The transitivity problem of Turing machines*

Anahí Gajardo[1], Nicolas Ollinger[2], and Rodrigo Torres-Avilés[1]

[1] Departamento de Ingeniería Matemática and Centro de Investigación en Ingeniería Matemática (CI^{2}MA), Universidad de Concepción, Centro de Modelamiento Matemático (CMM), Universidad de Chile, Casilla 160-C, Concepción, Chile
{anahi,rtorres}@ing-mat.udec.cl

[2] Univ. Orléans, INSA Centre Val de Loire, LIFO EA 4022, FR-45067 Orléans, France
nicolas.ollinger@univ-orleans.fr

Abstract. A Turing machine is topologically transitive if every partial configuration — that is a state, a head position, plus a finite portion of the tape — can reach any other partial configuration, provided that they are completed into proper configurations. We study topological transitivity in the dynamical system models of Turing machines with moving head, moving tape and for the trace-shift and we prove its undecidability. We further study minimality, the property of every configuration reaching every partial configuration.

Keywords: reversible computing, discrete dynamical systems, symbolic dynamics, topological dynamics, computability

1 Introduction

Turing machines [16] provide a simple mechanical model of computation: an agent equipped with a finite internal memory moves over a tape full of symbols which can read and modify it; its movement as well as its interaction with its memory and the tape is deterministically governed by a transition rule. Eventually, this rule may carry the machine to a halting configuration. A computation can be performed by initializing the tape with an input word and decoding the output on the halting tape. The *universality* of the model, asserted by the Church-Turing thesis, the *undecidability* of the *Halting Problem* and its generalizations like *Rice's theorem* provide convenient tools to assert that any kind of system, even "natural" systems like lattice gases dynamics [12], able to perform universal computation has a particular strong form of unpredictability: even if the complete information about the state of the system is available, determining its asymptotic behavior is impossible. On the other hand, qualitative properties of the system can be, and frequently are, undecidable. Even if all the system parameters are given, there is no general procedure to determine whether the system satisfies a given property.

* This work has been supported by ECOS Sud – CONICYT project C12E05 and partially supported by FONDECYT#1140684 and BASAL project CMM, Universidad de Chile, by Centro de Investigación en Ingeniería Matemática (CI^{2}MA).

This is the case, for example, of dynamics over a piecewise constant vector field in dimension three [1], where the *Reachability Problem*, *i.e.*, the problem of determining whether one region is reachable from another, is undecidable. Proper dynamical properties, as *periodicity*, *sentitivity*, *transitivity* and *limit set*, have also been proved to be undecidable for some classes of systems, in particular in the context of *cellular automata* [7, 8, 11]. The main difficulty to establish such result is that, contrarily to the Turing machine in the context of the Halting Problem, dynamical properties describe the system behavior over the whole set of configurations. The property studied in this paper, the transitivity of a Turing machine, is of the same nature.

The appropriate point of view is to study Turing machines as proper dynamical systems as introduced by Moore [13] and further developed by Kůrka [9] who formalized two associated topologies and establishes several properties for them. Following that trend, several dynamical properties of Turing machines were recently studied: periodicity [2, 3, 8], entropy [6, 14] and equicontinuity [4]. Here we focus on *topological transitivity*: the existence of a point whose orbit passes close to every other point of the space. Moreover, "most" of the points have this characteristic. This gives to the system some kind of *homogeneity* in the sense that some point wise properties become global when transitivity is present. For example, in a transitive system there is a dichotomy between *almost-equicontinuity* and *sensitivity*: either almost every point is stable or every point is sensitive. On the other hand, transitivity is included in the most accepted definitions of "chaos".

In the context of Turing machines, transitivity imply that every "local configuration" is reachable from every other; but here the meaning of "local" is ambiguous and it depends on the topology that we choose for our model. Kůrka [9] proposes two different topologies for Turing machines, one gives preponderance to the cell contents around the head (Turing Moving Tape model (TMT)) and the other focuses on the cells that surround the position "0" of the tape (Turing Moving Head model (TMH)). In this paper, we consider both models, as well as the column factor of the TMT model: the *trace-shift*.

We establish that transitivity is undecidable in all the three considered models. The proof is performed by reduction from the Reachability Problem, through a technique that consists into *embedding* a machine inside another in such a way the first avoids transitivity of the second when reachability is satisfied. This technique has proved to be very flexible and useful. It was also used in [3] to prove the undecidability of the existence of periodic points in the TMT model. It is also used here to prove the undecidability of another important dynamical property: *minimality*.

2 Definitions

2.1 Turing machines

A *Turing machine* $\mathcal{M}$ is a triple (Q, Σ, δ) where Q is the finite *set of states*, Σ is the finite *alphabet* and $\delta : Q \times \Sigma \to Q \times \Sigma \times \{-1, 0, 1\}$ is the partial *transition function* of $\mathcal{M}$. The machine is *complete* if δ is a total function.

A *configuration* of $\mathcal{M}$ is a triple (s, c, p) where $s \in Q$ is the state of the machine, $c \in \Sigma^{\mathbb{Z}}$ is the content of the *bi-infinite tape* and $p \in \mathbb{Z}$ is the position of the head on the tape. Instead of (s, c, p), we might use the following convenient notation: $(\overset{\cdots\ c_{-1}\ \cdot c_0\ \cdots\ c_p\ c_{p+1}\ \cdots}{s})$, where the *dot* before c_0 is there to distinguish the coordinate 0. A configuration (s, c, p) is *halting* if (s, c_p) is a *halting pair*, that is if $\delta(s, c_p)$ is undefined.

A *transition* of the machine transforms a non-halting configuration (s, c, p) into the configuration $(t, c', p + d)$ where $\delta(s, c_p) = (t, a, d)$ and c' is equal to c in every position except for $c'_p = a$. The transition function is denoted by $\vdash$ and its iteration zero or more times by $\vdash^*$.

A configuration is *periodic* if the machine returns to that same configuration after a finite number of transitions. The machine is *aperiodic* if it has no periodic configuration.

The machine is *surjective* if every configuration can be obtained in one transition from at least one configuration and *injective* if every configuration can be obtained in one transition from at most one configuration, called its preimage. In a complete machine, surjectivity is equivalent to injectivity.

Every injective machine is a *reversible* machine: it can be assigned a reverse. Indeed, an injective machine is characterized by a pair (ρ, μ), where $\rho : Q \times \Sigma \to Q \times \Sigma$ is a partial injective function and $\mu : Q \to \{-1, 0, 1\}$, such that $\delta(s, a) = (t, b, \mu(t))$ where $\rho(s, a) = (t, b)$ for all state s and symbol a. The *reverse machine* $\mathcal{M}^{-1}$ is the reversible Turing machine (Q, Σ, δ^{-1}) where $\delta^{-1}(t, b) = (s, a, -\mu(s))$ for all s, a, t, b such that $\rho(s, a) = (t, b)$. For every non-halting configuration (s, c, p) transformed by $\mathcal{M}$ into the configuration (t, c', p'), the configuration $(t, c', p' - \mu(t))$ is transformed by $\mathcal{M}^{-1}$ into the configuration $(s, c, p - \mu(s))$. A *starting configuration* of $\mathcal{M}$ is a halting configuration of $\mathcal{M}^{-1}$ and a *starting pair* of $\mathcal{M}$ is a halting pair of $\mathcal{M}^{-1}$. A reversible Turing machine has as many starting pairs as halting pairs.

A *partial configuration* is a configuration (s, c, p) where c is a partial function defined only on a finite and connected portion of the tape. Transitions extend to partial configurations and are defined only when the head is pointing on a defined symbol. A configuration *completes* a partial configuration if they coincide on the intersection of their domains.

Reachability problem *Given a Turing machine and a pair of states, decide if, starting from a configuration in the first state, the machine can reach a configuration in the second state after a finite number of transitions.*

The reachability problem is known to be Σ_1^0-complete even if the input is restricted to aperiodic reversible Turing machines [8] from a starting pair to a halting pair. Moreover, one can fix the alphabet of the machine to be binary.

Transitivity problem *Given a Turing machine, decide if, for every pair of partial configurations, one can complete the first partial configuration in a configuration that reaches the second partial configuration after a finite number of transitions.*

Theorem 1. *The transitivity problem is in Π_2^0 and is Π_1^0-hard.*

Before proving this main result of the paper, we relate it to topological properties of the Turing machine dynamical systems and introduce a last key ingredient: the SMART machine.

2.2 Topological and symbolic dynamics

A *topological dynamical system* is a pair (X, T), where the topological space X is the *phase space* and the continuous function $T : X \to X$ is the *global transition function* of the system. The *orbit* of a point $x \in X$ is the infinite sequence $\mathcal{O}(x) = (T^n(x))_{n\in\mathbb{N}}$. A point x is *periodic* if $T^n(x) = x$ for some $n > 0$.

