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.

TM as a DS

Theorem (essentially the definition)

For every Π_1^0 class S, there exists a TM for which the set S' of inputs (starting from the initial state) on which the TM halts is Medvedev equivalent to S.

Theorem (Jeandel 2012)

For every Π_1^0 class S, there exists a TM for which the set S' of inputs on which the TM halts is Muchnik equivalent to S.

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

The previous result about Muchnik equivalence can be transcoded into a result about tilings, which would be slightly weaker than Simpson 2013 (which have a Medvedev equivalence).

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.

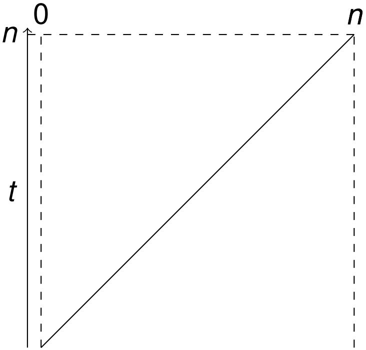
Some example(s)

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)

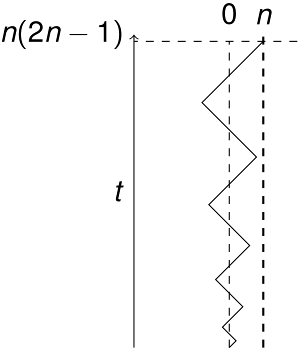
If c contains only a's, $s(c) = 1$.



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*)

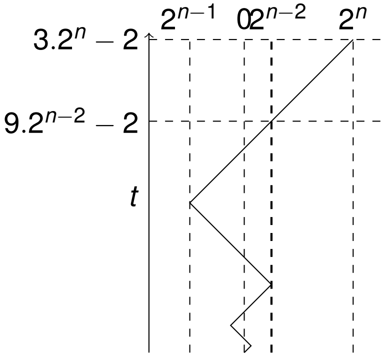
If c contains only b's, $s(c) = 0$.



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.

A few notes about the speed

- The maximal speed is "usually" not reached by random configurations
- Nevertheless, $S(M) = \int s \, d\mu$ for some invariant measure μ.
- $s(c) = \int s \, d\mu$ for μ-random points if μ ergodic

(generalization of Birkhoff theorem to Kingman subadditive functions obtained by combining V'yugin + Hochman (2009))

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

- 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)

Note: equivalence between the two max also follows by subadditivity:
$K(u_1 \dots u_{n+m}) \leq K(u_1 \dots u_n) + K(u_{n+1} \dots u_{n+m}) + O(1)$

Consequences

$$H(M) = \max_{c \in \mathcal{C}} \overline{K}(T(c))$$

(Similar to the formula for the speed)

Consequences

$T(c)_{1\dots n}$ can be computed if we know the $s_n(c)$ symbols read, the initial position of the head, and the initial state.

$$K(T(c)_{1\dots n}) = K(c_{|S_n(c)}) + O(\log s_n(c)) + O(\log n)$$

$$K(T(c)_{1\dots n}) \leq s_n(c)|\log \Sigma| + O(\log n)$$

$$H(M) \leq S(M) \log |\Sigma|$$

$$S(M) \geq \frac{H(M)}{\log |\Sigma|}$$

(Topological) Pressure

Let M^A be the same as the machine M, but over the alphabet $\Sigma \times A$, that ignores the alphabet A.

$$S(M) = \lim_{|A| \to \infty} \frac{H(M^A)}{\log |\Sigma \times A|}$$

The speed is the entropy for a very large alphabet relative to its size.

If we denote $P_s(x) = H(M^A)$ for $x = \log |A|$, $P_s(x)$ is called the *topological pressure* of $(s_n)_{n \in \mathbb{N}}$.
This result has been proven in this context in Feng-Hang,2010.

Consequences

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

Plan

1. Entropy vs Speed
2. Technical lemmas
3. Core of the proof

The goal

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

$S(M)$ is computable from above due to the last definition. We need to prove it is computable from below.

What is the behaviour of a configuration of maximal speed ?

Lemma 1

Starting from c (of maximal speed) M will visit each cell finitely many times.

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

Corollary

The maximal speed is obtained for a configuration that never goes back to the cell at 0.

The maximal speed is obtained (wlog) for a configuration that visits only cells with nonnegative coordinates.

Lemma 2

Let $f_n(c)$ be the first time we visit cell n, and $l_n(c)$ the last time we visit cell n:

$$S(M) = \max_c \lim \frac{n}{f_n(c)} = \max_c \lim \frac{n}{l_n(c)}$$

$1/S(M)$ is somehow the "average running time".

