Turing Machines with Two Letters and Two States

Maurice Margenstern

Université Paul Verlaine — Metz, LITA, EA 3097
UFR MIM, Campus du Saulcy, 57045 METZ Cédex, France
margens@univ-metz.fr

In this paper we provide a survey of the technique that allows giving a simple proof that all Turing machines with two letters and two states have a decidable halting problem. The result was proved by L. Pavlotskaya in 1973.

1. Introduction

The notion of Turing machines appeared in 1936 in the famous paper by Alan M. Turing [1]. Turing's notion has since given rise to computer science. A few months later a paper by Emil Post appeared, describing the same object. Post's paper gives a very precise and simple description of the Turing machine which is more or less what is used today.

In this paper we follow the classical notion of a Turing machine. It is a device consisting of an infinite tape of squares indexed by $\mathbb{Z}$, a head that looks at the *scanned* square and which is in a *state* belonging to a fixed finite set of states. The index x of a square is called its *address* and we shall also say the *square* x for the *square with address* x. Each square contains a symbol belonging to a fixed finite set of *letters* also called the alphabet. Among these letters, a symbol is distinguished and called the *blank*. The device also contains a finite sequence of *instructions* described as a quintuple: two data constitute the *input* of the instruction, the scanned symbol, and the state of the head; three data constitute the *output*, the letter written by the head in place of the scanned letter, the new state that replaces the current state of the head, and the *move* performed by the head. After executing the instruction, the next cell to be scanned is to the left or right of the previously scanned cell, or is the same cell. A Turing machine with ℓ letters and s states is called an $s\times\ell$*-machine*. Note that classical Turing machines are *deterministic*, meaning that the input of two distinct instructions are different. The symbols ℓ and s have this meaning throughout this paper. The *computation* of a Turing machine is defined by the sequence of successive configurations obtained from an initial configuration, where all but a finite number of cells are blank.

https://doi.org/10.25088/ComplexSystems.19.1.29

The computation continues until a possible final configuration which occurs, in Turing's definition, after a halting state was called. In Turing's definition too, the result of the computation is what is written on the tape once the machine halted.

In [1] the description is more sophisticated and the basic properties of the set of Turing machines are given: the existence of universal machines, which, by definition, are able to simulate any Turing machine and the existence of a limit to the model as a problem that cannot be solved by any Turing machine. This problem is now known as the *halting problem*. It is an essential feature of Turing machines that their computation may halt or not and that to determine whether this is the case or not, which is the halting problem, turns out to be undecidable: there is no algorithm to solve it.

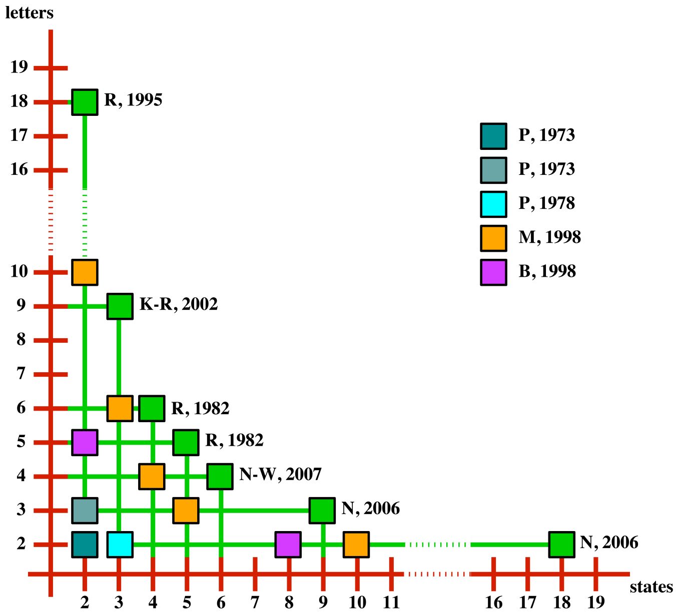

Figure 1. The small universal Turing machines and those with a decidable halting problem (in blue). The figure indicates the best known results only. The machines indicated with an orange or a purple square simulate the iterations of the $3x+1$ function.

Later, in the 1950s, Claude Shannon raised the problem of what is now called the *descriptional complexity* of Turing machines: how many states and letters are needed in order to get universal machines? A race ensued to find the smallest Turing machine that was stopped by Yurii Rogozhin's result in 1982 [2]. Seven universal Turing machines were given, one in each of the following sets of machines: 2×21, 3×10, 4×6, 5×5, 7×4, 11×3, and 24×2 (see Figure 1). Nothing changed during the next 10 years. In 1992, Rogozhin improved

https://doi.org/10.25088/ComplexSystems.19.1.29

his 11×3 universal machine into a 10×3. In 1995, he proved that there are universal 2×18-machines. After an exchange of mails with the author, who had found a 2×21-machine, in 1998 Rogozhin found a 22×2 universal machine. In 2001, Claudio Baiocchi found a universal 19×2-machine. Then, in 2002, Rogozhin and Manfred Kudlek found a universal 3×9-machine. Recently, in 2006, Turlough Neary found a universal 18×2-machine and in 2007, Neary and Damien Woods found a universal 6×4-machine. Note that all of the machines mentioned from 1995 onwards were found at the occasion of a forthcoming edition of Machines, Computations, and Universality (MCU) conferences organized by the author.

