A Turing machine formalism suitable for the characterisation of determinism and reversibility

Armando B. Matos

LIACC, Universidade do Porto

Rua do Campo Alegre 823, 4150 Porto, Portugal

Abstract

In this note we introduce a new formulation of Turing machines that captures the following duality between determinism and reversibility: reversibility is backward-time determinism while determinism is backward-time reversibility. The transition relation of the machine running backwards as well as the property of left/right tape symmetry are also easily characterised. The main difference between our model and the usual one is that the transition relation δ involves not only the symbol scanned by the head before the transition but also the symbol of the cell to which the head will move. Unlike what happens in the conventional model, the relation δ must be viewed just as a list of possible moves and not a cause-effect relation; this is because the direction in which the head will move depends on the symbol that will be scanned (and, of course, the symbol that will be scanned depends on the direction of the movement).

We also present some results that show how the Kolmogorov complexity of successive instantaneous descriptions changes during deterministic or reversible computations.

Keywords. Turing machines, determinism, reversibility.

1 Introduction and motivation

Real computers are usually deterministic. Reversibility is another property of computations which has been studied in models of computation: reversible cellular automata ([6]), reversible grammars ([5]) and reversible two counter machines ([4]). Reversible algorithms such as parsers may have practical advantages (see for instance [5]) and reversibility may in a not too distant future turn out to be an important physical consideration in the sense that there is necessarily dissipation of energy when a computation is not reversible, see for instance [1] and chapter 8 of [3]). These two properties – determinism and reversibility — are very closely related. We would like to formalise Turing machines so as to capture the symmetry between determinism and reversibility. A related problem which is not studied in this paper is the following: given a Turing machine, define an equivalent one which is deterministic and/or reversible.

Reversibility was not a major concern when Turing machines were first formalised. Moreover machines were implicitly assumed to be deterministic and that is reflected in the fact that δ is usually defined as a (partial) *function*. Nondeterministic machines were later introduced by allowing δ to be a *relation* so that we could express the *possibility* of a transition by saying that the corresponding tuple belongs to δ. The characterisation or construction of reversible Turing machines is easier if we use a generalisation of the transition relation δ where the reversibility or non-reversibility of a machine is immediately apparent. For this purpose Bennet in [1] decomposed each transition (quintuple)[1] into two $(qs \to q's'm)$ different transitions (quadruples): $(qs \to q's')$ (write/change of state) and $(q \to m)$ (move the head). We quote from [1]:

"...because the write and shift operators do not commute, the inverse of a read-write-shift quintuple, though it exists, is of a different type, namely, shift-read-write. In constructing a reversible machine it is necessary to include quintuples of both types, or else to use a formalism in which transitions and their inverses have the same form. Here the later approach is taken – the reversible machine will use a simpler type of transition formula in which, during a given transition, each tape is subject to a read-write or to a shift operation but no tape is subject to both."

In this note we propose a third alternative: all transitions are of the same type (they are hepttuples) and the inverse of a transition is also a transition of the same type. Besides, the above cited dualism between determinism and reversibility is clearly expressed in this new formalism.

1.1 Duality between determinism and reversibility

The following simple but interesting observation should be easily expressible in the new formulation: a Turing machine and more generally any model of computation is reversible iff it is deterministic in the backward time direction and it is deterministic iff it is reversible in the backward time direction:

$$\begin{aligned} \text{Reversibility} &\equiv \text{Backward determinism} \\ \text{Determinism} &\equiv \text{Backward reversibility} \end{aligned}$$

[1] s and s' are the symbols in the cell scanned by the head respectively before and after the transition, q and q' are the machine states respectively before and after the transition and $m \in \{-1, 0, 1\}$ is the shift of the tape head.

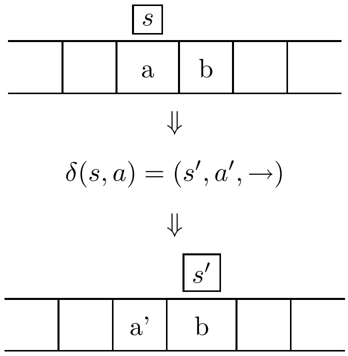

Figure 1: A typical transition

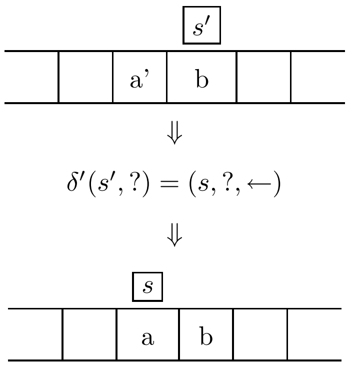

Figure 2: The backward transition

1.2 Motivation

In order to motivate our approach let us consider a typical transition (Figure 1). As we will see, the usual Turing machine model does not easily characterise the machine running backwards nor does it capture the symmetry between the concepts of determinism and reversibility.