Plan

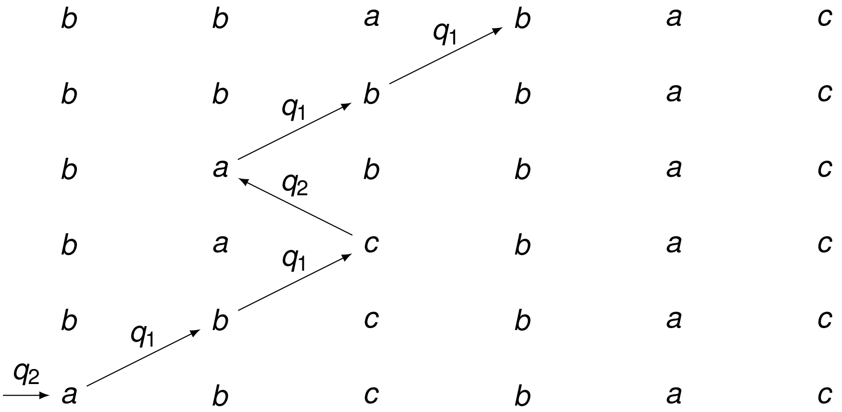
1 Entropy vs Speed

2 Technical lemmas

3 Core of the proof

The proof

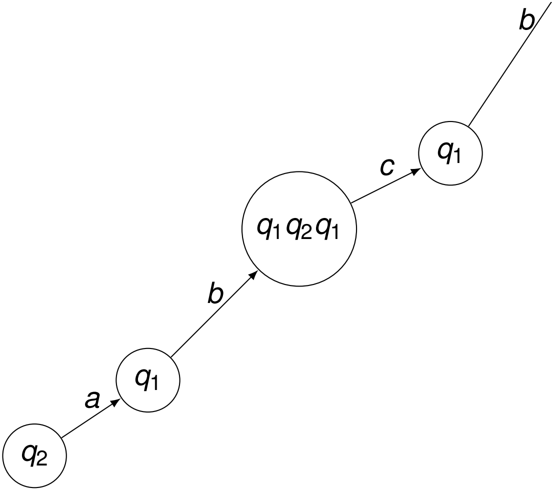
Let c be of maximal speed



Both vertices labeled q_1 represent the same vertex.

The proof

Let c be of maximal speed



Both vertices labeled q_1 represent the same vertex.

Graph

Transform the execution into a graph

- Vertices are all possible (finite) sequence of states
- There is an edge from w to w' labeled by a if it seems possible to see w followed by w' around a cell labeled by a

This accurately represents the behaviour of the TM, as the only transfer of information between cells $]-\infty, m]$ and cells $[m, +\infty[$ occurs at cell m.

Formal definition

We define L and R inductively

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

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).
This graph is computable (for all reasonable definitions of computable)

Key Lemma

To every configuration c corresponds a path in the graph

To every path $(w_1, x_1, w_2, x_2, w_3, x_3 \dots)$ in this graph corresponds a configuration c. For this configuration, $C_n(c)$ is a prefix of w_n.

Speed on the graph

Let $|w|$ (the length of w) be the *weight* of vertex w.
To each path $(w_1, x_1, w_2, x_2, w_3, x_3 \dots)$ we can define its average speed:

$$S(p) = \limsup_n \frac{n}{\sum_{i<n} |w_i|}$$

and its average complexity

$$K(p) = \limsup_n \frac{K(x_1 \dots x_n)}{\sum_{i<n} |w_i|}$$

Reformulation

$$S(M) = \max_p S(p)$$

$$H(M) = \max_p K(p)$$

The maximum is over paths that start from a vertex of weight 1.
Preuve: For any configuration c,
$f_n(c) \leq |C_1(c)| + |C_2(c)| + |C_{n-1}(c)| \leq I_n(c)$.

Key Theorem

Let G_k be the subgraph of vertices with weight at most k

$$S(M) = \sup_k \max_{p \subset G_k} S(p)$$

$$H(M) = \sup_k \max_{p \subset G_k} K(p)$$

Fake proof (1/2)

Suppose the speed/entropy is $\alpha \neq 0$.

- Take a path p that uses possibly vertices of weight $> k$.
- There should be few vertices of big weight (a proportion at most $1/(\alpha k)$)
- Uses alternate paths of length $\leq \beta(k)$ to bypass these vertices.
- Let p' be the new path

The new path is obtained from the previous one by deleting at most a proportion $1/\alpha k$ of vertices, and adding at most a proportion $\beta(k)/\alpha k$ of vertices.
(If done correctly, we can take $\beta(k) = o(k)$)

Fake proof (2/2)

We must show the Kolmogorov complexity does not decrease too much.
The average Kolmogorov complexity of p relative to p' is

$$E(\beta(k)/\alpha k) + E(1/\alpha k) + \log \Sigma/\alpha k$$

where $E(q) = -q \log q - (1-q)\log(1-q)$.

- Specify the vertices that disappear
- Specify where to insert
- Specify what to insert

This converges to 0 when k goes to infinity.

Speed

S(M) is computable

For a finite graph, the maximal speed is obtained by the cycle of minimal weight.
As weights are integers, this implies that the maximal speed is obtained in the infinite graph by the cycle of minimal weight.

Corollaire

$S(M)$ is a rational number. It is achieved by a periodic configuration.

Entropy

How to compute $\max_{p \subset G} K(p)$ for a finite graph G ?

Not clear in the general case. In our case, the graph has no diamond.
For a path $(w_1, x_1, w_2, x_2 \dots)$

$$K(w_1 x_1 w_2 x_2 \dots w_n x_n) = K(x_1 \dots x_n) + O(1)$$

This implies we can unfold the graph, to have only vertices of weight 1.
If all vertices have the same weight, then the maximum complexity is the same as the entropy of the graph (SFT) by Entropy=Complexity.

The entropy is computable

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.