Remember that all of these machines are universal in the sense that they simulate any Turing machine starting from a finite configuration and that when their computation stops the halting instruction is not taken into account when counting the instructions.

Turing machines on infinite configurations were also studied. The immortality problem, first studied by Philipp K. Hooper in 1966, consists of finding an initial infinite configuration on which the Turing machine never halts, whatever the initial state [3]. Other models of discrete computations were studied in this regard, in particular cellular automata and planar Turing machines. There, it turned out that by using initial infinite configurations it was possible to reduce the number of states and letters in order to obtain a universal device.

Now, we have to be careful about universality in this context: what does it mean? The reason is that if we allow arbitrary initial infinite configurations, then the halting problem becomes solvable. It is enough to encode the characteristic function of the set of all n for which the n^{th} Turing machine with input n halts on the tape of the Turing machine! This is why, during a certain time, initial infinite configurations were required to be ultimately periodic. This means that outside some finite interval, what remains of the tape on the left- and on the right-hand side is periodic, the periods being possibly different on each side of the tape. The rest of the simulation is the same as in the case of a classical Turing machine. Note that the classical situation is a particular case of this definition: the period is 1 and the content of the square is given; it must be the blank. This extended definition of universality is called *weak universality*. Although this generalization is very natural, there is a sharp difference from the classical case. The results indicated later also point to this difference.

Not everybody makes use of the term weak universality. Many a researcher does not think it that important to make a distinction on properties of the initial configuration leading to universal computations and, as an example, calls rule 110 universal.

In this context, the works of Stephen Wolfram on cellular automata inspired research that reached an important result: the weak universality of rule 110, [4, 5] an elementary cellular automaton. The corollary was the construction of very small weakly universal Turing machines, already announced in 2002, with significantly fewer instruc-

https://doi.org/10.25088/ComplexSystems.19.1.29

tions than the machines in Figure 1: eight instructions in 2005 [5] and five instructions in 2007 [5, 6]. Another difference is that the halting of these very small machines is not obtained by a halting instruction. This point about the way of halting was already raised in [7] where a universal planar Turing machine with eight instructions is constructed that does not halt on a specific instruction. It was also raised in the construction of reversible computations, first of cellular automata and then of Turing machines, which forced people to slightly change the notion of halting: in this frame, it could no more be characterized by a unique configuration. In 2003 the author, in a joint work with Ludmila Pavlotskaya, proved that a Turing machine with four instructions, even coupled with a finite automaton, has a decidable halting problem [8]. In the same paper, the authors proved that there is a Turing machine with five instructions and a particular finite automaton such that the resulting couple is universal. This can be compared with the result in [6] established after the well-known challenge launched by Wolfram. The result in [6] is stronger than that of [8] as in [6] the tape of the Turing machine is initially fixed. Its initial configuration is not exactly periodic, but it is "regular" in the sense that the infinite word written on the tape can be generated by a Muller finite automaton.

In this paper, we are interested by the decidability side of the question, about which very little is known [9]. Marvin Minsky mentions an unpublished proof by him and one of his students in [10] as unreadable because it involves a huge number of cases. The first readable proof was published by Pavlotskaya and states the following.

Theorem 1. (Pavlotskaya [11]) The halting problem is decidable for any 2×2-Turing machine.

Later in [12], Kudlek proved the same result in a very different way, classifying the machines according to what the computations produce, thus including machines that never halt. It is interesting to note that all computations are more or less trivial except one case, putting aside the trivial permutations and symmetries on states and letters. This case was also found in [8] where it was proved to have an exponential time computation on a sequence of patterns of the form 1^n.

In this paper, we give a simple proof of Theorem 1 that is based on an analysis of the motion of the Turing machine head on its tape. Section 2 deals with this analysis. In Section 3 we prove Theorem 1.

2. Motion of the Turing Machine Head on Its Tape

In this section, we fix notions and notations for the rest of the paper. We denote by t the current *time* of execution, t being a non-negative integer. Usually, the initial time is denoted by t_0 and, most often,

https://doi.org/10.25088/ComplexSystems.19.1.29

$t_0 = 0$. The current instruction is performed at time t and we get the result at time $t+1$ when the next instruction is performed.

