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)

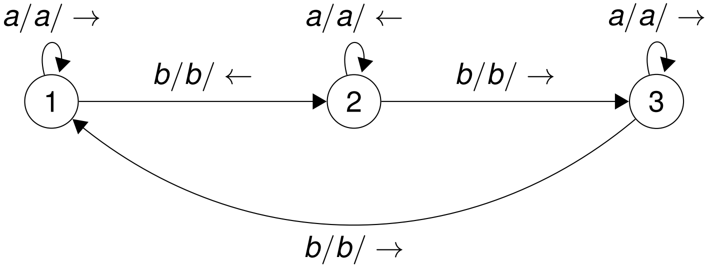
- No specified initial state (very important)
- No specified initial configuration (crucial)
- Might have final states (anecdotal)

TM as a DS

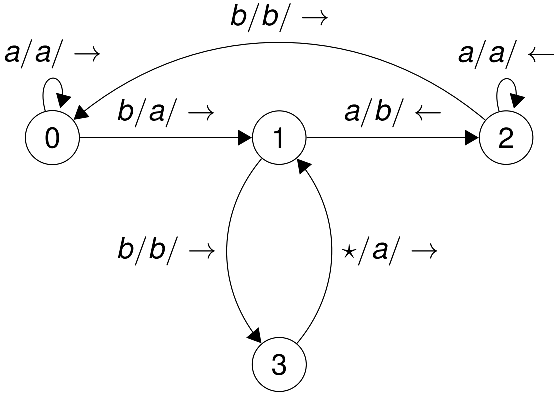
Seeing Turing machines as a dynamical system changes a lot of things:

- 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.

Interesting examples



Interesting examples



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.

The examples

In the first example

- each symbol is read at most three times, so $\overline{s}(c) \geq 1/3$ for all c.
- There exists configurations for which $s(c) = 1$

In the second example

- Some parts of the configuration may be read n times. There are configurations of arbitrary small speed (but no configurations of zero speed)
- There exists configurations for which $s(c) = 1$

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

The examples

The first example:

$$\mathcal{T} \sim \{a^n b^n c^n | n \in \mathbb{N}\}^\omega$$

Gives an entropy of $\log \sqrt[3]{2}$.

Second example:

$$\mathcal{T} \sim \{a^{n^2-3} b | n \geq 2\}^\omega$$

Gives an entropy of $-\log x$ where $x = 0.820863$ is solution of $\theta_3(0, x) = 1 + 2x + 2x^2$ (x is a transcendental number).

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

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

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

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

$$h(\mathcal{T}) = \sup_c \limsup \frac{K(p_n)}{n}$$

Where p_n are the (new) letters read during the first n steps.

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

Plan

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

$$\mathcal{T} \sim \{a^n b^n c^n | n \geq 1\}^\omega$$

$$\mathcal{T} \sim \{(ABC)(abc)^{n-1} | n \geq 0\}^\omega$$

The entropy is easily computable for the second language:
$\mathcal{T} \sim \{ABC, abc\}^\star$

- What is this transformation from the p.o.v. of the TM ?
- For any configuration, let $T'(c)$ be the word that gives, for each cell i, the symbol in position i, then the set of states going from cell i to cell $i+1$.

Main idea

- $T'(c)$ is well defined when c matters.
- Speed and complexity can be read from T' instead of T.
- T' is easy to understand.

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

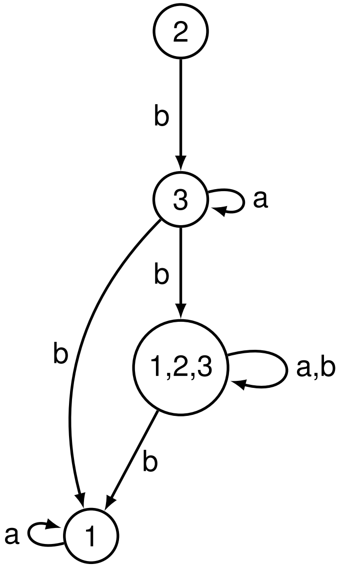
The speed on c is the average number of letters.

Plan

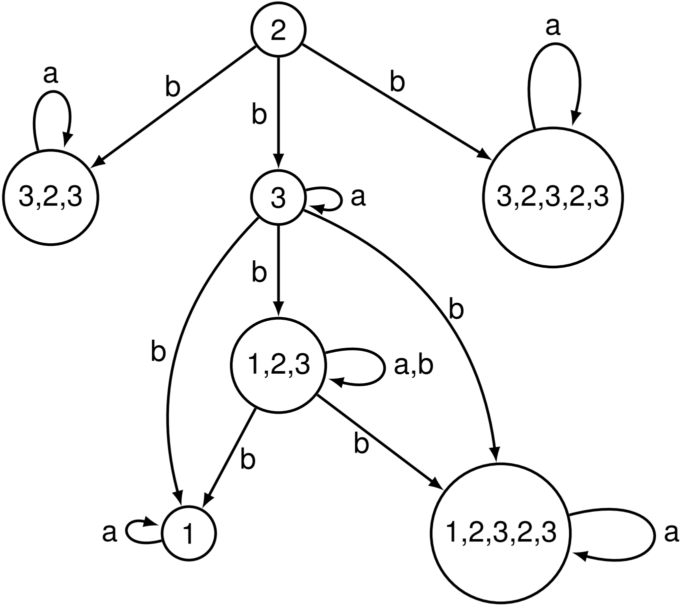
1. Entropy vs Speed
2. Main idea
3. Core of the proof

T' can be obtained as a graph.

The graph



The graph



Formal definition

We define L and R inductively

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

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 G is the set of all words where there is an edge from w to w' labeled by a if $(w, w', a) \in L$.

Lemmas

Size of the vertices are seen as *weights*.

- Each execution of the TM corresponds to an infinite path
- To each infinite path corresponds an execution of the TM, of smaller weight
- The speed corresponds to the maximum average weight of a path.
- The entropy corresponds to the maximum weighted complexity of a path.

Key lemma

Speed and entropy are well approximated when considering only finite subgraphs.

The maximum average weight of a path is the limit over all finite graphs G_p of the maximum average weight of a path in G_p.

The maximum weighted complexity of a path is the limit over all finite graphs G_p of the maximum weighted complexity of a path in G_p.

The speed

Entropy and speed are computable

Because they are computable for finite graphs.

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

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.