Information in timed words

or size of timed languages

Eugene Asarin Aldric Degorre
Cătălin Dima Bernardo Jacobo Inclán

2024, MPRI 2-8-2

Outline

Our research project 2009–

Problem

- *understand what is quantity of information in timed words/languages*
- *compute it for timed regular languages*
- *create timed theory of codes, with channels, transducers etc*
- *get insights, apply to other theoretical questions*
- *go to applications to data transmission/compression*

Our research project 2009–

Problem

- *understand what is quantity of information in timed words/languages*
- *compute it for timed regular languages*
- *create timed theory of codes, with channels, transducers etc*
- *get insights, apply to other theoretical questions*
- *go to applications to data transmission/compression*

Participants since 2009: four of us and...

Nicolas Basset, Dominique Perrin, Marie-Pierre Béal, Romain Aïssat

Our previous work

More or less solved for information in *bits per event*

This talk

A progress report for information in *bits per second*

Outline

Size of (Information in) Languages

Defining entropy (Chomsky-Miller)

- Take a language $L \subset \Sigma^*$.
- Count the words of length n: find $\#L_n$
- Typically it grows exponentially
- Growth rate - entropy $\mathcal{H}(L) = \limsup \frac{\log_2 \#L_n}{n}$

Size of (Information in) Languages

Defining entropy (Chomsky-Miller)

- Take a language $L \subset \Sigma^*$.
- Count the words of length n: find $\#L_n$
- Typically it grows exponentially
- Growth rate - entropy $\mathcal{H}(L) = \limsup \frac{\log_2 \#L_n}{n}$

Explaining the definition

- Size measure: $\#L_n \approx 2^{n\mathcal{H}}$.
- Compression rate (in bits/symbol) for a typical $w \in L$, i.e. $|w.\mathtt{zip}| \approx \mathcal{H}|w|$
- Information content of a typical $w \in L$ (bits/symbol)
- Topological entropy of a subshift.

Entropy of Regular Languages

Computing $\mathcal{H}(L(A))$ for a deterministic A

- Remove unreachable states
- Write down the adjacency matrix M.
- Compute $\rho = \rho(M)$ - its spectral radius.
- Then $\mathcal{H} = \log \rho$.

Entropy of Regular Languages

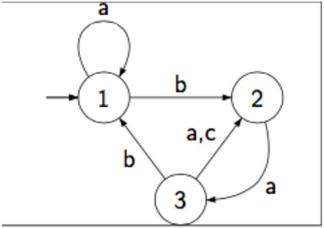
Computing $\mathcal{H}(L(A))$ for a deterministic A

- Remove unreachable states
- Write down the adjacency matrix M.
- Compute $\rho = \rho(M)$ - its spectral radius.
- Then $\mathcal{H} = \log \rho$.

Proof

- $\#L_n(i \to j) = M^n_{ij}$
- Hence $\#L_n =$ sum of some elements of M^n
- Perron-Frobenius theory of nonnegative matrices
  $\Rightarrow \#L_n \approx \rho(M)^n \Rightarrow \mathcal{H}(L) = \log \rho(M)$

Entropy of regular languages — example



$$M = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 2 & 0 \end{pmatrix}$$

- Words of lengths $0, 1, 2 \ldots$:
  $\{\varepsilon\}; \{a, b\}; \{aa, ab, ba\}; \{aaa, aab, aba, baa, bab, bac\};$
  $\{aaaa, aaab, aaba, abaa, abab, abac, baaa, bab, baca, baba, babb\} \ldots$
- Cardinalities: 1,2,3,6,11, ...
- $|L_n| \approx (1.80194)^n = \rho(M)^n = 2^{0.84955n}$.
  entropy: $\mathcal{H} = \log \rho(M) \approx 0.84955$.

Outline

Constrained channel coding

Given:

- Language S: possible messages issued by a source
- Language C: words that can be transmitted through a channel

Goal:

- How to encode the messages for transmission?

The EFMPlus code [Immink]

Used in DVD.

Description of the coding problem

- Source: $\{0,1\}^*$
- Channel: words of $\{0,1\}^*$ without blocks 11, 101, 00000000000.

Efficiency of EFMPlus

- EFMPlus rate: $1/2$.
- Optimal rate for this problem: 0.5418.