2.1 Two Position Lemmas

Let $h(t)$ be the position of the head on the tape at time t. We denote by $\eta(t)$ the state of the head at time t and by $\sigma(t, x)$ the content of the square x at time t. By definition, $\sigma(t, x) \ldots \sigma(t, x+L)$ is the word whose letters consist of the contents of the squares with addresses from x to $x+L$ at time t. This word will also be called the interval $[x, x+L]$ of the tape at time t. We define ℓ_0 and r_0 to be the left- and right-hand side ends of the smallest interval that contains all the nonblank squares of the tape together with the square scanned by the head at time 0, the initial time. We define two functions ℓ and r to indicate the limits of the current configuration at time t as

$$\begin{aligned}
\ell(0) &= \ell_0, \\
r(0) &= r_0, \\
\ell(t+1) &= \min(h(t+1), \ell(t)), \\
r(t+1) &= \max(h(t+1), r(t)).
\end{aligned}$$

In other terms, $\ell(t+1) < \ell(t)$ if and only if $h(t) = \ell(t)$ and the machine performs an instruction with a move to the left at time t. Symmetrically, $r(t+1) > r(t)$ if and only if $h(t) = r(t)$ and the machine performs an instruction with a move to the right at time t. The configuration at time t is denoted by C_t.

The functions ℓ and r allow us to define the notion of the head *exiting* the limits of the current configuration. Define the times of exit E as

$$\begin{aligned}
E(0) &= 0 \\
E(i+1) &= \min\{t \mid h(t) < \ell(E(i)) \vee h(t) > r(E(i))\}.
\end{aligned}$$

From this definition, when $E(i) \leq t < E(i+1)$, we have:

$$[\ell(t), r(t)] \subseteq [\ell(E(i)), r(E(i))]$$

and, by *abus de langage*, we call $E(i)$ the i^{th} exit.

Now, such an exit is called a *left-* or *right-hand side* exit, depending on whether $\ell(E(i)) \neq \ell(E(i-1))$ or $r(E(i)) \neq r(E(i-1))$. This allows us to define functions LE and RE to denote the i^{th} left-hand or i^{th} right-hand side exit as

$$\begin{aligned}
LE(0) &= 0, \\
LE(i+1) &= \min\{t \mid h(t) < \ell(LE(i))\}, \\
RE(0) &= 0, \\
RE(i+1) &= \min\{t \mid h(t) < r(RE(i))\}.
\end{aligned}$$

https://doi.org/10.25088/ComplexSystems.19.1.29

The motion of the Turing machine head on its tape consists of a sequence of consecutive runs over an interval during which the head moves in the same direction each time. Let us call such a run *sweeping* and note that a sweeping may be finite or infinite. Also note that in the case of an infinite sweeping the motion of the machine is ultimately periodic. After a certain time the head encounters an infinite interval of blank cells and, because the move at each step is constant, it always scans a blank and the only changing parameter is its state. As the number of states is finite, there must be a repetition and this interval between two occurrences of the same state is a period of the motion.

Any sweeping has at least one *half-turn*, that is, a time and a position such that the next move is in the opposite direction of the previous move. The half-turn is called a *left-* or *right-hand side* half-turn, depending on whether it occurs on the left- or right-hand side of the sweeping. Now, we say that an exit is *extremal* if and only if the new limit of the configuration which it defines is a half-turn. We now define the functions *LEE* and *REE* to denote the left- and right-hand side extremal exits, respectively:

$$\begin{aligned} LEE(0) &= 0, \\ LEE(i+1) &= \min\{t \mid h(t) < \ell(LEE(i)) \wedge h(t+1) > h(t)\}, \\ REE(0) &= 0, \\ REE(i+1) &= \min\{t \mid h(t) > r(REE(i)) \wedge h(t-1) < h(t)\}. \end{aligned}$$

We now have the following first property: a finite interval $[a, b]$ of the tape is a *trap zone* for the machine starting from a time t_1 if and only if $a \leq h(t) \leq b$ for all t with $t \geq t_1$. We say that $[a, b]$ is a trap zone for the machine if there is a time t_1 starting from which it is a trap zone. Here is an easy lemma to test whether a given finite interval is a trap zone for the machine.

Lemma 1. Let $[a, b]$ be a finite interval of the tape of the machine M. Then, we know whether $[a, b]$ is a trap zone starting from a given time t_1 after at most $n^{b-a+1}(b-a+1)\,s+1$, where n is the size of the alphabet of the machine and s is the number of its states.

The obvious proof is left to the reader.

From Lemma 1, we have the following corollary, whose trivial proof is also left to the reader.

Corollary 1. The functions E, ℓ, and r are recursive and the finiteness of the domain of definition of E is recursively enumerable.