We would like to use δ or some modification of it to characterise the corresponding inverse transition, see Figure 2.

We see that the usual characterisation of a Turing machine does not easily represent the behaviour in the reverse time direction. In fact, from the description of a Turing machine M it is not possible to obtain directly the relation δ' that characterises M running backwards. Yet one would expect that δ' should be easy to get from δ. Intuitively, δ' seems to be, in some sense, the inverse of δ — certainly not in the usual sense because only binary relations have an inverse.

It seems that we should include in the transition function (or relation) both the tape symbol scanned by the head before the transition (a in this case) *and* the symbol scanned after the transition (b in this case). The transitions mentioned above would be described as (in functional notation)

$$\begin{aligned} \delta(s,a,b) &= (s',a',b,\rightarrow) \quad \text{(forward transition)} \\ \delta'(s',b,a') &= (s,a,b,\leftarrow) \quad \text{(backward transition)} \end{aligned}$$

where δ' characterises backward transitions. Let us now consider the general situation where δ is a relation defined on the set

$$S^2 \times \Gamma^4 \times \{\leftarrow, \rightarrow\}$$

The fact that for instance $(s,\ s',\ a,\ b,\ a',\ b',\ \rightarrow) \in \delta$ means that a transition is possible (in the forward time direction) if and only if the following constraints are verified.

- Current state is s and the state after the transition will be s'.
- The head will move to the right.
- The head is scanning a cell containing symbol a. This symbol will be changed to a' after the transition.
- The head (after the transition) will be scanning a cell containing symbol b. The symbol b' will be written in this cell.

Notice that in this formalisation there is no clear distinction between cause and effect. The symbol b is in the cell that the head will be scanning *after* the transition. In a sense, the direction of the movement depends on the symbol that will be scanned (and, of course, the symbol that will be scanned depends on the direction of the movement)! Notice also that "traditional" transitions have $b = b'$ and a head movement that does not depend on b.

Now consider the relation δ' which corresponds to transitions in the backward direction. We have

$$(s,s',a,b,a',b',\rightarrow) \in \delta$$

if and only if

$$(s',s,b',a',b,a,\leftarrow) \in \delta'$$

We see that usual transitions are a special case of our model: $b = b'$ (the symbol in the cell that the head will be scanning after the transition does not change) in the usual forward transitions and $a = a'$ (the symbol over the head does not change) in the usual backward transitions (see again Figures 1 and 2). Our formulation models both forward and backward transitions.

Throughout this paper we have adopted the following convention. When we talk about a "Turing machine" without further qualification we are referring to a machine characterised by the new formulation introduced in this paper. "Usual Turing machines" refer to the traditional characterisation (see for instance [2]) where we assume that in every transition the head must move 1 cell to the left or to the right and that the tape is infinite in both directions.

The rest of this paper is organised as follows. In the next Section we describe formally our characterisation and use it to define the inverse machine and study several properties: determinism, reversibility and tape symmetry. In Section 3 the deep relationship between determinism and reversibility is studied in more detail. Section 4 presents two simple examples of Turing machines whose inverses are studied; the reader may find it interesting to read this Section before proceeding. Some results relating the change of the Kolmogorov complexity of the machine configuration during a computation and the properties of determinism and reversibility are presented in Section 5. We end in Section 6 with some conclusions and prospects for future work.

2 The characterisation

Through this paper we use the following mathematical conventions.

- By "function" we mean a total function.
- The notation "$f(n)\downarrow$" means that $f(n)$ is defined for the argument n.
- If A, B,..., C are sets then $A \times B \times \cdots \times C$ denotes the set of all tuples $(a, b, \cdots, c)$ with $a \in A$, $b \in B$,..., $c \in C$.
- A relation defined on sets A, B,... and C is any subset of $A \times B \times \cdots \times C$.
- $(a_1, \cdots, a_n) = (b_1, \cdots, b_n)$ means that $a_1 = b_1$ and... and $a_n = b_n$.