Discrete case: definition of coding

Definition

$\phi : S \to C$ is an encoding with rate α and delay d if:

- Its rate is bounded by α: $|w| \geq \alpha|\phi(w)|$: no size explosion
- It is injective with delay d: if $|w| = |w'|$ et $|u| = |u'| \geq d$ then $\phi(wu) = \phi(w'u') \Rightarrow w = w'$: decoding is possible

Discrete case: characterization

Definition (a key tool)

The entropy rate of a language: $h(L) = \lim_{n\to\infty} \frac{\log |L_n|}{n}$

Intuition:

$h(L)$ is the information content of the language in bits/symbol

Theorem (Existence of encodings)

For "regular" S and C:

- *Sufficient condition:* $\alpha h(S) < h(C)$
- *Necessary condition:* $\alpha h(S) \leq h(C)$

Outline

I will recall now

Timed automata and languages: our main object of study

A distance on timed words, because we should proceed with some precision ε wrt to a distance

ε-entropy and capacity: a standard way of measuring *information* in continuous sets

Pseudo-distance on timed words — [ABD18]

Required: a distance

- should be meaningful
- should compare timed words with different numbers of events
- should be compact on bounded duration timed words

Definition

Given $u = (a_1, t_1) \dots (a_n, t_n)$ and $v = (b_1, s_1) \dots (b_m, s_m)$, let

$$\overrightarrow{d}(u, v) = \overleftarrow{d}(v, u) = \max_i \min_j \{|t_i - s_j| : b_j = a_i\}; \quad d = \max(\overrightarrow{d}, \overleftarrow{d})$$

Comment

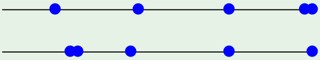
$\overrightarrow{d}(u, v)$ is small if for each event in u, exists a matching event in v, at a close date.

Distance: examples

Definition (from the previous slide)

$$\overrightarrow{d}(u,v) = \overleftarrow{d}(v,u) = \max_i \min_j \{|t_i - s_j| : b_j = a_i\}\,; \quad d = \max(\overrightarrow{d}, \overleftarrow{d})$$

Example (for one letter it is Hausdorff distance)



$u = \{0.7, 1.8, 3, 4, 4.1\}$

$v = \{0.9, 1, 1.7, 3, 4.1\}$

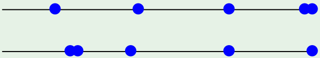
$$\overrightarrow{d}(u,v) = 0.2; \quad \overleftarrow{d}(u,v) = 0.3; \quad d(u,v) = 0.3$$

Distance: examples

Definition (from the previous slide)

$$\overrightarrow{d}(u,v) = \overleftarrow{d}(v,u) = \max_i \min_j \{|t_i - s_j| : b_j = a_i\}; \quad d = \max(\overrightarrow{d}, \overleftarrow{d})$$

Example (for one letter it is Hausdorff distance)

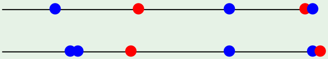


$u = \{0.7, 1.8, 3, 4, 4.1\}$

$v = \{0.9, 1, 1.7, 3, 4.1\}$

$$\overrightarrow{d}(u,v) = 0.2; \quad \overleftarrow{d}(u,v) = 0.3; \quad d(u,v) = 0.3$$

Example (for two letters)



$(a, .7), (b, 1.8), (a, 3), (b, 4), (a, 4.1)$

$(a, .9), (a, 1), (b, 1.7), (a, 3), (a, 4.1), (b, 4.2)$

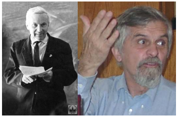
$$\overrightarrow{d}(u,v) = 0.2; \quad \overleftarrow{d}(u,v) = 0.3; \quad d(u,v) = 0.3$$

Reminder: information in continuous sets

- Q: Given a continuous set M, how much information contains $x \in M$ (what is the file size to describe x)?
- A: ∞, infinitely many bits needed... it was a stupid question.

Reminder: information in continuous sets

- Q: Given a continuous set M, how much information contains $x \in M$ (what is the file size to describe x)?
- A: ∞, infinitely many bits needed... it was a stupid question.
- Q: Given a continuous set M, and $\varepsilon > 0$, how much information contains $x \in M$ (what is the file size to describe x with precision $\varepsilon > 0$)?
- A: Nice question, the answer by Kolmogorov & Tikhomirov is ε-entropy (and ε-capacity).