Now, we turn to an important lemma that relies on the same idea as Lemma 1.

https://doi.org/10.25088/ComplexSystems.19.1.29

Lemma 2. (Margenstern [13, 14]) Let M be a Turing machine and assume that there are two right-hand side exits at times t_1 and t_2, with $t_1 < t_2$ and that there is an address a such that:

i. $\forall\, t \in [t_1, t_2], h(t) \geq a$

ii. let $\delta = h(t_2) - h(t_1) \geq 0$; then: $\forall\, x \in [a, h(t_1)], \sigma(t_1, x) = \sigma(t_2, x + \delta)$

iii. $\eta(t_1) = \eta(t_2)$.

Then, the sequence of instructions on the time interval $[t_1, t_2 - 1]$ is an execution pattern that is endlessly repeated and we say that the motion of the machine is *ultimately periodic*.

Proof. The conditions of the lemma are illustrated by Figure 2. The statement assumes that between times t_1 and t_2 the head never goes to the left of square a, that at times t_1 and t_2 the head of the machine is under the same state **u**, and that the words of the interval $[a, h(t_1)]$ at time t_1 and of the interval $[a + \delta, h(t_2)]$ at time t_2 are the same.

Let C_{t_1} be the configuration at time t_1. Now, imagine that at time t_1 we replace the interval $]-\infty, a-1]$ by the same interval with all squares filled up with the blank. Let C_1' be this new configuration. Then, the motion of the Turing machine on the tape between t_1 and t_2 is the same, whether it starts from C_{t_1} or C_1'. Now, we clearly can repeat the same for C_{t_2} being replaced by C_2' where all squares on the left of $a + \delta$ are replaced by the blank for times t_2 and $t_2 + t_2 - t_1$. And so, the same motion is repeated during the time interval $[t_2, 2\,t_2 - t_1]$. And this can be repeated endlessly by an easy induction left to the reader. □

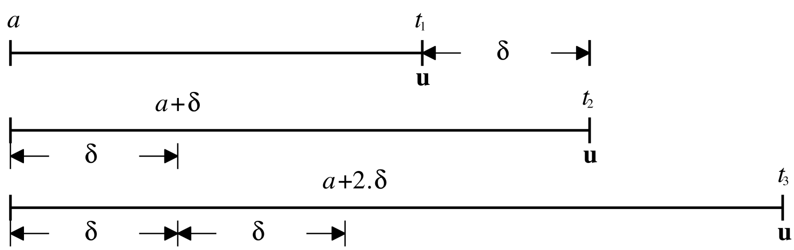

Figure 2. Illustrating the assumptions of Lemma 2 and its conclusion.

Note that the conditions of the assumption of Lemma 2 are recursively enumerable.

https://doi.org/10.25088/ComplexSystems.19.1.29

2.2 Additional Tools

The occurrence of a halting instruction is always recursively enumerable. And so, to prove that the halting problem is decidable, it is enough to focus on the nonhalting situations and prove that the general nonhalting situation is also recursively enumerable. This happens each time we have an algorithmic way to decide, after a certain time which may depend on the considered instance of the problem, that the machine will not halt. Later on, we shall only consider the nonhalting situations.

If the machine does not halt but instead remains within a finite interval, Lemma 1 indicates that the motion of the machine is ultimately periodic: the same finite sequence of instructions is endlessly repeated.

From now on, we assume that the Turing machine head traverses an infinite interval, which implies, of course, that the machine does not halt. If the motion involves an interval of time $[t, t+s]$ such that $t+i$, for $i \in \{0 .. s\}$ is an instance of a right-hand side exit, then there will be two times $t+h$ and $t+k$, with $h < k$, such that the head of the machine is under the same state at these times. As a right-hand side instruction is performed at these times and because the head always scans a blank due to the exit, the same sequence of instructions performed between $t+h$ and $t+k-1$ will be repeated endlessly, involving a motion which is ultimately periodic. Of course, we can perform a similar argument for successive $s+1$ times at which a left-hand side exit occurs.

If we do not have this situation, necessarily, there are infinitely many extremal exits. In general, we have three cases.

i. There are infinitely many right-hand side extremal exits and finitely many left-hand side ones.

ii. There are infinitely many left-hand side extremal exits and finitely many right-hand side ones.

iii. There are infinitely many left-hand side extremal exits and infinitely many right-hand side ones.

Consider case i. We may assume that, after a time t_1, there are no more left-hand side exits. Denote by $lmp(j)$ the leftmost position of the machine head between $REE(j)$ and $REE(j+1)$. Call *lmp-time* for $lmp(j)$ the first time after $REE(j)$ that the machine head scans the square $lmp(j)$. Let a be the address of the last left-hand side exit. Then, $lmp(j) \geq a$ for all j. Let $\lambda = \liminf_{j\to\infty} lmp(j)$. Of course $\lambda \geq a$ whether λ is finite or not. Let $a_m = \inf_{j\geq m} lmp(j)$. Because $a_m \geq a$, and as a_m is an integer, a_m is reached: there is an integer $n_m \geq j$ such that $lmp(n_m) = a_m$ and so, $lmp(j) \geq lpm(n_m)$ for all $j \geq n_m$. Note that the position $lmp(n_m)$ is absolute in the following sense: after $REE(n_m)$, the machine never goes to the left of the square $lmp(n_m)$.

https://doi.org/10.25088/ComplexSystems.19.1.29

The sequence of the $lmp(n_m)$ is nondecreasing and we may assume that the sequence of the n_m is increasing.