2.1 Basic definitions

The definitions below formalise the concepts introduced in the previous Section.

Definition 1 *A* Turing machine *is characterised by a finite set S of states, a tape alphabet Γ and a relation δ defined on*

$$S^2 \times \Gamma^4 \times \{\leftarrow, \rightarrow\}$$

where $(s, s', a, b, a', b', d) \in \delta$ iff the following transition is possible: (i) the current state is s and the future state is s'; (ii) the current and future symbols of the cell currently scanned by the head are respectively a and a'; (iii) the current and future symbols of the cell scanned by the head after the transition are respectively b and b'; (iv) the direction of the movement of the head is given by d. We define $\rightarrow^{-1} \equiv \leftarrow$ and $\leftarrow^{-1} \equiv \rightarrow$

Perhaps we should emphasise that in Definition 1 the symbol b may be either the symbol at the cell at right (if $d = \rightarrow$) or at left (if $d = \leftarrow$) of the current cell (which contains a).

In this paper we do not deal with machines recognising languages so that we have not included initial and final states in our formalism. We can study a computation starting in any configuration we like, that is, we can specify that the machine begins with a certain tape contents, state, head location and cell it is scanning.

Instantaneous descriptions and computations

An *instantaneous description ID* or *configuration*

$$\overbrace{x_1 \cdots x_i}^{x}\ \overset{\boxed{s}}{a}\ \overbrace{y_1 \cdots y_j}^{y}$$

will be denoted as $x\, a_s\, y$; that is, we use the state s as a subscript of the symbol a currently scanned by the head. As usual, x is finite and, if not empty, its leftmost symbol is non-blank; and similarly for y.

Definition 2 *A* computation *of a Turing machine is a possibly infinite sequence*

$$(\cdots,\, I_i,\, I_{i+1},\, \cdots)$$

where each I_i is an ID and such that, for every i, the transition $I_i \rightarrow I_{i+1}$ is possible (according to the relation δ of M). Moreover if the sequence is not infinite and starts with instantaneous description I_0 (ends with I_f), there is no possible previous (respectively future) IDs (according to δ).

We will also use the following functional notation.

Definition 3 *Suppose that M is a Turing machine characterised by a transition relation δ.*

- *If s is a state and a and b are symbols $\delta(s, a, b)$ is defined as*

$$\delta(s, a, b) = \{(s', a', b', d) \mid (s, s', a, b, a', b', d) \in \delta\}$$

- *If $\mathcal{I}$ is a set of IDs we define $\delta(\mathcal{I})$ as the set of all IDs that may result by a move of M (according to δ) from some member of $\mathcal{I}$.*

The inverse of a Turing machine

Definition 4 *The* inverse *(or* reverse*) of a Turing machine M having a relation δ is a Turing machine M' whose relation δ' is characterised as follows:*

$$(s, s', a, b, a', b', d) \in \delta \quad \text{iff} \quad (s', s, b', a', b, a, d^{-1}) \in \delta'$$

We prefer the designation "inverse" to "reverse" because any Turing machine (reversible or not) has an inverse.

Let us present some properties that show that the relation δ' is in some sense the inverse of relation δ (there is no simple generalisation of the notion of inverse to relations defined in more than 2 sets). From the definition of δ' we have for any instantaneous descriptions I_1 and I_2

$$I_1 \Rightarrow_\delta I_2 \quad \textit{iff} \quad I_2 \Rightarrow_{\delta'} I_1$$

Theorem 1 *The inverse M' of a Turing machine M characterises the backward computations of M (computations in reverse time) in the following sense. A sequence*

$$(I_1, I_2, \cdots, I_n)$$

is part of a computation of M if and only if

$$(I_n, I_{n-1}, \cdots, I_1)$$

is part of a computation of M'.

Theorem 2 *The inverse operator is involutive in the following sense: $M'' = M$ for any Turing machine M.*

Proof. We have

$$(s, s', a, b, a', b', d) \in \delta'' \text{ iff } (s', s, b', a', b, a, d^{-1}) \in \delta' \text{ iff } (s, s', a, b, a', b', d) \in \delta$$

$\diamond$

Theorem 3 is an immediate consequence of the definition of δ and δ'.

