Entropy and Speed of Turing machines

E. Jeandel

LORIA (Nancy, France)

Context

Turing machines with one head and *one tape*.

- States Q.
- Symbols Σ.
- Transition map: $Q \times \Sigma \to Q \times \Sigma \times \{-1, 1\}$

Turing machines as a dynamical system: $M : Q \times \Sigma^{\mathbb{Z}} \to Q \times \Sigma^{\mathbb{Z}}$
(the tape moves, not the head)

- No specified initial state (very important)
- No specified initial configuration (crucial)
- Might have final states (anecdotal)

TM as a DS

Seeing Turing machines as a dynamical system changes a lot of things:

- Interested in the behaviour starting from *all* configurations, not only *one* configuration.
- Hard to conceive of a TM with no (temporally) periodic configurations.
- Nevertheless, intricate TMs do exist.

TM as a DS

Theorem (essentially Turing 1937)

There is no algorithm to decide whether a TM does not halt on its input configuration.

Theorem (Hooper 1966)

There is no algorithm to decide whether a TM does not halt on some input configuration.

simplified proof by Kari-Ollinger (2008), which leads to the undecidability of the existence of a periodic point.

Dynamical Systems

Part of a recent trend which sees computational models as dynamical systems.
Good alternative to the classical Robinson technique for tilings:

- Turing machines (as a Dyn. Sys.) can be easily encoded into piecewise affine maps.
- Piecewise affine maps can be easily encoded into tilings

This talk

We will show why some thing are actually computable for 1-tape Turing machines, namely:

- its speed
- its entropy

Speed

For c a configuration, let $S_n(c)$ be the set of (different) cells visited during the first n steps of the computation on input c, and $s_n(c) = \#S_n(c)$

$s_n(c)$ is (Kingman)-subadditive

$$s_{n+m}(c) \leq s_n(c) + s_m(M^n(c))$$

If $d(x, y) \leq 2^{-s_n(x)}$ then $d(M^n(x), M^n(y)) \leq 1/2$.

Speed

$$\overline{s}(c) = \limsup \frac{s_n(c)}{n} \quad \underline{s}(c) = \liminf \frac{s_n(c)}{n}$$

If $\liminf = \limsup$, we denote by $s(c)$ the *speed* of c.

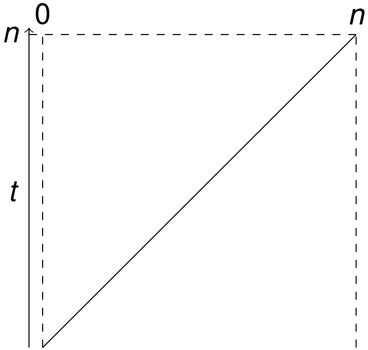
Some example(s)

Consider a Turing machine that stays in the same direction when reading a symbol a, and changes direction when reading a b (changing it into an a)

Some example(s)

Consider a Turing machine that stays in the same direction when reading a symbol *a*, and changes direction when reading a *b* (changing it into an *a*)

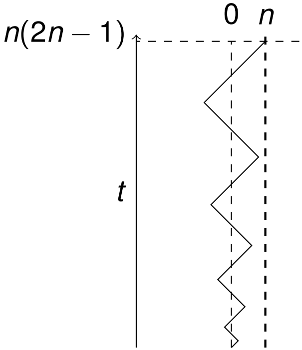
If c contains only *a*'s, $s(c) = 1$.



Some example(s)

Consider a Turing machine that stays in the same direction when reading a symbol a, and changes direction when reading a b (changing it into an a)

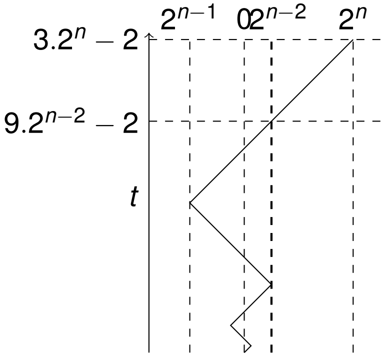
If c contains only b's, $s(c) = 0$.



Some example(s)

Consider a Turing machine that stays in the same direction when reading a symbol *a*, and changes direction when reading a *b* (changing it into an *a*)

If c contains b at positions $(-2)^i$
$\underline{s}(c) = 1/3, \overline{s}(c) = 1/2$



The speed

Definition

$$S(M) = \max_{c \in \mathcal{C}} \underline{s}(c) = \max_{c \in \mathcal{C}} \overline{s}(c) = \lim_n \sup_c \frac{s_n(c)}{n} = \inf_n \sup_c \frac{s_n(c)}{n}$$

All definitions are indeed equivalent. This is due to compactness of the set of configurations and subadditivity.
Note that it is a maximum, not a supremum.

Entropy

Here is an equivalent definition, from Oprocha(2006).