Note that case ii can be dealt with symmetrically. We can define $rmp(j)$, the rightmost position between two consecutive left-hand side extremal exits and rmp-time, the first time that $rmp(j)$ is reached after $LEE(j)$.

2.3 Laterality and Color of an Instruction

Consider an instruction of the program of the Turing machine M. The instruction can be written $\mathbf{i}xyM\mathbf{j}$ where x is the scanned symbol at the current time, the current state of the machine is $\mathbf{i}$, and y is the symbol written by the machine head in place of x. Then, $\mathbf{j}$ is the new state taken by the machine head and M is the move performed by the head just after writing y: M is L, D, or S depending on whether the move makes the head go one square to the left, to the right, or to stay on the same square. Call the *color* of an instruction $\mathbf{i}xyM\mathbf{j}$ the triple xMy and call *laterality* of the instruction the value of M. Note that when we have a stationary instruction I, that is, $M = S$, the next instruction is either stationary or not. Repeating this argument, either there is a cycle of stationary instructions, or a halting or, after a certain sequence of consecutive stationary instructions, the next instruction is not stationary. If we are in the first or second case, we say that I is a *blocking* instruction. Consequently, if I is not blocking, it is ultimately followed by a nonstationary instruction J: by definition, the laterality of I is that of J. We say that a machine is *unilateral* if and only if all of its instructions have the same laterality: either L or R. By this we mean that when a stationary instruction is not blocking, its laterality is defined by one of the other instructions. We also consider that a halting instruction is a particular case of a blocking instruction.

Lemma 3. The halting problem is decidable for any unilateral Turing machine.

Proof. We may assume that all of the instructions are right-hand side. Because the occurrence of a blocking instruction is recursively enumerable it is enough to wait for its possible execution and we may assume that the machine head always goes to the right. It eventually exits from the right-hand side limit of the initial configuration and, later on, the machine head only sees a blank in the scanned cell. If after $s+1$ steps the machine does not find a blocking instruction, it will fall under two identical states and the sequence of instructions performed between the times of two consecutive occurrences of this state will be endlessly repeated meaning that the machine will not halt. □

However, note that if the machine has two heads, Lemma 3 is no longer true when stationary instructions are allowed [15].

Say that a Turing machine M is *erasing to the left* if all its left-hand side instructions write a blank. We have the following lemma.

https://doi.org/10.25088/ComplexSystems.19.1.29

Lemma 4. The halting problem is decidable for a Turing machine that is erasing to the left.

Note that if M is a unilateral machine whose laterality is L, we obtain another unilateral machine by changing L to R in all of the instructions. This operation is called *lateral symmetry*. In particular, Lemma 2 has a left-hand side version which is obviously obtained by lateral symmetry. We shall use this version only without further justification.

Proof of Lemma 4. We consider that the Turing machine M is erasing to the left and that it does not halt. We may assume that we are not in a trap zone nor in the situation of a sequence of times $t, \ldots, t+s$ of exit on the same side. Accordingly, we are in one of the cases defined in Section 2.2.

Case i. We may assume that after a time t_1, all exits are right-hand side.

Consider the sequence of absolute *lmp* defined in Section 2.2, which, here also, we denote by $lmp(n_m)$, with increasing n_m.

Consider the configurations $C_{t_{n_1}}, \ldots, C_{t_{n_s}}$ where t_{n_m} is the first time between $REE(n_m)$ and $REE(n_m+1)$ that $lmp(n_m)$ is reached by the machine head. For two of these configurations, the machine head will be under the same state. By definition of $C_{t_{n_m}}$, the head is on the $lmp(n_m)$ at its *lmp*-time. Now, because the machine writes blanks only when it performs a left-hand side instruction, there are only blanks on the right-hand side of the $lmp(n_m)$ at this time, from the address $lmp(n_m)+1$ up to infinity. Let h and k be the indices for which the state of the head is the same at t_{n_h} and t_{n_k}, with $h < k$. Because the $lmp(n_m)$ are absolute, the machine never goes to the left of $lmp(n_h)$ between t_{n_h} and t_{n_k}. And so, the sequence of instructions between these two times is repeated endlessly. Now, the occurrence of two configurations with these conditions is also recursively enumerable.

Case ii. For this case we can prove a stronger result and distinguish it as Sublemma 1.

Sublemma 1. Assume that for a Turing machine M, there are infinitely many left-hand side extremal exits and finitely many right-hand side ones and that all of the left-hand side instructions write the same letter y. Then the halting problem of M is decidable.