Theorem 3 *For every set $\mathcal{I}$ of instantaneous descriptions the following holds.*

- *If for every $I \in \mathcal{I}$ we have $\delta(\{I\}) \neq \emptyset$, then $\mathcal{I} \subseteq \delta'(\delta(\mathcal{I}))$.*
- *If for every $I \in \mathcal{I}$ we have $\delta'(\{I\}) \neq \emptyset$, then $\mathcal{I} \subseteq \delta(\delta'(\mathcal{I}))$.*

Determinism and reversibility

Let us now introduce a natural definition of determinism and reversibility.

Definition 5 *A Turing machine M having a relation δ is*

- deterministic *if for every instantaneous description I, $|\delta(\{I\})| \leq 1$*

– reversible *if for every instantaneous description* I, $|\delta'(\{I\})| \leq 1$

Thus a machine is deterministic (reversible) iff *every* computation is deterministic (reversible). Clearly, a particular computation of a nondeterministic (non-reversible) machine may well be deterministic (reversible).

2.2 The traditional characterisation

The usual characterisation of Turing machines correspond to a special case of our formulation in which the transition does not depend on the contents b of the cell that will be scanned and the contents of that cell does not change.

Definition 6 *An* usual *Turing machine is a Turing machine where the relation* δ *is such that for all states* s *and* s' *and symbols* a*,* b*,* a' *and* b' *we have*

$$(s, s', a, b, a', b', d) \in \delta \quad \Rightarrow \quad b = b'$$

and

$$(s, s', a, b, a', b, d) \in \delta \;\; \textit{iff} \;\; (s, s', a, b', a', b', d) \in \delta$$

Let us now consider the definitions of determinism and reversibility for the usual Turing machines.

Definition 7 *An* usual deterministic *Turing machine is a Turing machine where*

– δ *is a partial function* $\delta: \; S \times \Gamma^2 \to S \times \Gamma^2 \times \{\leftarrow, \rightarrow\}$

– $\forall s \in S$, $\forall a, b_1, b_2 \in \Gamma$ *either* $\delta(s, a, b_1)$ *and* $\delta(s, a, b_2)$ *are both undefined or they are both defined and have the same value. This means that, in the usual characterisation, the symbol that the head will scan after transition does not matter.*

– $\forall s \in S$, $\forall a, b \in \Gamma \;\; \delta(s, a, b) \downarrow \;\Rightarrow\; \exists s' \in S$, $\exists a' \in \Gamma$, $\exists d \in \{\rightarrow, \leftarrow\} \;\; \delta(s, a, b) = (s', a', b, d)$. *This means that the symbol* b *does not change in an usual transition.*

As the transition does not depend on the contents b *of the cell that will be scanned (whose value does not change), that parameter can be ignored and we can write* $\delta(s, a) = (s', a', d)$*.*

Obviously, "our" Turing machines, just like the usual ones, recognise exactly the set of recursively enumerable languages.

Theorem 4 *Deterministic Turing machines have the computation power of usual Turing machines.*

Proof. An *usual* deterministic Turing machine M can easily be transformed into an equivalent deterministic Turing machines M_e. In particular, every transition $\delta(s,a) = (s',a',d)$ of M is translated into several transitions of M_e:

$$\delta(s,a,b) = (s',a',b,d) \quad \text{(one for each symbol } b\text{)}$$

Conversely it is obvious that our deterministic Turing machine can be simulated by a traditional one . ◇

2.3 On tape-symmetric Turing machines

We would like to define the concept of machines whose behaviour is symmetric with respect to the direction of the tape (left/right) and express that fact in terms of our formulation.

Definition 8 *A Turing machine is* tape-symmetric *if, for every initial instantaneous description $x\,a_s\,x^R$ (a symmetric configuration) and integer $n \geq 0$, if $y\,b_{s'}\,z$ is a possible ID after n moves then $z^R\,b_{s'}\,y^R$ is also a possible ID after n moves.*

Theorem 5 *A Turing machine having a transition relation δ is tape-symmetric iff the following condition holds*

$$(s,s',a,b,a',b',d) \in \delta \quad \textit{iff} \quad (s,s',a,b,a',b',d^{-1}) \in \delta$$

Proof. We suppose first that the condition in the statement of this Theorem is verified. Assume that, after n moves, a configuration is possible iff its symmetric is possible; that will be also true after $n+1$ moves. This is because the condition ensures that the symmetric transition is also possible.