A system (Y, T) is a *subsystem* of (X, T) if $Y \subseteq X$ is closed and $T(Y) \subseteq Y$. The smallest subsystem that contains a given point x is the closure of its orbit $\overline{\mathcal{O}(x)}$. A system (X, T) is *transitive* if it admits a *transitive point*, *i.e.* a point x such that $\overline{\mathcal{O}(x)} = X$. A system where every point is transitive is a *minimal* system. A system (Z, F) is a *factor* of (X, T) if there exists a continuous and onto function $\varphi : X \to Z$ such that $F \circ \varphi = \varphi \circ T$. Factors inherit several properties including transitivity. A convenient characterization of transitivity is the following (see [10] for this and other details about dynamical systems).

Proposition 1. *Given a compact and perfect set X, the dynamical system (X, T) is transitive if and only if for every pair of open sets U, V, there exists a time t such that $T^t(U) \cap V \neq \emptyset$.*

Let Σ be a finite alphabet. Words, infinite words and bi-infinite words are respectively finite, right-infinite and bi-infinite sequences of symbols from Σ whose sets are denoted respectively by Σ^*, the *one-sided full shift* $\Sigma^{\mathbb{N}}$ and the *two-sided full shift* $\Sigma^{\mathbb{Z}}$. A finite word v is a *factor* of another (finite or infinite) word z, denoted by $v \sqsubseteq z$, if there exist two indexes i and j, such that $v = z_i z_{i+1} ... z_j$. The length of a finite word u is denoted by $|u|$.

Let $\mathbb{T}$ denote either $\mathbb{Z}$ or $\mathbb{N}$. An element x of $\Sigma^{\mathbb{T}}$ is an ordered sequence $x = (x_i)_{i\in\mathbb{T}}$. The *shift* function σ is defined on $\Sigma^{\mathbb{T}}$ by $\sigma(y)_i = y_{i+1}$, and it is a bijective function if $\mathbb{T} = \mathbb{Z}$. The metric of the full shift is the Cantor metric: $d(x, y) = 2^{-i}$, where $i = \min\{|n| \ : \ x_n \neq y_n\}$. With this metric, $\Sigma^{\mathbb{T}}$ is compact and $(\Sigma^{\mathbb{T}}, \sigma)$ is a topological dynamical system. The subsystems of $(\Sigma^{\mathbb{T}}, \sigma)$ are called *subshifts*. Frequently, we will prefer to denote $\Sigma^{\mathbb{N}}$ by Σ^ω, the set of right-infinite sequences; and symmetrically, we will use ${}^\omega\Sigma$ to denote the left-infinite sequences.

The *language* $\mathcal{L}(S)$ of a subshift S is the set of its factors, *i.e.* $\mathcal{L}(S) = \{u \in \Sigma^* | \exists z \in S, \ u \sqsubseteq z\}$. Conversely, every language L defines a subshift $\mathcal{S}_L = \left\{z \in \Sigma^{\mathbb{N}} \middle| \forall u \sqsubseteq z, \ u \in L\right\}$. A set S is a subshift if and only if $\mathcal{S}_{\mathcal{L}(S)} = S$.

2.3 Turing machines seen as dynamical systems

Let $X = Q \times \Sigma^{\mathbb{Z}} \times \mathbb{Z}$ be the set of configurations of a complete Turing machine. Endowing X with the product topology defines a topological dynamical system (X, T). However, X is not a compact set. Following Kůrka [9], we reformulate X to overcome this problem.

Turing machine with moving head (TMH) In this model, the head is added as an element of the tape; then, the phase space is the set $X_h \subset (\Sigma \cup Q)^{\mathbb{Z}}$, defined by $X_h = \{x \in (\Sigma \cup Q)^{\mathbb{Z}} \mid |\{i \in \mathbb{Z} : x_i \in Q\}| \leq 1\}$. The transition function T_h consists in one application of the local transition function δ, taking in consideration that the head position is at the right of the unique cell that contains a state on the tape. Configurations with no state in the tape are *headless* configurations and are fixed points. The coding function $\psi : X \to X_h$ transforms a configuration (s, c, p) into $\psi(s, c, p) = x$, where x is defined by $x_i = c_i$ if $i < p$, $x_p = s$ and $x_i = c_{i-1}$ if $i > p$.

By construction, X_h is a subshift. With the Cantor metric, ψ is continuous and one-to-one and $X_h = \overline{\psi(X)}$. Configurations from $X_h \setminus \psi(X)$ are exactly the headless configurations. The transition function T_h is continuous and partial configurations are represented by factors $x_i...x_j$.

Turing machine with moving tape (TMT) In this model, the head is fixed at the origin and it is the tape which moves. The phase space is $X_t = {}^{\omega}\Sigma \times Q \times \Sigma^{\omega}$ and T_t consists in one application of δ by moving the tape instead of the head.

The onto coding function $\Gamma : X \to X_t$ transforms a configuration (s, c, p) into $(\cdots c_{p-2}c_{p-1}, s, c_p c_{p+1} \cdots)$. Endowing X_t with the product topology results in a compact phase space and both Γ and T_t are continuous. By applying Γ to a configuration, one loses information about the original head position. Partial configurations are triples $(u, s, v) \in \Sigma^* \times Q \times \Sigma^*$.

The trace-shift The *trace-shift* S_t is the column shift associated to the moving tape model. It is obtained from the projection $\pi : X_t \to Q \times \Sigma$ defined by $\pi(u, s, av) = (s, a)$. The *trace-shift* is the image of the factor map $\tau : X_t \to S_t$, defined as $\tau(x) = (\pi(T_t^n(x)))_{n\in\mathbb{N}}$. The definition is generalized to partial configurations. The map τ is not invertible, but given a semi-infinite word $w \in S_t$, its pre-image x is uniquely defined over the set of visited cells. For a given word w either in S_t or in $\mathcal{L}(S_t)$, we define its canonical pre-image (u, s, v) as the smallest finite (or infinite) configuration whose image by τ is w.

It is noteworthy to remark that when T is a reversible machine, S_t can be defined as a subshift of $\Sigma^{\mathbb{Z}}$ by redefining $\tau(u, r, u') = (\pi(T_t^n(x)))_{n\in\mathbb{Z}}$, and all of the results that we will establish in this paper remain true.

3 Transitivity of Turing machines

3.1 Characterizing transitivity properties

By proposition 1, the property described in the Transitivity Problem is indeed the topological transitivity in the TMH model.

Proposition 2. *Let (Q, Σ, δ) be a complete Turing machine.*

The TMH system (X_h, T_h) is transitive if and only if for every pair of partial configurations $(u.u')$ and $(v.v')$, there exists a completion $x \in X_h$ of $(u.u')$ and a time $n \in \mathbb{N}$ such that $T_h^n(x)$ is a completion of $(v.v')$.

The TMT system (X_t, T_t) is transitive if and only if for every pair of partial configurations (u, s, u') and (v, t, v'), there exists a completion $(w, s, w') \in X_t$ of (u, s, u') and a time $n \in \mathbb{N}$ such that $T_t^n(w, s, w')$ is a completion of (v, t, v').

The trace-shift (S_t, σ) is transitive if and only if for every $u, v \in \mathcal{L}(S_t)$, there exists a third word $w \in \mathcal{L}(S_t)$ such that $uwv \in \mathcal{L}(S_t)$.

We see from this that transitivity is in Π_2^0, since the existing configuration needs to be specified only on a finite number of cells. A peculiarity of Turing models is the relation between transitivity and periodic points of T_h. In these points, the head is enclosed in a finite part of the tape. Any perturbation of the configuration that does not affect this part of the tape will not perturb the head. Thus, no periodic point can be attained by a point outside its orbit, and the system cannot be transitive. Moreover, when T_h has a periodic point, transitivity is excluded both from (X_t, T_t) and (S_t, σ) [5].

Proposition 3. (X_h, T_h) *transitive* $\underset{(1)}{\Rightarrow}$ (X_t, T_t) *transitive* $\underset{(2)}{\Rightarrow}$ (S_t, σ) *transitive.*

Proof. (1) Any finite configuration of X_t corresponds to several finite configurations of X_h, thus if a point exists that visits any finite configuration of X_h, the same point will visit any finite configuration of X_t. (2) (S_t, σ) is a factor of (X_t, T_t) thus it inherits its transitivity. ■

Surjectivity is a necessary condition for transitivity, thus if the TMT or the TMH models are transitive, the machine needs to be reversible. Note this is not the case for trace-shift, as there exist non surjective Turing machines with surjective trace-shift [15].

3.2 The SMART machine

The SMART machine is the 4-state 3-symbols reversible complete Turing machine depicted on figure 1. Among other properties, it is aperiodic, its trace-shift is substitutive and it is minimal in TMT, as proven in [3]. We prove below that it is also transitive in TMH.

The head of SMART zigzags over the lagoons of 0s, either to the right or to the left, depending on its states and the surrounding symbols, as formalized below and proved by recurrence over n in [3].

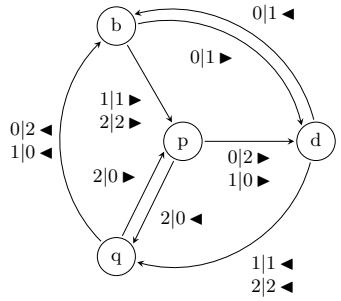

Fig. 1. The SMART machine.