For c a configuration, let $T(c)$ be the *trace* of the configuration, i.e. the sequence (states, symbols) visited by the machine. Let $\mathcal{T}$ be the set of all traces

Definition (Oprocha (2006))

$$H(M) = H(\mathcal{T}) = \lim \frac{1}{n} \log |T_n|$$

where T_n are all possible words of length n of the trace

Note: The machine in the example has zero entropy (any word of T_n has “few” symbols b)

In this talk

Theorem

Entropy and speed are computable for one-tape Turing machines.
That is, there is an algorithm, that given every ϵ, can compute an approximation upto ϵ.
Furthermore, the speed is always a rational number

Plan of the talk

- Link between entropy and speed
- Some technical lemmas
- Graphs

Comments

- Surprising, usually every dynamical quantity is semi-computable but not computable
- The speed is not computable as a rational number.
  - Starting from M, we can effectively produce a TM M' for which $S(M') \sim 2^{-t}$ where t is the number of steps before M halts on empty input.
- There is no algorithm to decide if the entropy is zero.
- None of the techniques work with multi-tape TM. The entropy is not computable anymore.

Plan

Entropy = Complexity

- Kolmogorov complexity $K(x)$ of a word x is the size of the smallest program that outputs x
- The (average) complexity of a infinite word u is

$$\overline{K}(u) = \limsup \frac{K(u_{1\dots n})}{n}$$

(same with $\underline{K}(u)$)

Theorem (Brudno 1983, see also Simpson 2013)

For a subshift $\mathcal{T}$,

$$h(\mathcal{T}) = \max_{u \in \mathcal{T}} \overline{K}(u) = \max_{u \in \mathcal{T}} \underline{K}(u)$$

(More exactly, the maximum is reached μ-a.e, for μ ergodic of maximal entropy)

Consequences

Proofs for entropy and speed are relatively the same.
We will deal with speed in the talk.

Plan

1. Entropy vs Speed
2. Main idea
3. Core of the proof

The goal

$$S(M) = \max_{c \in \mathcal{C}} s(c) = \inf_n \sup_c \frac{s_n(c)}{n}$$

$S(M)$ (and $H(M)$) is computable from above due to the last definition.
We need to prove it is computable from below.
We need lower bounds on the speed and the entropy.

Main idea

$$T(c) = (q_1, a)(q_2, b)(q_1, c)(q_1, a)(q_3, a)(q_1, c)(q_3, c)(q_1, a)(q_2, c)(q_3, b)\dots$$

Main idea

$$\begin{array}{lcccccccccc} & 0 & 1 & 2 & 1 & 2 & 3 & 2 & 1 & 2 & 3 \\ T(c) = & (q_1, a) & (q_2, b) & (q_1, c) & (q_1, a) & (q_3, a) & (q_1, c) & (q_3, c) & (q_1, a) & (q_2, c) & (q_3, b) \dots \end{array}$$

Main idea

$$
\begin{array}{ccccccccccc}
 & 0 & 1 & 2 & 1 & 2 & 3 & 2 & 1 & 2 & 3 \\
T(c) = & (q_1, a) & (q_2, b) & (q_1, c) & (q_1, a) & (q_3, a) & (q_1, c) & (q_3, c) & (q_1, a) & (q_2, c) & (q_3, b)\ldots \\
T(c) = & (q_1, a) & (q_2, b) & (q_1, c) & (q_1, \circ) & (q_3, \circ) & (q_1, c) & (q_3, \circ) & (q_1, \circ) & (q_2, \circ) & (q_3, \circ)\ldots
\end{array}
$$

Deleted information can be recovered (no loss in Kolmogorov complexity)

Main idea

$$
\begin{array}{rcccccccccc}
 & 0 & 1 & 2 & 1 & 2 & 3 & 2 & 1 & 2 & 3 \\
T(c) = & (q_1,a) & (q_2,b) & (q_1,c) & (q_1,a) & (q_3,a) & (q_1,c) & (q_3,c) & (q_1,a) & (q_2,c) & (q_3,b)\ldots \\
T(c) = & (q_1,a) & (q_2,b) & (q_1,c) & (q_1,\circ) & (q_3,\circ) & (q_1,c) & (q_3,\circ) & (q_1,\circ) & (q_2,\circ) & (q_3,\circ)\ldots \\
 & 0\to 1 & 1\to 2 & 2\to 1 & 1\to 2 & 2\to 3 & 3\to 2 & 2\to 1 & 1\to 2 & 2\to 3 & 3\to 4\ldots
\end{array}
$$

Main idea

$$
\begin{array}{lcccccccccc}
 & 0 & 1 & 2 & 1 & 2 & 3 & 2 & 1 & 2 & 3 \\
T(c) = & (q_1, a) & (q_2, b) & (q_1, c) & (q_1, a) & (q_3, a) & (q_1, c) & (q_3, c) & (q_1, a) & (q_2, c) & (q_3, b) \ldots \\
T(c) = & (q_1, a) & (q_2, b) & (q_1, c) & (q_1, \circ) & (q_3, \circ) & (q_1, c) & (q_3, \circ) & (q_1, \circ) & (q_2, \circ) & (q_3, \circ) \ldots \\
 & 0 \to 1 & 1 \to 2 & 2 \to 1 & 1 \to 2 & 2 \to 3 & 3 \to 2 & 2 \to 1 & 1 \to 2 & 2 \to 3 & 3 \to 4 \ldots
\end{array}
$$