Conversely assume that the machine is tape-symmetric and that the condition does not hold, say that $(s,s',a,b,a',b',\rightarrow) \in \delta$ but $(s,s',a,b,a',b',\leftarrow) \notin \delta$. Then, considering an initial symmetric instantaneous description $b\,a_s\,b$ (where we may chose $a = b$ if there is only one tape symbol), the transition $b\,a_s\,b \;\Rightarrow\; b\,a'\,b'_{s'}$ is possible while its mirror image $b\,a_s\,b \;\Rightarrow\; b'_{s'}\,a\,b$ is not possible (contradiction). ◇

The following Theorem shows that machines that are simultaneously tape-symmetric and deterministic are not very interesting...

Theorem 6 *A tape-symmetric Turing machine is deterministic iff it satisfies: $\delta(s,a,b) = \emptyset$ for every state s and symbols a and b.*

Proof. The "if" part is obvious because there are no possible future configurations. To prove the implication on the other direction note that the computation that starts in the configuration $b\,a_s\,b$, is not deterministic. ◇

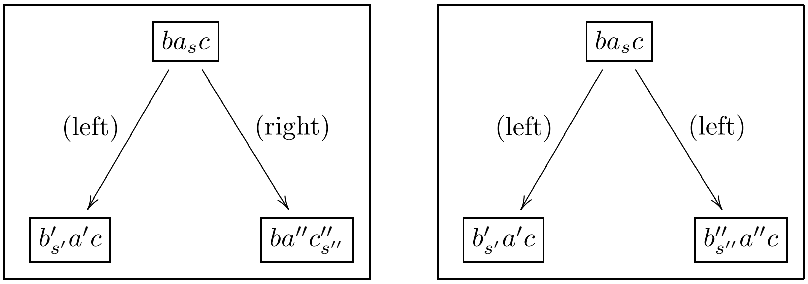

Figure 3: Nondeterministic behaviour: either there are two possible directions of the head movement or two different moves in the same direction (in this case to the left).

3 Determinism and reversibility

In Definition 5 we have defined determinism and reversibility in terms of IDs. We will now characterise these properties in purely syntactic terms, that is, solely in terms of the relation δ. This is not possible (for the reversibility) with the usual Turing machine model. In our formulation reversibility and determinism are dual concepts as the following Theorem shows.

Theorem 7 *A Turing machine M is reversible if and only if its inverse M' is deterministic. A Turing machine M is deterministic if and only if its inverse M' is reversible.*

Proof. Immediate from the definitions ◇

The possible sources of nondeterminism are as follows (see Figure 3): from the current configuration,

– The head may move either to the left or to the right.

– The head may move left in two or more ways.

– The head may move right in two or more ways (not represented in Figure 3).

Theorem 8 gives a characterisation of nondeterminism in terms of δ.

Theorem 8 *A Turing machine is deterministic iff the relation δ satisfies the two following conditions.*

1. It is not possible to move in both directions – *For every state s and symbol a, there are not two or more transitions in different directions, that is, there are no state and symbols (below represented as anonymous by "_") such that*

$$(s, \_, a, \_, \_, \_, \rightarrow) \in \delta$$
$$(s, \_, a, \_, \_, \_, \leftarrow) \in \delta$$

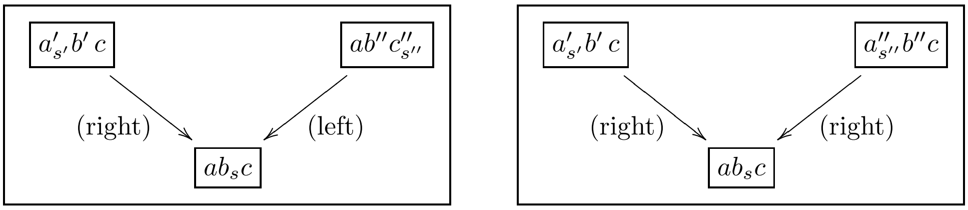

Figure 4: Non-reversibility: either the current ID may result from two transitions in different directions (at left) or two different transitions in the same direction (in this case to the right).