Lemma 1. *For all $n \in \mathbb{N}$ and all symbols $s_+ \in \{1,2\}$ and $s_* \in \{0,1,2\}$,*

$$\begin{pmatrix} s_* & 0^n & 0 & s_+ \\ & & b & \end{pmatrix} \vdash^* \begin{pmatrix} s_* & 0^{n+1} & s_+ \\ b & & & \end{pmatrix} \qquad (B(n))$$

$$\begin{pmatrix} s_+ & 0 & 0^n & s_* \\ & d & & \end{pmatrix} \vdash^* \begin{pmatrix} s_+ & 0^{n+1} & s_* \\ & & d & \end{pmatrix} \qquad (D(n))$$

$$\begin{pmatrix} 0 & 0^n & s_+ \\ p & & \end{pmatrix} \vdash^* \begin{pmatrix} 0^{n+1} & s_+ \\ & p & \end{pmatrix} \qquad (P(n))$$

$$\begin{pmatrix} s_+ & 0^n & 0 \\ & & q \end{pmatrix} \vdash^* \begin{pmatrix} s_+ & 0^{n+1} \\ q & & \end{pmatrix} \qquad (Q(n))$$

A key lemma for the minimality of SMART in [3] is that every partial configuration can be produced inside a large enough block of 0s guarded by 2s.

Lemma 2. *For every word $u \in \{0,1,2\}^*$ of length n, state s and every $1 \leqslant i \leqslant n$, there exist $k, k' \in \mathbb{N}$ such that* $\begin{pmatrix} .2 & 0^{k+k'+n-3} & 0 & 2 \\ & & b & \end{pmatrix} \vdash^* \begin{pmatrix} .2^k & u_1 & \dots & u_i & \dots & u_n 2^{k'} \\ & & & s & & \end{pmatrix}$.

We now introduce two new technical lemmas.

Lemma 3. *The configuration* $\begin{pmatrix} {}^w2 & .2 & 2^w \\ & p & \end{pmatrix}$ *reaches each of the configurations of the family* $\{\begin{pmatrix} {}^w2\ 0\ 0^k & .0^k & 0 & 2\ 0\ 2^w \\ & & b & \end{pmatrix}\}_{k\in\mathbb{N}} \cup \{\begin{pmatrix} {}^w2\ 0\ 0\ 0^k & .0^k & 0 & 2\ 0\ 0\ 2^w \\ & & b & \end{pmatrix}\}_{k\in\mathbb{N}}$.

Proof. By a simple recurrence over k. From k to $k+1$ apply the recurrence hypothesis then $B(2k)$, one step, $P(2k)$, one step. From there one step gives the even case. For the odd case apply $Q(2k)$, one step, $P(2k+2)$ then 2 steps. ■

Lemma 4. *For every $k \leq n-1$,* $\begin{pmatrix} .2 & 0^{n+2} & 0 & 2 \\ & & b & \end{pmatrix}$ *reaches both* $\begin{pmatrix} .2\ 2\ 0^k & 0 & 2\ 0^{n-k} & 2 \\ & b & & \end{pmatrix}$ *and* $\begin{pmatrix} .2\ 0^{n-k}\ 2\ 0^k & 0 & 2\ 2 \\ & b & & \end{pmatrix}$.

Proof. Apply $B(n+2)$, 2 steps, then $D(n+1)$. In the first case continue by 2 steps then repeat $n-k-1$ times the sequence $B(n-i)$, one step, $P(n-i)$, 2 steps, from $i=0$ to $n-k-1$. In the second case continue by one step then repeat $n-k-1$ times the sequence $Q(n-i+1)$, 2 steps, $D(n-i)$, one step, and finish by one step. ■

Theorem 2. *The SMART machine is topologically transitive in TMH.*

Proof. By lemma 2 any possible partial configuration is reachable from a partial configuration x' with a certain amount of 0 in a certain position. Lemma 4 establishes that x' is reachable from a configuration x'' with the 0s in the center. Finally, lemma 3 asserts that x'' is always reachable from $\left(\begin{smallmatrix} {}^{w}2 & & 2^{w} \\ & {}^{.2}_{p} & \end{smallmatrix}\right)$. Therefore, configuration $\left(\begin{smallmatrix} {}^{w}2 & & 2^{w} \\ & {}^{.2}_{p} & \end{smallmatrix}\right)$ is a transitive point, and SMART is transitive. ■

4 The complexity of topological transitivity

4.1 Construction techniques

Our proof combines partial Turing machines to construct bigger ones. One key technique from [8] is *Reversing the time*: given a reversible Turing machine $M = (Q, \Sigma, \delta)$, one creates two new reversible machines $M_+ = (Q \times \{+\}, \Sigma, \delta^+)$, and $M_- = (Q \times \{-\}, \Sigma, \delta^-)$, where $(s, +)$ and $(s, -)$ states represent M in state s running respectively forwards and backwards in time.

The second key technique is the *Embedding* that inserts a machine inside the transitions of another in such a way that the new machine has one or more properties that depends on some properties of the original machines. We distinguish a *host* machine $H = (Q, \Sigma, \delta)$ and a reversible and *innocuous* invited machine I that share the same alphabet.

Definition 1. *A reversible machine is* innocuous *if every starting pair (s, a) is associated to a unique halting pair (t, a) so that the evolution of every starting configuration $(s, c, p + \mu(s))$ where $c(p) = a$ either is infinite or stops in the halting configuration (t, c, p).*

Remark 1. Innocuous machines can be obtained by gluing together the halting pairs of M_+ to the starting pairs of M_-, so that a halting configuration computes back to the starting configuration.

The embedding H^I of the invited machine I in the host machine H is constructed from the disjoint union of H and I. Let $(s_1, a_1), \ldots, (s_n, a_n)$ be the starting pairs of I and $(t_1, a_1), \ldots, (t_n, a_n)$ be the associated halting pairs. Let $\delta(r, a) = (q, b, \Delta)$ be a fixed transition of H. That transition is removed from H^I and replaced by the following transitions: $\delta(r, a) = (s_1, a_1, \mu(s_1))$, $\delta(t_1, a_1) = (s_2, a_2, \mu(s_2))$, $\delta(t_2, a_2) = (s_3, a_3, \mu(s_3))$, $\ldots$, $\delta(t_n, a_n) = (q, b, \Delta)$, as depicted on figure 2. Notice that H^I is complete when H is complete.

If we start at a state of H in the resulting machine, we will see the evolution of H, alternated with some intervals of time in which it is the machine I that evolves.

4.2 Undecidability of transitivities

Theorem 3. *The problem to decide if a given reversible complete Turing machine is transitive is Π_1^0-hard in all three models: TMH, TMT and trace-shift.*

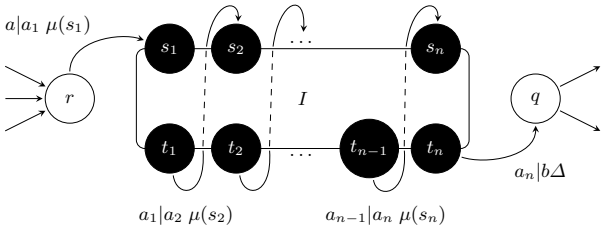

Fig. 2. Embedding technique.

Proof. The proof proceeds by reduction of the Reachability Problem for binary reversible aperiodic Turing machines. Let M be such a machine with a starting pair (s, a) and a halting pair (t, b). Let \$ be a new symbol not in Σ and consider M' the copy of M with this new symbol: all the pairs $(r, \$)$ are both starting and halting pairs of M'. Let (p_1, a_1), ..., (p_m, a_m) be all the starting pairs of M' except (s, a) and (q_1, b_1), ..., (q_m, b_m) be all the halting pairs of M' except (t, b). Take two copies of M' and apply *Reversing the time* to obtain 4 machines M'_{1-}, M'_{1+}, M'_{2-} and M'_{2+}, then connect them according to figure 3 by adding the following transitions to obtain the invited machine I:

$$\begin{aligned}
\delta(t^{1+}, b) = (t^{1-}, b, -\mu(t)),\ \delta(s^{1-}, a) = (s^{1+}, a, \mu(s)),\\
\delta(t^{2+}, b) = (t^{2-}, b, -\mu(t)),\ \delta(s^{2-}, a) = (s^{2+}, a, \mu(s)),\\
\delta(q_i^{1+}, b_i) = (q_i^{2-}, b_i, -\mu(q_i)) \quad \forall i \in \{1, \dots, m\}\\
\delta(p_i^{1-}, a_i) = (p_i^{2+}, a_i, \mu(p_i)) \quad \forall i \in \{1, \dots, m\}
\end{aligned}$$

The starting pairs of I are the pairs (p_i^{1+}, a_i) and the pairs (q_i^{1-}, b_i) for all i. The machine I is innocuous as only three scenarios are possible from a starting pair: (1) entering M_{1+} (or M_{1-}) and staying there forever; (2) exit M_{1+} by (t^{1+}, b) (or M_{1-} by (s^{1-}, a)), the computation is then reversed inside M_{1-} (or M_{1+}) and enters M_{2+} (or M_{2-}) replaying the same scenario to exit M_{2-} (or M_{2+}) leaving the tape identical to the beginning ; (3) exit M_{1+} (or M_{1-}) by a halting pair (q_i^{1+}, b_i), the computation is then reversed in M_{2-} (or M_{2+}) leaving the tape identical to the beginning. Consider the embedding SMART^I.