Proof of Sublemma 1. We assume that we are at a time after t_1, starting from which there are only left-hand side exits. Under these assumptions, the configuration of the i^{th} left-hand side exit after t_1 is

$$\mathbf{u}_i\, y^{n_i}\, W_i$$

https://doi.org/10.25088/ComplexSystems.19.1.29

where $\mathbf{u}_i$ is the state of the head at the exit, n_i is the number of consecutive y written by the head before the exit, and W_i is a word on the alphabet of the machine that does not start with y when not empty. Because there are no more right-hand side exits, the right-hand side limit of the configuration is always $r(t_1)$.

Denote by a_i the address of the rightmost y in the word y^{n_i}. We get that $a_i \leq r(t_1)$ for all i. As all the left-hand side instructions write y on the tape, it is plain that $a_{i+1} \geq a_i$. Let $a = \limsup_{i \to \infty} a_i$. We have that $a \leq r(t_1)$ and a is reached, because the a_i are all integers. If a is reached at a time t_2, then, after $s+1$ exits after t_2, we can see two exits which satisfy the assumptions of the left-hand side version of Lemma 2. Now, the occurrence of two such exits is recursively enumerable.

Case iii. Because there are infinitely many left- and right-hand side exits, there are infinitely many extremal right-hand side exits such that the next exit is on the left-hand side. And so, there is an increasing sequence of n_m such that $E(n_m)$ is a right-hand side exit and that $E(n_m + 1)$ is a left-hand side one. Because the left-hand side instructions write the blank only, at time $E(n_m + 1)$, the tape of the machine contains blanks only. Now, looking at the configurations at the times $E(n_1 + 1), \ldots, E(n_{s+1} + 1)$, two of them are identical: the tape is empty and the head is under the same state. This induces a sequence of endlessly repeating instructions and the occurrence of two such configurations is recursively enumerable. The motion of the machine is ultimately periodic because if there is no shift in the position of the machine head on the two detected configurations with an empty tape, the motion remains trapped in a finite interval. If there is a shift, the motion goes infinitely on one side of the tape only. And so, in all three situations, there cannot be infinitely many left- and right-hand side exits.

This means that Case iii does not occur for the considered machines.

Accordingly, the nonhalting of M is recursively enumerable in both possible cases of the motion of the machine head, which proves the lemma. □

3. Proof of Theorem 1

The proof relies on the following property.

Lemma 5. (Pavlotskaya [11]) Let M be a Turing machine on the alphabet $\{0, 1\}$ such that M has a single instruction whose laterality is L. Then the halting problem of M is decidable.

https://doi.org/10.25088/ComplexSystems.19.1.29

Theorem 1 is an immediate corollary of Lemma 5. Indeed, if a 2×2-machine has no halting instruction, it never halts. So, it has at least one halting instruction. On the others, it has at most one instruction whose laterality is not shared by the others. By lateral symmetry, we may assume that the laterality with a unique instruction is L.

Proof of Lemma 5. From Lemma 4, we may rule out the colors $x\,L\,0$ for the unique left-hand side instruction. And so, we remain with the colors $x\,L\,1$ with $x = 0$ or $x = 1$.

Color $1\,L\,1$. In this case, if the machine head reads a blank, it does not move to the left. Consider the case of a stationary instruction which is unique: otherwise we have a unilateral machine for which Lemma 3 applies. Then, if the color of the instruction is $0\,S\,0$, this blank square is a trap zone or it calls a right-hand side instruction. If the stationary instruction has the color $0\,S\,1$, and because we assumed that the machine has a single instruction with the laterality L, the stationary instruction calls the right-hand side one and so the machine goes to the right.

In all cases that the machine head reads a blank, if it does not halt or if it is not stuck in the same place, it goes to the right. In particular, if there is a right-hand side exit, the machine head goes to the right forever. Accordingly, we may assume that all exits are on the left-hand side. But now we are under the assumptions of Sublemma 1 as all left-hand side instructions write a 1. And so, we know that in this case, the nonhalting is recursively enumerable.

Color $0\,L\,1$. This time, if the machine reads a 1, it goes to the right.

Indeed, it cannot go to the left because one of the remaining nonhalting instructions is on the right-hand side and the other is either on the right-hand side or stationary. If the instruction is stationary, it is of the form $1\,S\,1$ or $1\,S\,0$. If it is stationary, and because we assume that there is a single instruction with the laterality L, a stationary instruction of the color $1\,S\,1$ reading a 1 will either keep the head on this square or call the right-hand side instruction. Now, if the stationary instruction is of the color $1\,S\,0$, and because we assume that there is a single instruction with the laterality L, it also necessarily calls the right-hand side instruction as the machine is not assumed to be unilateral.