2. There is at most one way to move in one direction – *For every state s, symbol a and direction d, there are no state s' and symbols a_1, a_2, b_1 and b_2 such that $(a_1, b_1) \neq (a_2, b_2)$ and*

$$(s, \_, a, b, a_1, b_1, d) \in \delta$$
$$(s, \_, a, b, a_2, b_2, d) \in \delta$$

Proof. If δ satisfies the conditions of the Theorem, there is at most one possible future ID. Conversely, if the machine is deterministic, the violation of any of the conditions would cause a nondeterministic behaviour if the machine is started in an appropriate configuration. $\diamond$

Let us now consider the characterisation of reversibility. Dually, one has the following.

The sources of non-reversibility are illustrated in Figure 4. The syntactic characterisation of reversibility is given in Theorem 9.

Theorem 9 *A Turing machine is reversible iff the relation δ satisfies the two following conditions.*

1. It is not possible to reach the current configuration from both tape directions — *For every state s' and symbol b', there are not two or more transitions in different directions that cause the machine enter state s' with the head over symbol b', that is, there are no state and symbols (below represented as anonymous by "_") such that*

$$(\_, s', \_, \_, \_, b', \rightarrow) \in \delta$$
$$(\_, s', \_, \_, \_, b', \leftarrow) \in \delta$$

2. It is not possible to reach the current configuration from the same tape direction in two or more ways - *For every state s', symbol b' and direction d, there are no states s_1 and s_2 and symbols a_1, a_2, b_1 and b_2 such that $(a_1, b_1) \neq (a_2, b_2)$ and*

$$(s, s', a_1, b_1, a', b', d) \in \delta$$
$$(s, s', a_2, b_2, a', b', d) \in \delta$$

4 Exploring the new characterisation

In this Section we will apply our formulation to two simple Turing machines. In both cases the tape alphabet is $\{0, 1\}$. The first machine is deterministic but not reversible. The second is reversible but not deterministic. For each machine we will build the inverse of the machine and study its properties. Notice that our formulation of Turing machines allows for a much simpler characterisation of the inverse machine than the usual one.

It may help to view the head of the machine as a particle moving (in discrete steps) in an unidimensional space (the tape). Then we can describe both machines as collision with a layer of n 1's. For simplicity we assume that

– Initially the tape contains a sequence of $n \geq 1$ ones, all the other cells being blank (0):

$$\cdots 0\,0\,0\,0\,1\,1\,1 \cdots 1\,0\,0\,0\,0 \cdots$$

– The head is located at left of the string 1^n.

– The machine is in state s_0.

4.1 A thickening reflection

This machine is described in Figure 5. The particle (the head) is reflected by the right extreme of the layer while incrementing its thickness by one; if we think of the layer 1^n as a representation of the number n, this machine corresponds to the successor function.

Let us check that the machine is deterministic. For $s = s_0$ and $a = 0$ (see Figure 5) the only possible direction is "$\rightarrow$". Similarly, for $s = s_0$ and $a = 1$ the only possible direction is also "$\rightarrow$". All transitions from s_1 are to the left. Condition 2 of Theorem 8 is also verified: for each (s, a, b, d) there is only one (s', a', b') such that $(s, s', a, b, a', b', d) \in \delta$.

The inverse of the machine can be obtained by using Definition 4 and is described in Figure 6. A look at this table shows that this machine is not deterministic, so that the original one is not reversible.

It describes a particle coming from the left, with the following possible behaviour: the thickness of the layer is decremented by 1 and the "particle" is reflected. That corresponds to the predecessor function. The reader can investigate other possible behaviours and try to find an equivalent Turing machine that is both deterministic and reversible.

No.	s	a	b	s'	a'	b'	d	Comment
1	s_0	0	0	s_0	0	0	$\rightarrow$	(go right)
2	s_0	0	1	s_0	0	1	$\rightarrow$	(go right)
3	s_0	1	1	s_0	1	1	$\rightarrow$	(go right)
4	s_0	1	0	s_1	1	1	$\rightarrow$	(write 1, goto s_1)
5	s_1	0	0	s_1	0	0	$\leftarrow$	(go left)
6	s_1	0	1	s_1	0	1	$\leftarrow$	(go left)
7	s_1	1	0	s_1	1	0	$\leftarrow$	(go left)
8	s_1	1	1	s_1	1	1	$\leftarrow$	(go left)

Figure 5: Machine A: increment 1 and reflect. Transition number 4 is responsible for the reflection and writing of a 1. Transitions 1-3 correspond to right and transitions 5-8 to left head movement.