(Assertion 1) SMART^I may be transitive only if M cannot reach (t, b) from (s, a). Indeed, if (s, a) can reach (t, b) in M then SMART^I admits a periodic point and its trace shift cannot be transitive and by proposition 3 none of the models can be transitive.

(Assertion 2) The TMH system of SMART^I is transitive if M cannot reach (t, b) from (s, a). Indeed, suppose that M cannot reach (t, b) from (s, a), then I is aperiodic, and let (s_u, u, i) and (s_v, v, j) be two partial configurations.

Case 1. s_u and s_v are states of SMART. In this case we know that there exists a *finite* context (s_u, u', i) that extends (s_u, u, i) so that SMART reaches

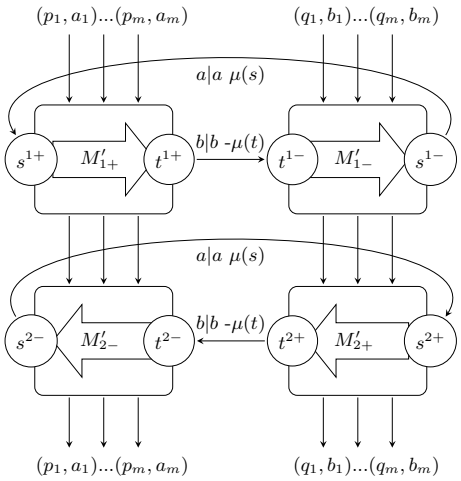

Fig. 3. Invited machine for an embedding that is transitive if and only if (s, a) cannot reach (t, b) in the evolution of M, used in the proof of theorem 3.

(s_v, v, j), because SMART is transitive. Let us complete u' with the symbol \$ and lets analyze the behavior of SMART^I over $x = (s_u, {}^{\omega}\$u'\$^{\omega}, i)$. First of all, SMART^I cannot stay an infinite amount of time inside I, because I is aperiodic, and non periodic behavior needs an infinite amount of space to be performed. The presence of the extraneous symbol \$ in x, avoid this to happen. Now, since I is innocuous, we will see the machine SMART evolving, and thus configuration (s_v, v, j) will be reached.

Case 2. s_u or s_v is a state of I. In this case, lets add \$ symbols around u and v, and lets evolve the time backward from $(s_v, \$v\$, j)$ until the embedded machine exits the invited machine I, and lets call $(s'_v, \$v'\$, j')$ the obtained finite configuration. In the same way, we evolve the time forward from $(s_u, \$u\$, i)$ until exiting I, and we call $(s'_u, \$u'\$, i')$ the so obtained configuration. If either s_u or s_v are already in SMART, we just add the \$ symbols. Now we can apply Case 1 to $(s'_u, \$u'\$, i')$ and $(s'_v, \$v'\$, j')$ to prove the existence of a completion u'' of $\$u'\$$ such that $(s'_u, u'', i') \vdash^* (s'_v, \$v'\$, j')$. Lets suppose that $u'' = w\$u'\$w'$. Therefore, by construction, we have that $(s_u, w\$u\$w', i) \vdash^* (s'_u, w\$u'\$w', i') \vdash^* (s'_v, \$v'\$, j') \vdash^* (s_v, \$v\$, j)$, which is the desired conclusion.

From proposition 3 we know that the classes of transitive machines for TMH, TMT and the trace-shift are nested, Assertions 1 and 2 prove that all three related problems are Π_1^0-hard. ■

5 The complexity of minimality

A dynamical system (X, T) is *minimal* if $\overline{\mathcal{O}(x)} = X$ for all $x \in X$. It is equivalent to not have any no trivial proper subsystem. Every minimal system is also transitive. In the context of Turing machines, we will found machines which have a minimal TMT and a minimal trace-shift, the SMART machine is an example of this [3]. There is no machine with a minimal TMH system, because this system always contains fixed points: the headless configurations. As for transitivity, minimality in the TMT model imply minimality in the trace-shift, by the factor relation.

Theorem 4. *The problem to decide if a given reversible complete Turing machine is minimal is Σ_1^0-hard for both TMT and trace-shift.*

Proof. The proof proceeds by reduction of the *Mortality problem* for reversible and aperiodic machines, proved undecidable in [8]. A Turing machine is mortal if every configuration eventually halts. The mortality problem, that is to decide, given a Turing machine, if it is mortal, is Σ_1^0-complete for reversible Turing machines.

Let M be a reversible and aperiodic ternary Turing machine. Apply *Reversing the time* to generate two machines M_+ and M_- and combine them into an invited machine as per figure 4: for every halting pair (q_i, b_i) of M, add a transition $\delta(q_i^+, b_i) = (q_i^-, b_i, -\mu(q_i))$. The machine I is innocuous, and we embed it into SMART to produce SMART^I.

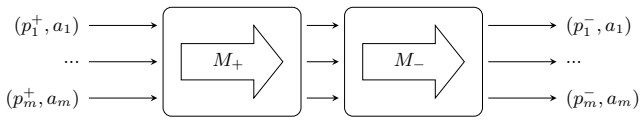

Fig. 4. Invited machine for an embedding that is minimal if and only if M is mortal, used in the proof of theorem 4.

(Assertion 1) If the trace-shift of SMART^I is minimal, then M needs to be mortal. Indeed, if M is not mortal, there is a trace that starts in a state of M_+ and never exits this machine. Such a behavior is impossible in a machine with a minimal trace-shift, where all the trajectories need to be transitive, and so they must visit all the states of the machine.
(Assertion 2) If M is mortal, the TMT system of SMART^I is minimal. Indeed, lets suppose that M is mortal, so its reverse is mortal too, and so is I. Let x be an arbitrary configuration in the TMT system of SMART^I, we will prove that it reaches every finite configuration (v, r, v') in the TMT system. First, if r is a state of SMART and x has also a state of SMART, it is clear that x reaches (v, r, v') because SMART is minimal and it is impossible to stay an infinite amount of time inside I. Second, if x has a state in I, it comes from a configuration x' that

do has a state in SMART, because the inverse of I is mortal. The orbit of x is equal to the orbit of x' except for a finite number of points, whose state is in I, thus, by the previous argument, x can attain any finite configuration with a state in SMART. Now, if r is a state of I, we can evolve SMART^I backward on (v, r, v') until to arrive to configuration (u, r', u') with a state r' of SMART, this is always possible because I is mortal. Through the former arguments, we know that x visits (u, r', u'), it thus visits (v, r, v') too, and we conclude the proof of assertion 2.

The class of machines with a minimal TMT system is contained in the class of machines with a minimal trace-shift, thus Assertion 1 and 2 prove that the two related problems are Σ_1^0-hard. ■

References

1. Asarin, E., Maler, O., Pnueli, A.: Reachability analysis of dynamical systems having piecewise-constant derivatives. Theor. Comput. Sci. 138(1), 35–65 (1995)
2. Blondel, V.D., Cassaigne, J., Nichitiu, C.: On the presence of periodic configurations in Turing machines and in counter machines. Theoret. Comput. Sci. 289, 573–590 (2002)
3. Cassaigne, J., Ollinger, N., Torres-Avilés, R.: A Small Minimal Aperiodic Reversible Turing Machine (2014), https://hal.archives-ouvertes.fr/hal-00975244v1, (submitted to a journal)
4. Gajardo, A., Guillon, P.: Zigzags in Turing machines. In: Ablayev, F. anf Mayr, E. (ed.) Computer Science Symposium in Russia (CSR 2010). LNCS, vol. 6072, pp. 109–119 (2010)
5. Gajardo, A., Mazoyer, J.: One head machines from a symbolic approach. Theor. Comput. Sci. 370, 34–47 (2007)
6. Jeandel, E.: Computability of the entropy of one-tape Turing machines. In: Mayr, E., Portier, N. (eds.) Symposium on Theoretical Aspects of Computer Science (STACS 2014). vol. 25, pp. 421–432 (2014)
7. Kari, J.: Rice's theorem for the limit sets of cellular automata. Theor. Comput. Sci. 127(2), 229–254 (1994)
8. Kari, J., Ollinger, N.: Periodicity and immortality in reversible computing. In: Ochmanski, E., Tyszkiewicz, J. (eds.) Mathematical Foundations of Computer Science (MFCS 2008). LNCS, vol. 5162, pp. 419–430 (2008)
9. Kůrka, P.: On topological dynamics of Turing machines. Theoret. Comput. Sci. 174(1-2), 203–216 (1997)
10. Kůrka, P.: Topological and Symbolic Dynamics. Société Mathématique de France, Paris, France (2003)
11. Lukkarila, V.: Sensitivity and topological mixing are undecidable for reversible one-dimensional cellular automata. Cellular Automata 5(3), 241–272 (2010)
12. Margolus, N.: Physics and Computation. Ph.D. thesis, M. I. T., Cambridge, Mass., U.S.A. (1987)
13. Moore, C.: Generalized shifts: unpredictability and undecidability in dynamical systems. Nonlinearity 4(2), 199–230 (1991)
14. Oprocha, P.: On entropy and Turing machine with moving tape dynamical model. Nonlinearity 19, 2475–2487 (2006)
15. Torres, R., Ollinger, N., Gajardo, A.: Undecidability of the surjectivity of the subshift associated to a Turing machine. LNCS (7581), 44–56 (2013)

16. Turing, A.: On computable numbers, with an application to the entscheidungsproblem. Proc. of the London Math. Soc. 42(2), 230–265 (1936)

A Complementary material for the reviewers

We provide here some more details that the space constraint did not allow us to include in the main text. We also discuss the syntactical choices for the machine model and their impact on the results. Moreover, as [3] is unpublished (submitted to a journal, not accepted yet), we copy here every material for this paper to be independent.