If the new state of an instruction is the same as its current one we say that the instruction is *stable*. Otherwise, we call it *unstable*.

First, assume that the left-hand side instruction is unstable. Then, if it scans the square a at time t, it scans the square $a+1$ at time at most $t+4$, unless it is blocked in between.

Indeed, if a right-hand side instruction is performed, then $a+1$ is reached at time $t+3$. Otherwise, we have several cases. If the head performs a stationary instruction, either the head is blocked or it calls a right-hand side instruction again and $a+1$ is reached at time $t+2$. If the head performs the left-hand side instruction at time t, we have

https://doi.org/10.25088/ComplexSystems.19.1.29

the configuration $\epsilon_1 \bullet 0\, \epsilon_2$, where $\bullet$ represents the position of the head at this time. Then we have

$$(\nabla) \qquad \epsilon_1 \bullet 0\, \epsilon_2 \to \bullet\epsilon_1\, 1\, \epsilon_2 \to^{\alpha} \epsilon_1' \bullet 1\, \epsilon_2 \text{ with } \alpha \in \{1, 2\},$$

with α indicating the number of instructions applied by the machine. From the given analysis, scanning 1 in the square a, if the machine is not blocked in between, its head reaches $a+1$ at most two steps later.

Accordingly, if the machine does not halt, it goes forever to the right. Now we want to find out if that motion is predictable. From Section 2.2 we may assume that there are infinitely many extremal right exits. At an extremal exit the head arrives on a blank on which it performs a half-turn, meaning that the sequence (∇) is repeated, with $\epsilon_2 = 0$. Now, it is plain that considering $REE(1)$, $REE(2)$, and $REE(3)$, we can find two exits among them for which ϵ_1 is the same. Accordingly, in between the two times, we have a sequence of instructions that is endlessly repeated. Note that there are at most s steps between $REE(i)$ and $REE(i+1)$ but s may be very large.

At last, we remain with the case of a stable left-hand side instruction.

Note that if there is a left-hand side exit, it is performed by the left-hand side instruction and, since it is stable, the head goes infinitely to the left.

And so, we may assume that there are no left-hand side exits. Again, from the study of Section 2.2, we may assume that we have infinitely many extremal right-hand side exits. From the stability of the left-hand side instruction, we conclude that $lmp(i)$ is the position of the rightmost 1 of the tape at $REE(i)$. Because we assume there are no left-hand side exits, there is an address a such that $lmp(i) \geq a$ for all i.

Consider the sequence of absolute lmp defined by $lmp(n_m)$ with increasing n_m. From the previous remark about the stability of the left-hand side instruction, there must be a 1 in the interval of the tape $\left[lmp\left(n_k\right), h\left(REE\left(n_k + 1\right)\right)\right]$.

Now we consider $\lambda_i = h(REE(i)) - lmp(i)$. We know that $\lambda_k > 0$ for all k. Let $\lambda = \liminf_{k\to\infty} \lambda_{n_k}$. There are two cases, depending on whether $\lambda < +\infty$ or $\lambda = +\infty$. In the latter case, λ is a natural number.

When $\lambda < +\infty$, considering $REE(n_1), \ldots, REE\left(n_{\lambda+1}\right)$, we find two indices h and k such that $\lambda_h = \lambda_k$. Now, at the times j_{n_α}, with $\alpha \in \{h, k\}$, of lmp-time at $lmp(n_\alpha)$ we have the same configuration from the address $lmp(n_\alpha)$ and to its right-hand side at time j_{n_α}, also because the state at time j_{n_α} is always the same for both values of α. Indeed, this latter property follows from the fact that we have a single left-hand side instruction. Accordingly, assuming $h < k$, the same sequence of instructions between $REE\left(n_h\right)$ and $lmp\left(n_k\right)$ is endlessly repeated.

https://doi.org/10.25088/ComplexSystems.19.1.29

Now we consider the case when $\lambda = +\infty$. In this case we assume that λ_{n_k} is big enough. On a large interval of ones, the machine is unilateral and because we assume that it does not halt, its motion is ultimately periodic. Since the state at the time of an $lmp(j)$ is always the same, the sequence of instructions on an interval of ones is the beginning of the same infinite sequence ω which is periodic starting from a certain rank. We may assume that λ_{n_k} is big enough to contain at least one complete period of this sequence. Define $m_i = h(REE(i+1)) - h(REE(i))$ as the length of the interval of zeros traversed by the head starting from the first exit to the right after $h(REE(i))$ until the next right-hand side half-turn. If a 1 is written during the period of the motion on ones or on the m_i zeros, we have infinitely many situations when $\lambda_i \leq 2\,s$, because the period cannot be greater than the number of states. Accordingly, $\liminf_{k\to\infty} \lambda_{n_k} < +\infty$, a contradiction with our assumption and the 1 written after the *lmp*-time of $lmp(n_k)$ is written by the head in the aperiodic part of its motion on the interval of ones. By possibly taking a subsequence of the λ_{n_k}, we may assume that $\lambda_{n_k} < \lambda_{n_{k+1}}$.