Defining ε-entropy

Definition (ε-net)

Given M a metric space and $\varepsilon > 0$, a subset $S \subset M$ is an ε-net if
$\forall x \in M\, \exists y \in S : d(x, y) < \varepsilon$
If M is compact, a finite ε-net always exists.

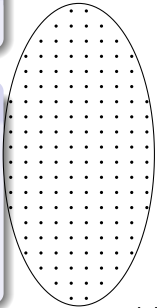
Definition (ε-entropy)

$H_\varepsilon(M) = \log \min\{|S| : S \subset M \text{ an } \varepsilon\text{-net}\}$

Explanation

To describe $x \in M$ with precision ε in $H_\varepsilon(M)$ bits:

- fix an optimal ε-net S;
- choose $y \in S$ such that $d(x, y) < \varepsilon$;
- write in binary the ordinal number of y in S.



Entropy & capacity: interpretation & inequalities

ε-entropy

Minimal information needed to describe A with precision ε

ε-capacity

Maximal information distinguishable in A if observed with error ε

Inequalities

$$\mathcal{C}_{2\varepsilon}(A) \leq \mathcal{H}_{\varepsilon}(A) \leq \mathcal{C}_{\varepsilon}(A)$$

Outline

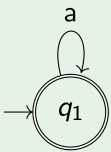
Our measure of information per time unit

Definition (Bandwidth of a timed language)

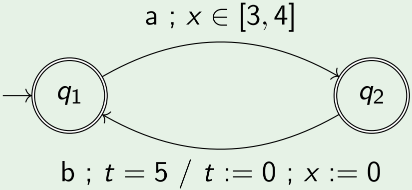
$$\mathcal{BH}_\varepsilon(L) = \limsup_{T\to\infty} \frac{\mathcal{H}_\varepsilon(L_T)}{T} \qquad \mathcal{BC}_\varepsilon(L) = \limsup_{T\to\infty} \frac{\mathcal{C}_\varepsilon(L_T)}{T}$$

Shorthands: $\mathcal{B}$ for any of $\mathcal{BC}, \mathcal{BH}$. Also $\mathcal{B}(\mathcal{A})$ for $\mathcal{B}(L(\mathcal{A}))$

Example



$$\mathcal{BC}_\varepsilon(L) = 1/\varepsilon$$
$$\mathcal{BH}_\varepsilon(L) = 1/2\varepsilon$$



$$\mathcal{BC}_\varepsilon(L) = \frac{1}{5}\log(\lceil 1/\varepsilon \rceil)$$
$$\mathcal{BH}_\varepsilon(L) = \frac{1}{5}\log(\lceil 1/2\varepsilon \rceil)$$

Why bandwidth is relevant?

Why, indeed?

- mathematically natural measure of info/time
- always $< \infty$, often > 0.
- our previous work on channel coding- Bernardo's MPRI internship, [Formats'22]
  - $\mathcal{B}(\text{Source}) < \mathcal{B}(\text{Channel}) \Rightarrow$ bounded-delay coding is possible
  - $\mathcal{B}(\text{Source}) > \mathcal{B}(\text{Channel}) \Rightarrow$ it is impossible

  up to some dirty details.
- and it will provide theoretical insights

Two problems

Problem (Our big challenge, not yet completely done)

Given a timed automaton $\mathcal{A}$ and ε, compute the bandwidth $\mathcal{B}_\varepsilon(\mathcal{A})$, as precisely as possible.

Problem (In next slides)

Given a timed automaton $\mathcal{A}$, explore the rough asymptotic behavior of $\mathcal{B}_\varepsilon(\mathcal{A})$ w.r.t. $\varepsilon \to 0$.

Outline

Classification

Definition (Three classes)

A timed language L is

meager whenever $\mathcal{B}_\varepsilon(L) = O(1)$

normal whenever $\mathcal{B}_\varepsilon(L) = \Theta(\log 1/\varepsilon)$

obese whenever $\mathcal{B}_\varepsilon(L) = \Theta(1/\varepsilon)$

as $\varepsilon \to 0$.