A.1 Influence of the chosen model

Turing machines can be considered in two forms, the *quintuple model* and the *quadruple model*, and it acts on the configurations differently depending on the model, as follows.

Quintuple model. $\delta : Q \times \Sigma \to Q \times \Sigma \times \{-1, 0, 1\}$. The machine transforms a configuration (s, c, p) into $(t, c', p+d)$, if $\delta(s, c(p)) = (t, a, d)$, and c' is defined by $c'(p) = a$ and $c'(i) = c(i)$ for all $i \neq p$.

Quadruple model. $\delta : \left| \begin{array}{ll} Q_W \times \Sigma & \to \quad Q \times \Sigma \\ Q_M \times \{/\} & \to Q \times \{-1, 0, 1\} \end{array} \right.$, where $Q = Q_W \sqcup Q_M$. The machine transforms a configuration (s, c, p) into (t, c', p), if $s \in Q_W$, $\delta(s, c(p)) = (t, a)$ and c' is defined by $c'(p) = a$ and $c'(i) = c(i)$ for all $i \neq p$, but if $s \in Q_M$, then (s, c, p) is transformed into $(t, c, p + d)$, where $\delta(s, /) = (t, d)$. The first type of transformation is called *writing instruction*, and the second one *movement instruction*.

These two models are equivalent from a computational point of view but are slightly different from a dynamical point of view: reversibility is trivially characterized in the quadruple model and more subtle for the quintuple model, some properties are not preserved when transforming a machine from a model to the other (the head velocity for example is reduced to the half when going to the quadruple model). Fortunately, as we will prove in the full version of this paper, transitivity is not affected by these transformations: the results presented in this paper are valid in both models.

A.2 The reachability problem

Here we justify the following claim about the *Reachability problem* from the paper that might at first seem quite strong:

Reachability problem *Given a Turing machine and a pair of states, decide if, starting from a configuration in the first state, the machine can reach a configuration in the second state after a finite number of transitions.*

The reachability problem is known to remain Σ_1^0-complete when the input is restricted to aperiodic reversible Turing machines [8] from a starting pair to a halting pair. Moreover, one can fix the alphabet of the input machine to be binary or ternary.

Recall that the construction that support the *Reachability problem* undecidabilty in [8] proceeds as follows: given a reversible counter machine with 2 counters and initial state s_0, one constructs recursively a reversible Turing machine that simulates the counter machine using Hooper-style recursive encoding and has two states s_1 and s_2 such that there is a computation from a configuration in state s_1 to a configuration in state s_2 if and only if the counter machine eventually halts starting from the configuration $(s_0, 0, 0)$. The constructed Turing machine is aperiodic and reversible.

As the *Reachability problem* deals only with states (as opposed to partial configurations), one can simply prune the states s_1 and s_2 so that every pair (s_1, a) is a starting pair and every pair (s_2, a) is a halting pair.

Concerning the alphabet, we can prove that in fact, any reversible machine can be simulated by a reversible machine with a binary alphabet, in such a way that the new machine preserves dynamical properties such as periodicity, aperiodicity and mortality, and also computational properties as reachability.

Lemma 5. *Given a reversible Turing machine $M = (Q, \Sigma, \delta)$, there exists a binary and reversible Turing machine $\tilde{M} = (\tilde{Q}, \{0,1\}, \tilde{\delta})$ and an injective function $\Psi : Q \times \Sigma^{\mathbb{Z}} \times \mathbb{Z} \to \tilde{Q} \times \{0,1\}^{\mathbb{Z}} \times \mathbb{Z}$ such that, if T is the global transition function of M, $\tilde{T}$ is the global transition function of $\tilde{M}$, and $k = \lceil log_2(|\Sigma|) \rceil$, then $\tilde{T}^{3k+1} \circ \Psi = \Psi \circ T$.*

Proof. Let $M = (Q, \Sigma, \delta)$ be a reversible Turing machine and let us code Σ in a binary alphabet $\mathbf{2} = \{0, 1\}$, by tuples of length $k = \lceil log_2(|\Sigma|) \rceil$. Let us call $\phi : \Sigma \to \mathbf{2}$ the coding function. It needs to be injective, but not necessarily surjective. Now we adapt the machine to work over these coded symbols. We call the simulation machine as $\tilde{M} = (\tilde{Q}, \mathbf{2}, \tilde{\delta})$.

The idea behind this simulation is that machine $\tilde{M}$ do the same that machine M, but just in more steps. In this way, periodicity, reachability, aperiodicity and mortality are kept. Then, to simulate instruction $\delta(s, a) = (t, b, \Delta)$, we start in a state $s \in Q$, and $\tilde{M}$ reads in the next k cells a coded symbol $a \in \Sigma$. While coded symbol a is read, $\tilde{M}$ conveniently writes word 0^k in those cells to help us to keep reversibility. Now, machine $\tilde{M}$ comes back to the left writing coded symbol b until it reaches starting position. Finally it moves k spaces in direction Δ, and then machine $\tilde{M}$ reaches state $t \in Q$. An outline of this steps can be seen in Figure 5. Hence, instruction (s, a, t, b, Δ) is correctly simulated. Any instruction in δ is simulated in this way.

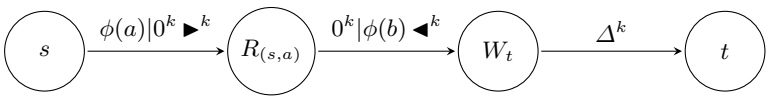

Fig. 5. Outline of the steps to simulate a instruction $\delta(s, a) = (t, b, \Delta)$ of machine M.

We just need to prove that machine $\tilde{M}$ is reversible. We can separate the simulation of a M instruction in three steps: Reading (from state s to state $R_{(s,a)}$), Writing (from state $R_{(s,a)}$ to state W_t) and Moving (from state W_t to state t).

The first part Reading is easily done in a reversible way performing a binary tree, where its leafs are the states $R_{(s,a')}$, with $a' \in \Sigma$ and the root node is s. The second part Writing can be done reversibly with a inverted binary tree, where the leafs are the states $W_{(t,T(b'))}$, with $T(b') \in \Sigma$ the transposed of $b' \in \Sigma$ ($T(a_0a_1...a_{k-1}) = a_{k-1}a_{k-2}...a_0$) and the root node is W_t. To perform the third part Moving in a reversible way, it is just needed $|k| - 1$ intermediate states and movement instructions from W_t to t.

The connection between first and second part is done with the instruction $\tilde{\delta}(R_{(s,a)}, 0) = (W_{(t,T(b))}, 0, 0)$. In this part can not be problem with reversibility, because if M is reversible, just one instruction can write symbol b going to state t in machine M, therefore state $W_{(t,T(b))}$ has just one in degree. A graph representation of simulation of a instruction can be seen in Figure 6.

