b, \text{left update}), 0) &= ((q, a, b, \text{left update}), 0, -1) \\
\delta'((q, a, b, \text{left update}), 1) &= ((q, a, b, \text{left drop}), 0, -c) \\
\delta'((q, a, b, \text{left drop}), 0) &= ((q, a, b, \text{left update return}), 1, 1) \\
\delta'((q, a, b, \text{left update return}), 0) &= ((q, a, b, \text{left update return}), 0, 1) \\
\delta'((q, a, b, \text{left update return}), @) &= ((q, a, b, \text{right update}), @, 1) \\
\delta'((q, a, b, \text{right update}), 0) &= ((q, a, b, \text{right update}), 0, 1) \\
\delta'((q, a, b, \text{right update}), 1) &= ((q, a, b, \text{right drop}), 0, d) \\
\delta'((q, a, b, \text{right drop}), 0) &= ((q, a, b, \text{right update return}), 1, -1) \\
\delta'((q, a, b, \text{right update return}), 0) &= ((q, a, b, \text{right update return}), 0, -1) \\
\delta'((q, a, b, \text{right update return}), @) &= (q', @, 0)
\end{aligned}$$

It is clear that no matter what the other transitions are, this realizes the counter machine simulation correctly: one can exactly calculate the sequence of moves performed on the configuration ${}^{\omega}010^m@0^n10^{\omega}$, and no unexpected situations can arise (note that by assumption, our counter machines never try to decrement a counter with value zero).

Now we let all other transitions enter the state $\perp$, and in this state, loop for ever without moving. We claim that then the machine has zero entropy. For this, we analyze a computation of the machine on an arbitrary configuration, for $N$ steps.

Observe that when we defined $X$ above, we listed it in a particular order. Our machine has the property that when it is in a state outside $Q \cup \{\perp\}$, the $X$-component of the state will evolve in this order, until the machine has either entered $\perp$ (and is in an infinite loop without moving), or is back to a state of $Q$, necessarily on top of the symbol @. From a quick look at the transitions we see that it moves to the next state of $X$ whenever it sees any nonzero symbol on the tape. Note that this implies that this initial segment of the computation can be described by at most four numbers and some constant information, thus has a description with $\log O(N^4) = O(\log N)$ bits.

Once we are in state $Q$, the computation in fact simulates the counter machine exactly as above, or enters the state $\perp$: The machine will look for the next 1 to the left of @, then for the next 1 to the right, and then update their positions.

If it never finds such 1, or runs into another @-symbol, it enters state $\perp$; or if it tries to move 1 further away from @, then that position must contain 0 or it enters state $\perp$.

Recall that positive entropy of a Turing machine implies positive entropy for the traces $S(N,U)$ ([29, 16]), and that positive entropy for a subshift (we apply this to the subshift whose forbidden words are finite words that do not appear in the words $S(N,U)$) implies that some infinite configuration has linear Kolmogorov complexity in all prefixes [32, 16]. Thus if we had positive entropy for the Turing machine, then some words in $S(N,U)$ would require at least $CN$ bits to describe for all large enough $N$. Suppose this is the case, and we show a contradiction by compressing them strictly more efficiently.

By the explanation above, any computation can be compressed by remembering $O(\log N)$ bits; then the part where we simulate the counter machine; then another $O(\log N)$ bit compressible suffix describing a computation that does not cycle through all of $X$; and finally possibly we remember a number indicating how much time we spend in state $\perp$, again requiring at most $O(\log N)$ bits.

We now analyze the part where we simulate the counter machine, as it is the only possible source of a linear amount of Kolmogorov complexity. Let $(q_1, m_1, n_1), (q_2, m_2, n_2), \ldots$ be the sequence of simulated states and counter values encountered during this simulation part. Note that we must have $N \geq \sum_i (m_i + n_i)$, as the machine certainly spends more than $m_i + n_i$ steps to read counter values $m_i$ and $n_i$ encoded in the distances of 1s from the @-symbol, and to update them (recall that we always read these values whether or not the machine actually "needs" to know their values).

We can compress the information about this sequence into $B \sum_i \log(m_i + n_i)$ bits for some constant $B$, by simply writing down the numbers in binary (more naturally we get $\log(m_i) + \log(n_i)$, but $\log(m_i) + \log(n_i) \leq 2\log(m_i + n_i)$ and the 2 disappears into $B$). Let $I$ be the set of $i$ such that $\log(m_i + n_i) < \frac{C}{2B}(m_i + n_i)$ Note that this is true whenever $m_i + n_i > D$ for some constant $D$. Let $J$ be the complement of $I$ (among indices of the $(q_i, m_i, n_i)$).