Before we start the examples

Trivial remarks on information

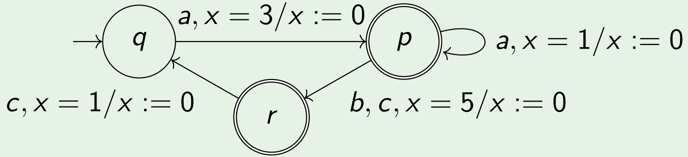
One letter $a \in \Sigma$ contains $\log(\#\Sigma)$ bits, i.e. $\mathcal{H}_\varepsilon(\Sigma) = \mathcal{C}_\varepsilon(\Sigma) = \log(\#\Sigma)$.

One real number $x \in [5,6]$ (with precision ε) contains $\log(1/\varepsilon)$ bits, i.e.

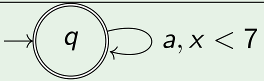
$$\mathcal{H}_{\varepsilon/2}[5,6] \approx \mathcal{C}_\varepsilon[5,6] \approx \log(1/\varepsilon).$$

Meager automata: $\mathcal{BH}_\varepsilon = O(1)$, simple examples

Example



discrete



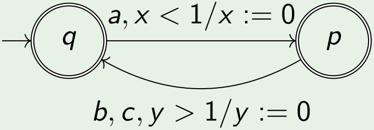
finite lifespan

Explanation of their meagerness

Impossible to encode info in reals, on the long run. Only discrete information.

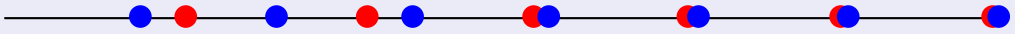
Meager automata: $\mathcal{BH}_\varepsilon = O(1)$, a subtle example

Example (A Zeno-like automaton)



- distance < 1 between consecutive a
- distance > 1 between consec. b/c
- they interleave

A typical word



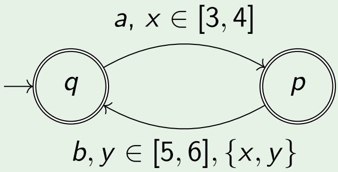
$$(a, .9)(b, 1.2)(a, 1.8)(b, 2.4)(a, 2.7)(b, 3.5)(a, 3.6)(b, 4.52)(a, 4.59)$$
$$(b, 5.53)(a, 5.58)(b, 6.535)(a, 6.575)$$

Explanation of their meagerness

Impossible to encode info in reals, on the long run, a and b/c are stuck. Only discrete information.

Normal automata: $\mathcal{BH}_\varepsilon = \Theta(\log 1/\varepsilon)$

Example

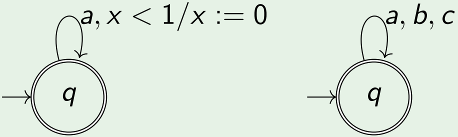


Explanation of its normality

Every 5-6 sec we can freely choose 2 real durations. Each time we transmit $\log 1/\varepsilon$ bits. Thus $\mathcal{B}_\varepsilon = \frac{2}{5} \log 1/\varepsilon$ bit/sec

Obese automata: $\mathcal{BH}_\varepsilon = \Theta(1/\varepsilon)$

Example

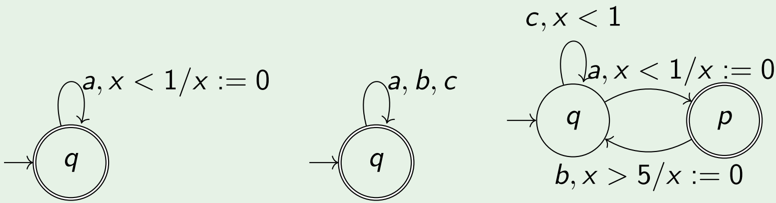


Explanation of their obesity

- Every ε sec we transmit one bit (a or nothing). $\mathcal{B} = 1/\varepsilon$ bit/sec.
- Every ε sec we transmit 3 bits (subset of $\{a, b, c\}$), thus $\mathcal{B} = 3/\varepsilon$.

Obese automata: $\mathcal{BH}_\varepsilon = \Theta(1/\varepsilon)$

Example



Explanation of their obesity

- Every ε sec we transmit one bit (a or nothing). $\mathcal{B} = 1/\varepsilon$ bit/sec.
- Every ε sec we transmit 3 bits (subset of $\{a, b, c\}$), thus $\mathcal{B} = 3/\varepsilon$.
- After 1 sec of transmission we spend 5 sec to reset clocks, $\mathcal{B} \approx 1/6\varepsilon$.