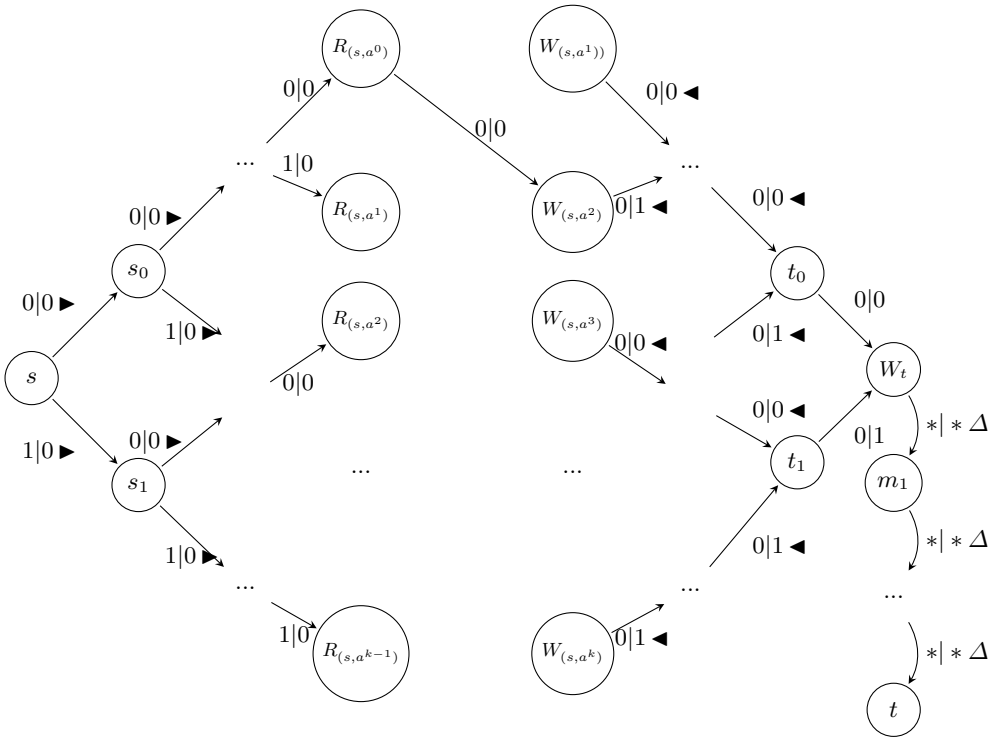

Fig. 6. Detailed instruction $\delta(s, a^0) = (t, a^{2^{|k-1|}}, \Delta)$ simulated in $\tilde{M}$. The symbols $a^n \in \Sigma$ represents the ordered symbols of machine M. Notice that $\phi(a^1) = 0^k$, $\phi(a^1) = 0^{k-1}1$ and so on.

Now, as the binary trees have a height of k, connected by one instruction and, finally, k movements instructions, each evolution of T is done in $3k + 1$ steps by $\tilde{T}$.

■

Finally, one might argue that the constructions in [8] are done in the quadruple model. Fortunately, every machine constructed in the cited paper alternates between moving and writing state: they can be recoded into equivalent quintuple machines with twice less states without loosing the required properties.

A.3 SMART lemmas

Proof (of lemma 1). The proof proceeds by recurrence over n for all s_+ and s_* and all four properties. The base case can be checked by hand. Let us take $n \geq 1$. We will do the proofs just for $B(n)$ and $P(n)$, because $D(n)$ and $Q(n)$ are symmetric. Let us suppose that $B(n-1)$, $D(n-1)$, $P(n-1)$ and $Q(n-1)$ are true. First we prove $B(n)$.

$$\left(\begin{array}{cccccc} s_* & 0 & 0 & 0^{n-2} & 0 & s_+ \\ & & & & b & \end{array} \right)$$

Apply $B(n-1)$

$$\left(\begin{array}{cccccc} s_* & 0 & 0 & 0^{n-2} & 0 & s_+ \\ & b & & & & \end{array} \right)$$

One step

$$\left(\begin{array}{cccccc} s_* & 1 & 0 & 0^{n-2} & 0 & s_+ \\ & & d & & & \end{array} \right)$$

Apply $D(n-1)$

$$\left(\begin{array}{cccccc} s_* & 1 & 0 & 0^{n-2} & 0 & s_+ \\ & & & & & d \end{array} \right)$$

One step

$$\left(\begin{array}{cccccc} s_* & 1 & 0 & 0^{n-2} & 0 & s_+ \\ & & & & q & \end{array} \right)$$

Apply $Q(n-1)$

$$\left(\begin{array}{cccccc} s_* & 1 & 0 & 0^{n-2} & 0 & s_+ \\ & q & & & & \end{array} \right)$$

One step

$$\left(\begin{array}{cccccc} s_* & 0 & 0 & 0^{n-2} & 0 & s_+ \\ b & & & & & \end{array} \right)$$

Now, for $P(n)$:

$$\begin{pmatrix} 0\ 0\ 0^{n-2}\ 0\ s_+ \\ p \end{pmatrix}$$

One step

$$\begin{pmatrix} 2\ 0\ 0^{n-2}\ 0\ s_+ \\ d \end{pmatrix}$$

Apply $D(n-1)$

$$\begin{pmatrix} 2\ 0\ 0^{n-2}\ 0\ s_+ \\ d \end{pmatrix}$$

One step

$$\begin{pmatrix} 2\ 0\ 0^{n-2}\ 0\ s_+ \\ q \end{pmatrix}$$

Apply $Q(n-1)$

$$\begin{pmatrix} 2\ 0\ 0^{n-2}\ 0\ s_+ \\ q \end{pmatrix}$$

One step

$$\begin{pmatrix} 0\ 0\ 0^{n-2}\ 0\ s_+ \\ p \end{pmatrix}$$

Apply $P(n-1)$

$$\begin{pmatrix} 0\ 0\ 0^{n-2}\ 0\ s_+ \\ p \end{pmatrix}$$

The cases of $D(n)$ and $Q(n)$ are symmetric. ■

It is important to remark that applying $B(n)$ always implies to apply $B(n-1)$ as the first step, and also $B(n-2)$ and $B(0)$. Analogously, applying $P(n)$ implies to apply $P(n-1)$ and $P(0)$ as the last step. Interestingly, just before applying $P(0)$ by the last time, i.e., 3 steps before finishing $P(n)$, the head is on the rightmost 0 with state p. This will be used in the future. Thus, we define the next two propositions which also hold.

$P'(n)$: $\forall s_+ \in \{1,2\}$, $\begin{pmatrix} 0\ 0^n\ s_+ \\ p \end{pmatrix} \vdash^* \begin{pmatrix} 0^n\ 0\ s_+ \\ p \end{pmatrix}$

$Q'(n)$: $\forall s_+ \in \{1,2\}$, $\begin{pmatrix} s_+\ 0^n\ 0 \\ q \end{pmatrix} \vdash^* \begin{pmatrix} s_+\ 0\ 0^n \\ q \end{pmatrix}$

When starting with one or more 0s around the head, SMART will recursively augment the number of 0s around the head. This is justified in the next lemma.

Lemma 6. *If we define, for every $n \geq 0$, the set $C_n = \{x \mid x$ is a completion of a shift of $\begin{pmatrix} s_+\ 0\ 0^n \\ q \end{pmatrix}$ or $\begin{pmatrix} 0^n\ 0\ s_+ \\ p \end{pmatrix}\}$, then for every $x \in C_n$, either x or the orbit of x will eventually visit C_m for arbitrary large m.*

Proof. We just make the proof for initial state q, since p is symmetrical. Let us start with n maximal such that $x \in C_n$. If there is no maximal n, then x is already on every C_m.

$$\begin{pmatrix} s_*\ s_+\ 0\ 0^n\ s_{++} \\ q \end{pmatrix}$$

one step

$$\begin{pmatrix} s_*\ s_+\ 2\ 0^n\ s_{++} \\ b \end{pmatrix}$$

one step

$$
\begin{array}{ll}
\text{If } s_+ = 1 & \text{If } s_+ = 2 \\
\left(\begin{smallmatrix} s_* & 1 & 2 & 0^n & s_{++} \\ & & p & & \end{smallmatrix} \right) & \left(\begin{smallmatrix} s_* & 2 & 2 & 0^n & s_{++} \\ & & p & & \end{smallmatrix} \right) \\
\text{one step} & \text{one step} \\
\left(\begin{smallmatrix} s_* & 1 & 0 & 0^n & s_{++} \\ & q & & & \end{smallmatrix} \right) & \left(\begin{smallmatrix} s_* & 2 & 0 & 0^n & s_{++} \\ & q & & & \end{smallmatrix} \right) \\
\text{one step} & \text{one step} \\
\left(\begin{smallmatrix} s_* & 0 & 0 & 0^n & s_{++} \\ b & & & & \end{smallmatrix} \right) & \left(\begin{smallmatrix} s_* & 0 & 0 & 0^n & s_{++} \\ & & p & & \end{smallmatrix} \right) \\
\text{If } s_* \neq 0 & P'(n) \\
\left(\begin{smallmatrix} s_* & 0 & 0 & 0^n & s_{++} \\ & p & & & \end{smallmatrix} \right) & \left(\begin{smallmatrix} s_* & 0 & 0^n & 0 & s_{++} \\ & & & p & \end{smallmatrix} \right) \\
P'(n+1) & \text{we are done} \\
\left(\begin{smallmatrix} s_* & 0 & 0^n & 0 & s_{++} \\ & & & p & \end{smallmatrix} \right) & \\
\text{we are done} &
\end{array}
$$

Now we study the case $s_* = 0$.

$$
\begin{array}{l}
\left(\begin{smallmatrix} 0 & 0 & 0 & 0^n & s_{++} \\ b & & & & \end{smallmatrix} \right) \\
\text{one step} \\
\left(\begin{smallmatrix} 1 & 0 & 0 & 0^n & s_{++} \\ & d & & & \end{smallmatrix} \right) \\
D(n+1) \\
\left(\begin{smallmatrix} 1 & 0 & 0 & 0^n & s_{++} \\ & & & d & \end{smallmatrix} \right) \\
\text{one step} \\
\left(\begin{smallmatrix} 1 & 0 & 0^n & 0 & s_{++} \\ & & q & & \end{smallmatrix} \right) \\
Q'(n+1) \\
\left(\begin{smallmatrix} 1 & 0 & 0^n & 0 & s_{++} \\ & q & & & \end{smallmatrix} \right) \\
\text{we are done}
\end{array}
$$

■

If we start with an arbitrary configuration, it can be verified by hand that in a finite number of steps we fall in one of the sets of Lemma 6.