No.	s	a	b	s'	a'	b'	d	Comment
1	s_0	0	0	s_0	0	0	$\leftarrow$	(go left)
2	s_0	1	0	s_0	1	0	$\leftarrow$	(go left)
3	s_0	1	1	s_0	1	1	$\leftarrow$	(go left)
4	s_1	1	1	s_0	0	1	$\leftarrow$	(erase 1, goto s_0)
5	s_1	0	0	s_1	0	0	$\rightarrow$	(go right)
6	s_1	1	0	s_1	1	0	$\rightarrow$	(go right)
7	s_1	0	1	s_1	0	1	$\rightarrow$	(go right)
8	s_1	1	1	s_1	1	1	$\rightarrow$	(go right)

Figure 6: The inverse of machine A: a possible behaviour (the machine is nondeterministic) is "decrement 1 and reflect". Transition number 4 is responsible for the reflection and writing of a 1. The inverse was obtained with the rule: $(s, s', a, b, a', b', d) \in \delta$ iff $(s, s', b', a', b, a, d^{-1}) \in \delta'$. Now, s_1 is the initial state and transition number 5 is the first one that can be applied (when the head comes from the 0's at left)

.

No.	s	a	b	s'	a'	b'	d	Comment
1	s_0	0	0	s_0	0	0	$\rightarrow$	(go right)
2	s_0	1	0	s_0	1	0	$\rightarrow$	(go right)
3	s_0	1	1	s_0	1	1	$\rightarrow$	(not used)
4	s_0	0	1	s_1	0	1	$\rightarrow$	(start of reflection)
5	s_0	0	1	s_2	0	1	$\rightarrow$	(go through)
6	s_1	1	0	s_1	1	0	$\leftarrow$	(go left)
7	s_1	0	0	s_1	0	0	$\leftarrow$	(go left)
8	s_2	1	1	s_2	1	1	$\rightarrow$	(go right)
9	s_2	1	0	s_2	1	0	$\rightarrow$	(go right)

Schematically:

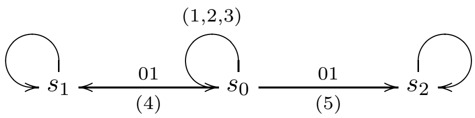

Figure 7: Machine B: in state s_0, with first pair "01" either (nondeterministically) use transition number 4 and reflect (state s_1) or transition number 5 and continue to the right (state s_2).

4.2 Come back or go through?

The second machine (Figure 7), being nondeterministic, is similar to a quantum process with two possible outcomes: either the particle (the head) is reflected by the first 1 or it goes through it to the right behaving in the same way (nondeterministically on 0-1 transitions) after that.

We have computed the inverse of the machine which is represented in Figure 8. It represents a situation where the particle may come either from the right or the left, going to the left in both cases. It is clearly deterministic but not reversible.

5 On the Kolmogorov complexity of instantaneous descriptions

Consider a particular, possibly nondeterministic, Turing machine M. Suppose that, after $n \geq 0$ moves, the instantaneous description I_1 gives rise to instantaneous description I_2.

$$I_1 \Rightarrow^\star I_2$$

Let us see how the Kolmogorov complexities of the two instantaneous descriptions are related.

Suppose for instance that M is deterministic. To describe I_2 it is enough to know I_1, the

No.	s	a	b	s'	a'	b'	d	Comment
1	s_0	0	0	s_0	0	0	$\leftarrow$	(go left)
2	s_0	0	1	s_0	0	1	$\leftarrow$	(go left)
3	s_0	1	1	s_0	1	1	$\leftarrow$	(not used)
4	s_1	1	0	s_0	1	0	$\leftarrow$	(start of reflection)
5	s_2	1	0	s_0	1	0	$\leftarrow$	(go through)
6	s_1	0	1	s_1	0	1	$\rightarrow$	(go right)
7	s_1	0	0	s_1	0	0	$\rightarrow$	(go right)
8	s_2	1	1	s_2	1	1	$\leftarrow$	(go left)
9	s_2	0	1	s_2	0	1	$\leftarrow$	(go left)

Schematically:

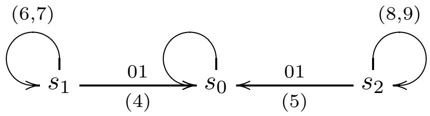

Figure 8: Inverse of machine B: transition number 4 corresponds to a "particle" coming from the left (state s_1) and reflecting to the left (state s_0). Transition number 5 corresponds to a "particle" coming from the right (state s_2) that continues to the left.

description of M and the number n of moves. So, using auto-delimited descriptions,

$$K(I_2) \leq K(I_1) + |d(M)| + |n|$$

Let us adopt the notation $a \leq_{f(n)} b$ to denote that $a \leq b + O(f(n))$ where a and b may depend on n. Similarly we may write $a \geq_{f(n)} b$ to denote that $b \leq_{f(n)} a$ and $a =_{f(n)} b$ to denote that $a \leq_{f(n)} b$ and $b \leq_{f(n)} a$. With this notation we have for the deterministic machine M considered above

$$K(I_2) \leq_{\log n} K(I_1)$$

The other two cases represented in Figure 9 can be analysed similarly and we can summarise the results as follows.

– M is deterministic: $K(I_2) \leq_{\log n} K(I_1)$. The complexity never increases. This corresponds to the usual situation of a (deterministic) computer programs without input.

– M is reversible: $K(I_1) \leq_{\log n} K(I_2)$. The complexity never decreases.

– M is deterministic and reversible: $K(I_1) =_{\log n} K(I_2)$. The complexity does not change.

I would like to emphasise the close relationship between nondeterminism and the growth of the Kolmogorov complexity. Consider for instance the nondeterministic phase of a guessing Turing

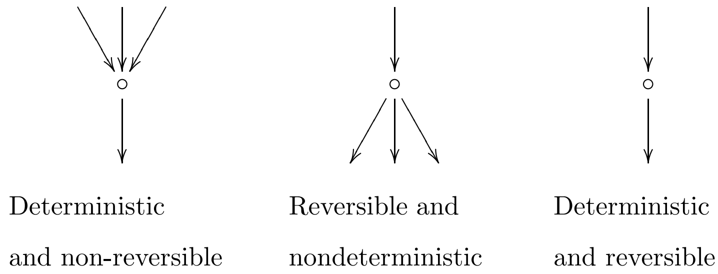

Figure 9: Determinism and reversibility

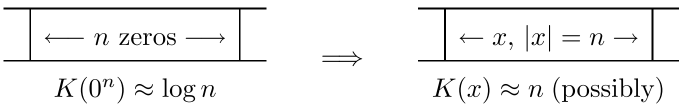

Figure 10: The complexity of Kolmogorov may increase during nondeterministic computations

machine where a string of 0's and 1's is written on an initially blank (all 0's) tape. The Kolmogorov complexity will almost certainly increase as shown in Figure 10.

6 Conclusions and future work

We have described a new formulation of Turing machines in which we can define the properties of determinism, reversibility, tape-symmetry and characterise the inverse (or backward time) machine solely in terms of the transition relation δ; see definition 4 and theorems 8, 9 and 5. The new formulation can possibly be applied to other areas such as the design of deterministic and/or reversible algorithms for Turing machine.

Let us also refer two more specific areas, where further research, possibly using the formulation presented in this paper, could produce interesting results. One is the characterisation of languages (or the computation of functions) in the terms described in Section 4 where we viewed computations as a motion of a particle (the head of the machine) that may change the state. Other is the the study of the relationship between the following three aspects of a computation: (i) the properties of determinism and reversibility, (ii) the change of the Kolmogorov complexity of the IDs and (iii) the physical properties of dissipation or absorption of energy.

References

[1] Charles H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 6:525–532, 1973.

[2] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison Wesley, 1979.

[3] Ming Li and Paul M. B. Vitàny. *An Introduction to Kolmogorov Complexity and its Application.* Texts and Monographs in Computer Science. Springer-Verlag, 1994.

[4] Kenichi Morita. Universality of a reversible two-counter machine. *Theoretical Computer Science*, 168:303–320, 1996.

[5] Tomek Strzalkowski, editor. *Reversible Grammars in Natural Language Processing.* Kluwer Academic Publishers, 1994.

[6] T. Toffoli and Norman Margolus. Invertible cellular automata: a review. *Physica*, D 45:229–253, 1990.