Towards structural characterization of the 3 classes

Plan: given a bounded deterministic timed automaton

- Split its states into regions (unit "triangles" — simplices)
- Trim it (remove useless states)
- Map each edge (and path) into a finite monoid
- Define two patterns[a] NM and O (in terms of monoids)
- Characterize classes by presence/absence of patterns

[a] non-meagerness and obesity terns

Our results on one slide

We define two patterns *NM* and *O*

Theorem (Structural criteria)

- $\mathcal{A}$ *contains pattern NM* $\Rightarrow \mathcal{B}_\varepsilon(\mathcal{A}) = \Omega(\log 1/\varepsilon)$ *(non-meager).*
- $\mathcal{A}$ *does not contain NM* $\Rightarrow \mathcal{B}_\varepsilon(\mathcal{A}) = O(1)$ *(meager).*
- $\mathcal{A}$ *contains pattern O* $\Rightarrow \mathcal{B}_\varepsilon(\mathcal{A}) = \Omega(1/\varepsilon)$ *(obese).*
- $\mathcal{A}$ *does not contain O* $\Rightarrow \mathcal{B}_\varepsilon(\mathcal{A}) = O(\log 1/\varepsilon)$ *(non-obese).*

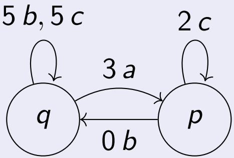
Theorem (Classification)

Every deterministic timed automaton is meager or normal or obese. The classification problem is PSPACE*-complete.*

Outline

Simply-Timed Graphs

An example



Its language

Words accepted by the graph, e.g. $(5, b)(10, c)(13, a)(13, b)(16, a)(18, c)$

Bandwidth of Simply-Timed Graphs

Definition

For a set of timed words S, its size $\Upsilon(S)$ is the cardinality of the largest 0-separated set in S

Definition

The growth rate of an STG $\mathcal{A}$ is defined as

$$\beta(\mathcal{A}) = \lim_{T\to\infty} \frac{\log \Upsilon(L_T(\mathcal{A}))}{T},$$

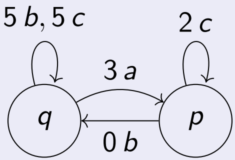
Remark

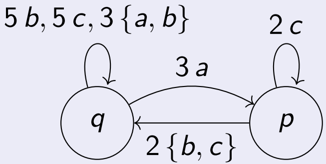
Let $1/D$ be the smallest non-zero timing in the STA. For $\varepsilon < 1/2D$ the growth rate coincides with both bandwidths:

$$\mathcal{BC}_\varepsilon(\mathcal{A}) = \mathcal{BH}_\varepsilon(\mathcal{A}) = \beta(\mathcal{A}).$$

Computing the growth rate of an STG

0-elimination and determinization





z-matrix and characteristic equation

$$m_{ij}(z) = \sum_{(q_i,d,a,q_j)\in\Delta} z^d,$$

$$M(z) = \begin{pmatrix} z^3 + 2z^5 & z^3 \\ z^2 & z^2 \end{pmatrix}$$

- Characteristic equation: $\det(I - M(z)) = 0$
- z_0: its root of smallest modulus
- The growth rate: $\beta(\mathcal{A}) = -\log_2 |z_0|$
- In our case: $2z^7 - 2z^5 - z^3 - z^2 + 1 = 0$, $z_0 \approx 0.698776$, $\beta(\mathcal{A}) \approx 0.517098$

Outline

Barycenters of a clock region

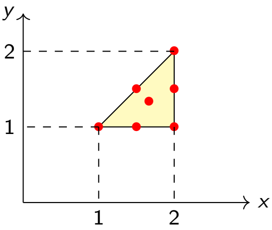
Definition (faces of a region)

A $(k-1)$-dimensional face is the convex hull of any subset of $k > 0$ vertices.

Faces are simplices. Each vertex or the whole closed region are considered as faces.