Another important property of SMART is its time symmetry. A restricted version of time symmetry [3] says that a reversible machine is time-symmetric if there exists two involutions $h_\Sigma : \Sigma \to \Sigma$ and $h_Q : Q \to Q$ such that for all $a, b \in \Sigma$ and $t, s \in Q$

$$\delta^{-1}(t, b) = (s, a, d) \Leftrightarrow \delta(h_Q(t), h_\Sigma(b)) = (h_Q(s), h_\Sigma(a), d).$$

Once this is defined, one can define a global function $h : Q \times \Sigma^{\mathbb{Z}} \times \mathbb{Z} \to Q \times \Sigma^{\mathbb{Z}} \times \mathbb{Z}$ by $s \to h_Q(s)$, $c_i \to h_\Sigma(c_i)$ and $p \to p + \mu(s)$, where μ is the function that gives the unique direction from which state s is attained. This function h results to be an involution and satisfy that $T \circ h \circ T \circ h$ is the identity.

SMART is time-symmetric through the involutions defined by: $h_\Sigma(0) = 0$, $h_\Sigma(1) = 2$, $h_Q(b) = p$ and $h_Q(d) = q$.

Proof (of lemma 2). First, we will use the fact that the SMART machine is time-symmetric, so applying $h \circ T^t \circ h$ is the same as applying T^{-t}, for

any time $t \in \mathbb{N}$. So now we just have to prove that $h\left(\begin{smallmatrix} .2^k\ u_1 \dots \underset{s}{u_i} \dots u_n 2^{k'} \end{smallmatrix}\right) = \left(\begin{smallmatrix} .1^k\ h_\Sigma(u_1) \dots \underset{h_Q(s)}{\dots} \dots h_\Sigma(u_n)\ 1^{k'} \end{smallmatrix}\right)$ will eventually reach $h\left(\begin{smallmatrix} .2\ 0^{k+k'+n-3}\ \underset{b}{0}\ 2 \end{smallmatrix}\right) = \left(\begin{smallmatrix} .1\ 0^{k+k'+n-3}\ 0\ \underset{p}{1} \end{smallmatrix}\right)$.

Based on the Lemma 6, we know that we can generate an increasing amount of 0s in the tape. For this reason, we know that $\left(\begin{smallmatrix} 1\ h_\Sigma(u_1) \dots \underset{h_Q(s)}{\dots} \dots h_\Sigma(u_n)\ 1 \end{smallmatrix}\right)$ will eventually reach one of the configurations considered in Lemma 6 with a block of 0s that will grow until arriving to one of the next configurations:

$$\left(\begin{smallmatrix} .1\ 0^j\ \underset{p}{0}\ v_1 \dots v_l\ 1 \end{smallmatrix}\right) \text{ or } \left(\begin{smallmatrix} .1\ \underset{q}{0}\ 0^j\ v_1 \dots v_l\ 1 \end{smallmatrix}\right) \text{ or } \left(\begin{smallmatrix} .1\ v_1 \dots\ v_l\ \underset{q}{0}\ 0^j\ 1 \end{smallmatrix}\right) \text{ or } \left(\begin{smallmatrix} .1\ v_1 \dots v_l\ 0^j\ \underset{p}{0}\ 1 \end{smallmatrix}\right).$$

Thus, applying either $Q(0)$ or $P(0)$ we arrive to one of the next situations:

$$\begin{matrix} (i) \\ \left(\begin{smallmatrix} .1\ 0^j\ \underset{p}{0}\ v_1 \dots v_l\ 1 \end{smallmatrix}\right) \end{matrix} \text{ or } \begin{matrix} (ii) \\ \left(\begin{smallmatrix} .1\ \underset{q}{0}\ 0^j\ v_1 \dots v_l\ 1 \end{smallmatrix}\right) \end{matrix} \text{ or } \begin{matrix} (iii) \\ \left(\begin{smallmatrix} .1\ v_1 \dots\ v_l\ \underset{q}{0}\ 0^j\ 1 \end{smallmatrix}\right) \end{matrix} \text{ or } \begin{matrix} (iv) \\ \left(\begin{smallmatrix} .1\ v_1 \dots v_l\ 0^j\ \underset{p}{0}\ 1 \end{smallmatrix}\right) \end{matrix}, \tag{1}$$

for some $v \in \{0,1,2\}^{n-j-1}$, and $v_1 \neq 0$ in situations (i) and (ii), or $v_l \neq 0$ in the other two.

In order to reach $\left(\begin{smallmatrix} 1\ 0^{k+k'+n-3}\ 0\ \underset{p}{1} \end{smallmatrix}\right)$, we will need a certain amount of 1s at the left or right of the initial configuration; this amount depends on v. Let $k(v)$ be the function that gives the amount of non-0 symbols in v. As before, we will do the proof only for configurations of the form (i) and (ii).

Case (i).1 $v_2 \neq 0$ or $v_1 = 2$.

$$\left(\begin{smallmatrix} 1^{k(v)} & 1 & 0 & 0 & 0^j & v_1 & v_2 & \dots & v_l & 1 \\ & & & & & p & & & & \end{smallmatrix}\right)$$

Regardless of the value of $v_1 \neq 0$
in 1 or 2 steps we reach the next, with $i = 0$ or 1

$$\left(\begin{smallmatrix} 1^{k(v)} & 1 & 0^{j-i} & 0 & 0 & 0^i & v_2 & \dots & v_l & 1 \\ & & & & q & & & & & \end{smallmatrix}\right)$$

Apply $Q(j-i)$

$$\left(\begin{smallmatrix} 1^{k(v)} & 1 & 0^j & 0 & 0 & 0 & v_2 & \dots & v_l & 1 \\ & q & & & & & & & & \end{smallmatrix}\right)$$

Two steps

$$\left(\begin{smallmatrix} 1^{k(v)} & 0 & 0^j & 0 & 0 & 0 & v_2 & \dots & v_l & 1 \\ & p & & & & & & & & \end{smallmatrix}\right)$$

$P(j+2)$

$$\left(\begin{smallmatrix} 1^{k(v)} & 0 & 0^j & 0 & 0 & 0 & v_2 & .. & v_l & 1 \\ & & & & p & & & & & \end{smallmatrix}\right)$$

applying the previous steps
iteratively $k(v) - 1$ times

$$\left(\begin{smallmatrix} 1 & 0 & 0^{k(v)+j+l} & 1 \\ & p & & \end{smallmatrix}\right)$$

Apply $P(k(v) + j + l)$

$$\left(\begin{smallmatrix} 1 & 0 & 0^{k(v)+n-1} & 1 \\ & & & p \end{smallmatrix}\right)$$

we are done with $k = k(v)$ and $k' = 1$

(2)

Case (i).2 $v_2 = 0$ and $v_1 = 1$. Let $g \in \{3, .., l\}$ be the smallest index such that $v_g \neq 0$.

$$\left(\begin{smallmatrix} 1^{k(v)} & 1 & 0 & 0^{j} & 1 & 0 & 0^{g-3} & v_g & \dots & v_l & 1 \\ & & & & p & & & & & & \end{smallmatrix}\right)$$

Two steps

$$\left(\begin{smallmatrix} 1^{k(v)} & 1 & 0 & 0^{j} & 0 & 1 & 0^{g-2} & v_g & \dots & v_l & 1 \\ & & & & b & & & & & & \end{smallmatrix}\right)$$

$B(j+l)$

$$\left(\begin{smallmatrix} 1^{k(v)} & 1 & 0 & 0^{j} & 0 & 1 & 0^{g-2} & v_g & \dots & v_l & 1 \\ & b & & & & & & & & & \end{smallmatrix}\right)$$

One step

$$\left(\begin{smallmatrix} 1^{k(v)} & 1 & 0 & 0^{j} & 0 & 1 & 0^{g-2} & v_g & \dots & v_l & 1 \\ & & p & & & & & & & & \end{smallmatrix}\right)$$

$P(j+l)$

$$\left(\begin{smallmatrix} 1^{k(v)} & 1 & 0 & 0^{j} & 0 & 1 & 0^{g-2} & v_g & \dots & v_l & 1 \\ & & & & p & & & & & & \end{smallmatrix}\right)$$

Repeat last four steps $g-2$ times

$$\left(\begin{smallmatrix} 1^{k(v)} & 1 & 0 & 0^{j} & 0 & 0^{g-2} & 1 & v_g & \dots & v_l & 1 \\ & & & & & & p & & & & \end{smallmatrix}\right)$$

Which reduces to case (i).1

Case (ii)

$$\left(\begin{smallmatrix} 1^{k(v)} & 1 & 0 & 0^{j} & v_1 & v_2 & \dots & v_l & 1 \\ & q & & & & & & & & \end{smallmatrix}\right)$$