If we do not have a repetition among the $(q_i, m_i, n_i)$, then have the (rough) upper bound

$$\begin{aligned} B \sum_i \log(m_i + n_i) &\leq C \sum_{i \in I} (m_i + n_i)/2 + B \sum_{i \in J} \log D \\ &\leq CN/2 + B|Q|D^2 \log D \end{aligned}$$

where $B, |Q|, D$ do not depend on $N$, so this is far smaller than $CN$ for large $N$, even together with the initial and final parts of the computation that took $O(\log N)$ bits to compress.

On the other hand, computations with repeated $(q_i, m_i, n_i)$ are periodic, and even easier to compress.

This contradiction proves that there cannot be a linear lower bound on the compressibility, which finally concludes the proof of zero entropy. $\square$

In the beginning of this appendix, we mentioned that more is known. We briefly outline this. A Turing machine admits a notion of *speed*, namely one calculates the maximal offset by which the head can move in $n$ steps, observes that this quantity is subadditive and takes a normalized limit using Fekete's lemma. It is easy to show that zero speed implies zero entropy.

In 1969, Hooper proved in [15] (see [20] for a reversible version with an arguably easier proof) that given a Turing machine, it is undecidable whether it admits configurations where the machine never halts. If a Turing machine halts on every configuration, then a simple compactness argument shows that there is a bound on the number of steps it takes to halt. Thus, the undecidability must come from computations that do not halt.

Thus, Hooper at least had to show that one can perform universal computation with a Turing machine such that there are no situations where it is easy to prove that the machine never halts (on infinite configurations). One such situation is an infinite "search" for a symbol. In all direct simulations (and definitely in the counter machine simulation we performed above), there are such infinite searches, and due to compactness of the configuration space, it is tempting to think that they are necessary. They are not, and the genius trick of Hooper was to show that one can trick compactness by starting computations recursively, so that even though there are infinite searches, there are other searches between them. This is analogous to Berger's proof in [3] of the undecidability of the domino problem.

It was later clarified by Jeandel that Hooper was in a sense literally fighting positive speed: [16] shows that if a Turing machine has positive speed (resp. positive entropy), then this can be proved in ZF, by showing that any such machine satisfies a type of generalized version of the notion of regularity studied in the present paper.

Thus, our conclusion is that at least infinitely many of Hooper's machines must have zero speed, thus zero entropy. Since they involve an undecidability problem, one should expect them to involve universal computation, and indeed Hooper's machines have literally the simulation property we described above (except the encoding is somewhat different, and there are many intermediate configurations where multiple @-symbols appear, due to the recursive computations started at all times).

Unfortunately, Hooper was not explicitly concerned with universal Turing machines, nor explicitly discusses speed or entropy, and thus we did not find it easy to use his results as a black box to prove even the existence of a zero entropy universal Turing machine. Nevertheless, there is no doubt that his construction implies that zero speed universal Turing machines exist, and as we have tried to argue here this is morally an automatic consequence of his result.

We state the stronger reversible statement: A reversible variant of Hooper's construction is given in [20]. This is also a direct simulation of a reversible counter machine, and such a machine can simulate an arbitrary (not necessarily reversible)

Turing machine up to a computable encoding. From the construction, one thus obtains the following result:

**Theorem 6.** *There exists a universal Turing machine which is reversible and has zero speed. In particular, it has zero topological entropy.*

Renzo Bruera, Universitat Politècnica de Catalunya, Departament de Matemàtiques, Avinguda Diagonal, 647, 08028 Barcelona. *e-mail: renzo.bruera@upc.edu*

Robert Cardona, Departament de Matemàtiques i Informàtica, Universitat de Barcelona, Gran Via de Les Corts Catalanes 585, 08007 Barcelona, Spain. *e-mail: robert.cardona@ub.edu*

Eva Miranda, Laboratory of Geometry and Dynamical Systems & Institut de Matemàtiques de la UPC-BarcelonaTech (IMTech), Universitat Politècnica de Catalunya, Avinguda del Doctor Marañon 44-50, 08028, Barcelona, CRM Centre de Recerca Matemàtica, Campus de Bellaterra Edifici C, 08193 Bellaterra, Barcelona *e-mail: eva.miranda@upc.edu*

Daniel Peralta-Salas, Instituto de Ciencias Matemáticas, Consejo Superior de Investigaciones Científicas, 28049 Madrid, Spain. *e-mail: dperalta@icmat.es*