Definition (barycenters of a region)

To each such face f with vertices $v_1, \ldots, v_k$ we associate its *barycenter* $\alpha(f) = \frac{1}{k} \sum_{i=1}^{k} v_k$

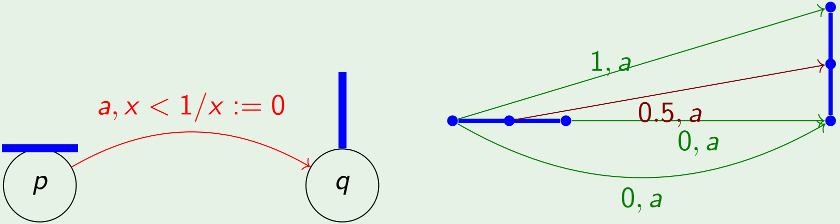


From Meager Automata to STG, 1/2

Definition (barycentric abstraction of a transition $p \xrightarrow{\delta} q$)

Link barycenters b_1 of p and b_2 of q whenever their faces have the same dimension and $L_\delta(b_1, b_2) = \{ta\}$ is a singleton. Label with duration t and letter a.

Example (a transition and its abstraction)

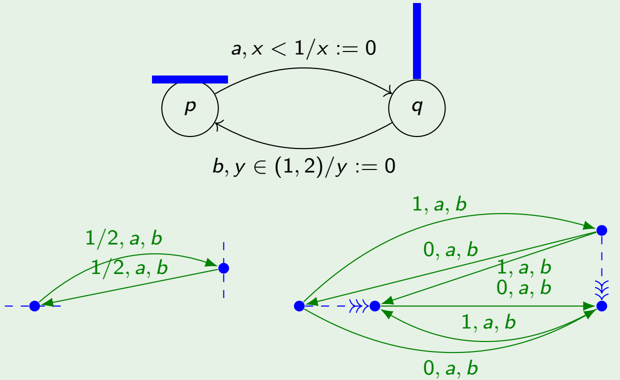


From Meager Automata to STG, 2/2

Definition (barycentric abstraction of an RsTA)

Abstract every transition of $\mathcal{A}$, the resulting STG is denoted $\alpha(\mathcal{A})$.

Example (an RsTA and its abstraction)



Main results

Theorem (characterisation of the bandwidth of Meager TA)

- *barycentric abstraction preserves the bandwidth of Meager RsTA*

$$\mathcal{B}_\varepsilon(\mathcal{A}) = \mathcal{B}_0(\alpha(\mathcal{A})) \text{ for } \varepsilon \text{ small enough}$$

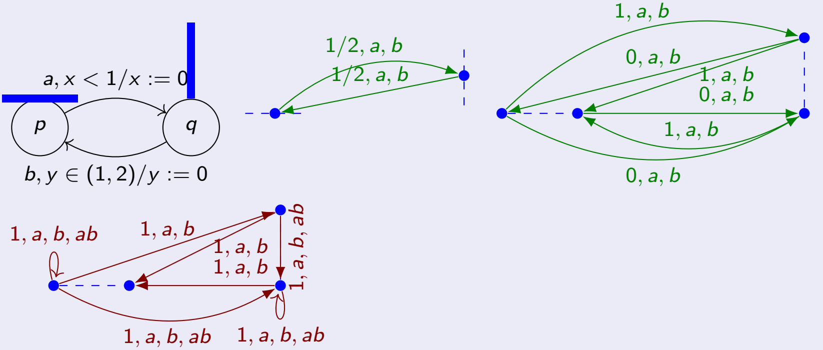
- *thus the bandwidth of meager TA is computable (as* log *of an algebraic number)*

Overall algorithm: given a meager TA

- put it into region-split form
- compute its barycentric abstraction (STG)
- proceed with $0-$elimination and determinization
- write down the characteristic equation
- find its smallest root
- compute the bandwidth

Example: computing the bandwidth of a meager TA

A meager RsTA, barycentric abstraction, 0-elimination



z-matrix and bandwidth

$$M(z) = \text{block diagonal with blocks } \begin{pmatrix} 0 & 2z^{1/2} \\ 2z^{1/2} & 0 \end{pmatrix}; \quad \begin{pmatrix} 3z & 0 & 3z & 2z \\ 0 & 0 & 0 & 0 \\ 0 & 2z & 3z & 0 \end{pmatrix}$$

Outline

Two challenges

compute the bandwidth

bandwidth $\approx \alpha, \alpha \log(1/\varepsilon), \alpha/\varepsilon$, compute the coefficient α

- already done for meager automata (CIAA'24)
- We believe we can do it for obese ones
- normal ones still resist

Can be started as MPRI internship!

invent optimal timed transducers, codes, compressions