Two steps

$$\left(\begin{smallmatrix} 1^{k(v)} & 0 & 0 & 0^{j} & v_1 & v_2 & \dots & v_l & 1 \\ & p & & & & & & & & \end{smallmatrix}\right)$$

$P(j+1)$

$$\left(\begin{smallmatrix} 1^{k(v)} & 0 & 0 & 0^{j} & v_1 & v_2 & \dots & v_l & 1 \\ & & & & p & & & & & \end{smallmatrix}\right)$$

Which reduces to case (i)

(3)

In this way, we have proved that, for (i) and (ii), we will always reach $\left(\begin{smallmatrix} 1 & 0^{n-1+k(v)} & 0 & 1 \\ & & & p \end{smallmatrix}\right)$. For cases (iii) and (iv), we can assure, by symmetry, that the machine will reach $\left(\begin{smallmatrix} 1 & 0^{n-1+k(v)} & 0 & 1 \\ q & & & \end{smallmatrix}\right)$, then:

$$\left(\begin{smallmatrix} 1 & 1 & 0^{n-1+k(v)} & 0 & 1 \\ & q & & & \end{smallmatrix}\right)$$

Two steps

$$\left(\begin{smallmatrix} 1 & 0 & 0^{n-1+k(v)} & 0 & 1 \\ & p & & & \end{smallmatrix}\right)$$

Apply $P(n-1+k(v)+1)$

$$\left(\begin{smallmatrix} 1 & 0 & 0^{n-1+k(v)} & 0 & 1 \\ & & & & p \end{smallmatrix}\right)$$

(4)

Concluding that, for the last two cases, we need just one additional 1 symbol to reach the desired configuration. ■

Proof (of lemma 3). We will prove this by induction over k. For $k = 0$, it is enough to simulate the machine during 7 steps. Now, let us suppose that the assertion is true for k, and let us prove that it is also true for $k + 1$.

$$
\begin{array}{l}
\left({}^{w}2\ 2\ 2\ 2^{k-1}\ \underset{p}{.2}\ 2^{k-1}\ 2\ 2\ 2\ 2\ 2^{w} \right) \\
\qquad\qquad \text{Induction hypothesis} \\
\left({}^{w}2\ 2\ 0\ 0^{k-1}\ .0^{k-1}\ \underset{b}{0}\ 2\ 0\ 2\ 2\ 2^{w} \right) \\
\qquad\qquad \text{Apply } B(2k) \\
\left({}^{w}2\ \underset{b}{2}\ 0\ 0^{k-1}\ .0^{k-1}\ 0\ 2\ 0\ 2\ 2\ 2^{w} \right) \\
\qquad\qquad \text{One step} \\
\left({}^{w}2\ 2\ \underset{p}{0}\ 0^{k-1}\ .0^{k-1}\ 0\ 2\ 0\ 2\ 2\ 2^{w} \right) \\
\qquad\qquad \text{Apply } P(2k) \\
\left({}^{w}2\ 2\ 0\ 0^{k-1}\ .0^{k-1}\ 0\ \underset{p}{2}\ 0\ 2\ 2\ 2^{w} \right) \\
\qquad\qquad \text{One step} \\
\left({}^{w}2\ 2\ 0\ 0^{k-1}\ .0^{k-1}\ \underset{q}{0}\ 0\ 0\ 2\ 2\ 2^{w} \right)
\end{array}
\tag{5}
$$

From this we obtain the "odd" case in one step: $\left({}^{w}2\ 2\ 0\ 0^{k-1}\ .0^{k-2}\ \underset{b}{0}\ 2\ 0\ 0\ 2\ 2\ 2^{w} \right)$.

Continuing from the last step of the former development we obtain the "even" case.

$$
\begin{array}{l}
\qquad\qquad \text{Apply } Q(2k) \\
\left({}^{w}2\ \underset{q}{2}\ 0\ 0^{k-1}\ .0^{k-1}\ 0\ 0\ 0\ 2\ 2\ 2^{w} \right) \\
\qquad\qquad \text{One step} \\
\left({}^{w}2\ 0\ \underset{p}{0}\ 0^{k-1}\ .0^{k-1}\ 0\ 0\ 0\ 2\ 2\ 2^{w} \right) \\
\qquad\qquad \text{Apply } P(2k+2) \\
\left({}^{w}2\ 0\ 0\ 0^{k-1}\ .0^{k-1}\ 0\ 0\ 0\ \underset{p}{2}\ 2\ 2^{w} \right) \\
\qquad\qquad \text{Two steps} \\
\left({}^{w}2\ 0\ 0\ 0^{k-1}\ .0^{k-1}\ 0\ \underset{b}{0}\ 2\ 0\ 2\ 2^{w} \right)
\end{array}
\tag{6}
$$

Proof (of lemma 4).

$$
\begin{array}{l}
\left(\begin{array}{cccccccccc} .2 & 0 & 0 & 0^k & 0 & 0^{n-k-2} & 0 & 0 & 2 \\ & & & & & & & b & \end{array}\right) \\
\qquad \text{Apply } B(n+2) \\
\left(\begin{array}{cccccccccc} .2 & 0 & 0 & 0^k & 0 & 0^{n-k-2} & 0 & 0 & 2 \\ b & & & & & & & & \end{array}\right) \\
\qquad \text{Two steps} \\
\left(\begin{array}{cccccccccc} .2 & 2 & 0 & 0^k & 0 & 0^{n-k-2} & 0 & 0 & 2 \\ & d & & & & & & & \end{array}\right) \\
\qquad \text{Apply } D(n+1) \\
\left(\begin{array}{cccccccccc} .2 & 2 & 0 & 0^k & 0 & 0^{n-k-2} & 0 & 0 & 2 \\ & & & & & & & & d \end{array}\right) \\
\qquad \text{Two steps} \\
\left(\begin{array}{cccccccccc} .2 & 2 & 0 & 0^k & 0 & 0^{n-k-2} & 0 & 2 & 2 \\ & & & & & & b & & \end{array}\right) \\
\qquad \text{Apply } B(n+2) \\
\left(\begin{array}{cccccccccc} .2 & 2 & 0 & 0^k & 0 & 0^{n-k-2} & 0 & 2 & 2 \\ & b & & & & & & & \end{array}\right) \\
\qquad \text{One step} \\
\left(\begin{array}{cccccccccc} .2 & 2 & 0 & 0^k & 0 & 0^{n-k-2} & 0 & 2 & 2 \\ & & p & & & & & & \end{array}\right) \\
\qquad \text{Apply } P(n+1) \\
\left(\begin{array}{cccccccccc} .2 & 2 & 0 & 0^k & 0 & 0^{n-k-2} & 0 & 2 & 2 \\ & & & & & & & p & \end{array}\right) \\
\qquad \text{Two steps} \\
\left(\begin{array}{cccccccccc} .2 & 2 & 0 & 0^k & 0^{n-k-2} & 0 & 2 & 0 & 2 \\ & & & & & b & & & \end{array}\right) \\
\qquad \text{Repeat four last steps } n-k-1 \text{ times} \\
\left(\begin{array}{cccccccccc} .2 & 2 & 0^k & 0 & 2 & 0 & 0^{n-k-2} & 0 & 2 \\ & & & b & & & & & \end{array}\right)
\end{array}
\tag{7}
$$

Now we prove the second part.

$$\left(.2\ 0\ 0\ 0\ 0^{n-k-1}\ 0^{k-1}\ 0\ \underset{b}{0}\ 2\right)$$

Apply $B(n+2)$

$$\left(\underset{b}{.2}\ 0\ 0\ 0\ 0^{n-k-1}\ 0^{k-1}\ 0\ 0\ 2\right)$$

Two steps

$$\left(.2\ 2\ \underset{d}{0}\ 0\ 0^{n-k-1}\ 0^{k-1}\ 0\ 0\ 2\right)$$

Apply $D(n+1)$

$$\left(.2\ 2\ 0\ 0\ 0^{n-k-1}\ 0^{k-1}\ 0\ 0\ \underset{d}{2}\right)$$

One step

$$\left(.2\ 2\ 0\ 0\ 0^{n-k-1}\ 0^{k-1}\ 0\ 0\ \underset{q}{2}\right)$$

Apply $Q(n+1)$ (8)

$$\left(\underset{q}{.2}\ 2\ 0\ 0\ 0^{n-k-1}\ 0^{k-1}\ 0\ 0\ 2\right)$$

Two steps

$$\left(.2\ 0\ 2\ \underset{d}{0}\ 0^{n-k-1}\ 0^{k-1}\ 0\ 0\ 2\right)$$

Apply $D(n)$

$$\left(.2\ 0\ 2\ 0\ 0^{n-k-1}\ 0^{k-1}\ 0\ 0\ \underset{d}{2}\right)$$

Repeat last two steps $n-k-1$ times

$$\left(.2\ 0\ 0^{n-k-1}\ 2\ 0\ 0^{k-1}\ 0\ 0\ \underset{d}{2}\right)$$

Two steps

$$\left(.2\ 0\ 0^{n-k-1}\ 2\ 0\ 0^{k-1}\ \underset{b}{0}\ 2\ 2\right)$$

■