$$T'(c) = a q_1 b q_2 q_1 q_2 q_3 q_1 c q_3 q_1 q_2 c q_3$$

Main idea

$$\begin{array}{lcccccccccc} & 0 & 1 & 2 & 1 & 2 & 3 & 2 & 1 & 2 & 3 \\ T(c) = & (q_1,a) & (q_2,b) & (q_1,c) & (q_1,a) & (q_3,a) & (q_1,c) & (q_3,c) & (q_1,a) & (q_2,c) & (q_3,b)\ldots \\ T(c) = & (q_1,a) & (q_2,b) & (q_1,c) & (q_1,\circ) & (q_3,\circ) & (q_1,c) & (q_3,\circ) & (q_1,\circ) & (q_2,\circ) & (q_3,\circ)\ldots \\ & 0\to 1 & 1\to 2 & 2\to 1 & 1\to 2 & 2\to 3 & 3\to 2 & 2\to 1 & 1\to 2 & 2\to 3 & 3\to 4\ldots \end{array}$$

$$T'(c) = aq_1 bq_2 q_1 q_2 q_3 q_1 cq_3 q_1 q_2 cq_3$$

$$T'(c) = \boxed{aq_1}\boxed{bq_2}q_1 q_2 q_3 q_1 \boxed{cq_3}q_1 q_2 \boxed{cq_3}$$

In the rest of the talk, states will be colored

Main idea

- $T'(c)$ is well defined when c matters.
- The speed on c is the average number of boxed symbols.
- The complexity of c is the average of the complexity of $T'(c)$.
- The speed and the complexity are easier to compute using T'.

Lemma 1

If c is of maximum speed/entropy, then M will visit each cell finitely many times.

If the TM zigzags on input c, then it is losing time.

Corollary

$T'(c)$ is well defined.

Lemma 2

Let c of maximum speed/entropy.
Let f_n be the first time we visit cell n, and l_n the last time we visit cell n
Then $f_n \sim l_n$

Corollary

The speed on c is the average number of boxed symbols.

The position p_n where the n-th boxed symbol appear satisfy
$f_n \leq p_n \leq l_n$

Plan

Now we explain why this T' helps.

The subshift

Let S be the subshift generated by all $T'(c)$.

- Points in S that are not of the form $T'(c)$ have smaller speed/entropy.
- S can be described explicitely.

Formal definition

We define L and R inductively

$$(cR\epsilon, \epsilon, a) \in L$$

If by reading a from state q, we write b, go right in state q'

$$(qw, q'w', a) \in L \iff (w, w', b) \in R$$

If by reading a from state q, we write b, go left in state q'

$$(qq'w, w', a) \in L \iff (w, w', b) \in L$$

(Similar definition for R).
Now S is the set of all words where all factors of the form $awbw'c$ satisfy $(w, b, w') \in L$

Cut and Paste lemma

States are synchronizing (magic) words.
If $xawc$ and $dwby$ are valid, then $xawby$ is valid.

In some way, S can be seen as the set of paths over an infinite graph (where states represent vertices).

Key lemma

For a word c, denote by $s(c)$ its average number of boxed symbol, and $K(c)$ its average complexity.
Let S_n be the subshift of S that forbids more than n consecutive states.
Then

$$S(M) = \sup_{c \in S} s(c) = \sup_n \sup_{c \in S_n} s(c)$$

$$H(M) = \sup_{c \in S} K(c) = \sup_n \sup_{c \in S_n} K(c)$$

Proof

Let c that achieves the maximum $s(c) = \alpha > 0$ and n big enough

- c may contain more than n consecutive states, but this should not happen so often
- Use the cut and paste property to replace these parts by some with a smaller number of consecutive states

If done properly, this will not decrease the speed, and only slightly decrease the complexity.

Proof

$$S(M) = \sup_{c \in S} s(c) = \sup_n \sup_{c \in S_n} s(c)$$

$$H(M) = \sup_{c \in S} K(c) = \sup_n \sup_{c \in S_n} K(c) = \sup_n H(S_n)$$

S_n is a computable sequence of subshifts of finite type, so we can compute an increasing sequence of reals that converges to $H(M)$.
We can say better for the speed

The speed

The speed is a rational number, and is achieved for some S_n by a periodic configuration

In each S_n, the maximum number of boxed symbols is achieved for a periodic configuration c_n. Let W be the set of states that appear in c_n.

- Each $w \in W$ appears only once in the period of c_n.
- If $|W|$ is too big, there will be many big words in W, so the speed will be too small.
- Hence $|W|$ contains at most b words for some b.
- If one of them is bigger than c, then the speed is at most $\frac{b}{c-b}$ hence c is also bounded.

Open problems

Characterize entropies of one-tape Turing machines.

The numbers are computable, and it cannot be all computable numbers.

Find how to compute the average speed.

Find a Turing machine with two tapes for which the entropy (resp. speed) is not a computable number.