Now let w be the length of the smallest period of ω and consider the times when the head reaches $lmp(n_k)$ at its *lmp*-time for $k \in [1 \mathinner{..} w+1]$. For two of them, say j_1 and j_2, the state under which the head reaches $h(REE(n_{j_\alpha}))$, $\alpha \in \{1, 2\}$ while coming from $lmp(n_{j_\alpha})$ after its *lmp*-time is the same, say $\mathbf{u}$, and it has the same place in the period of ω. Because $\lambda_{n_{j_\alpha}} > w$, and assuming $j_1 < j_2$, we may write $\lambda_{j_2} = \lambda_{j_1} + b.w$, for some integer b.

Now, the sequence of right-hand side half-turns in the tape interval $[REE(j_1+1), REE(j_2+1)[$ is endlessly repeated. The words of the tape defined by the intervals $[lmp(j_\alpha), h(REE(j_\alpha))]$ at the time of the first right-hand side exit after $REE(j_\alpha)$ have a common prefix and a common suffix, the rest being a word of the form W^{h_α} with $h_2 = h_1 + b$.

Because this situation is recursively enumerable, it is also the case when we have $\liminf_{k\to\infty} \lambda_{n_k} = +\infty$ to complete the proof. □

Acknowledgment

The author is very much in debt to Stephen Wolfram from asking him to write this paper. He is also very much in debt to the anonymous referees whose precious remarks allowed him to improve the paper.

References

[1] A. M. Turing, "On Computable Real Numbers, with an Application to the Entscheidungsproblem," *Proceedings of the London Mathematical Society,* **42**(2), 1936 pp. 230-265.

[2] Y. V. Rogozhin, "Sem' universal'nykh mashin T'juringa," *Matematicheskie Issledovaniya*, **69**, 1982 pp. 76-90 (Seven Universal Turing Machines) (in Russian).

[3] P. K. Hooper, "The Undecidability of the Turing Machine Immortality Problem," *Journal of Symbolic Logic*, **31**(2), 1966 pp. 219-234.

[4] M. Cook, "Universality in Elementary Cellular Automata," *Complex Systems*, **15**(1), 2004 pp. 1-40.

[5] S. Wolfram, *A New Kind of Science*, Champaign, IL: Wolfram Media, Inc., 2002.

[6] A. Smith, "Wolfram's 2,3 Turing Machine Is Universal," *Complex Systems*, to appear. (Aug 12, 2010) http://www.wolframscience.com/prizes/tm23/solved.html

[7] L. Priese, "Towards a Precise Characterization of the Complexity of Universal and Nonuniversal Turing Machines," *SIAM Journal of Computation*, **8**(4), 1979 pp. 508-523.

[8] M. Margenstern and L. Pavlotskaya, "On the Optimal Number of Instructions for Universal Turing Machines Connected with a Finite Automaton," *International Journal of Algebra and Computation*, **13**(2), 2003 pp. 133-202.

[9] M. Margenstern, "Frontier between Decidability and Undecidability: A Survey," *Theoretical Computer Science*, **231**(2), 2000 pp. 217-251.

[10] M. L. Minsky, *Computation: Finite and Infinite Machines*, Englewood Cliffs, NJ: Prentice Hall, 1967.

[11] L. M. Pavlotskaya, "Razreshimost' problemy ostanovki dlja nekotorykh klassov mashin T'juringa," *Matematicheskie Zametki*, **13**(6), 1973 pp. 899-909, (translation: "Solvability of the Halting Problem for Certain Classes of Turing Machines," *Mathematical Notes of the Academy of Sciences of the USSR*, **13**(6), Nov. 1973 pp. 537-541).

[12] M. Kudlek, "Small Deterministic Turing Machines," *Theoretical Computer Science*, **168**(2), 1996 pp. 241-255.

[13] M. Margenstern, "Non Erasing Turing Machines: A Frontier between a Decidable Halting Problem and Universality," in *Lecture Notes in Computer Science: Proceedings of the 9th International Symposium on Fundamentals of Computation Theory (FCT 1993)*, Szeged, Hungary (Zoltán Éstik, ed.), Berlin: Springer-Verlag, 1993 pp. 375-385.

[14] M. Margenstern, "The Laterality Problem for Non-Erasing Turing Machines on {0, 1} Is Completely Solved," *Informatique théorique et Applications/Theoretical Informatics and Applications,* **31**(2), 1997 pp. 159-204.

[15] M. Margenstern, "On Quasi-Unilateral Universal Turing Machines," *Theoretical Computer Science*, **257**(1-2), 2001 pp. 153-166.

https://doi.org/10.25088/ComplexSystems.19.1.29