Entropy of Timed Regular Languages

Eugene Asarin[1], Nicolas Basset[2], Marie-Pierre Béal[3] **Aldric Degorre**[1] and Dominique Perrin[3]

[1]IRIF — Université de Paris-Diderot
[2]LIP6 — Université Pierre et Marie Curie
[3]LIGM — Université Paris Est - Marne-la-Vallée

May 10 2016 – EQINOCS final Workshop – IRIF – Paris

Measuring Size of Timed Languages: Why?

Motivations

- Verification (original motivation):
  - Quality of an over-approximation $L \supset M$ (compare $\#L$ and $\#M$)
  - Quantitative model-checking
- Information theory:
  - Information content
  - Security: timed information flow
  - Timed channel capacity [ABBDP'12]
- Quasi-uniform random simulation [B'13]
- And of course: links with symbolic dynamics (entropy of timed subshifts)

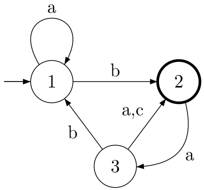
Reminder: Size of Languages

Size and entropy of discrete languages

- Take a language $L \subset \Sigma^*$.
- **Count** its words[a] of length n ($\#L_n$, $L_n =_{def} \Sigma^n \cap L$)

[a] we could also count prefixes or factors

An automaton:

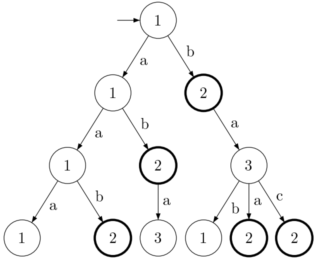


Reminder: Size of Languages

Size and entropy of discrete languages

- Take a language $L \subset \Sigma^*$.
- **Count** its words[a] of length n ($\# L_n$, $L_n =_{def} \Sigma^n \cap L$)
- Typically: exponential growth
- Growth rate - **entropy** $\mathcal{H}(L) = \limsup \frac{\log_2 \# L_n}{n}$

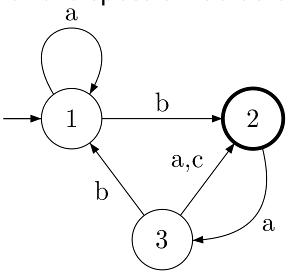
[a] we could also count prefixes or factors



- Languages $L_0, \ldots, L_4$:
  $\emptyset$; $\{b\}$; $\{ab\}$; $\{aab, baa, bac\}$;
  $\{aaab, abaa, abac, babb\}$;
  $\{aaaab, aabaa, aabac, ababb, babab,$
  $baaaa, baaac, bacaa, bacac\}$ ...
- Cardinalities: 0,1,1,3,4,9, ...

Computing the entropy of regular languages

Entropy for a deterministic automaton
= logarithm of the spectral radius of the adjacency matrix.



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

Spectral radius: maximal norm of the eigenvalues For this M: $\rho(M) \approx 1.80194$;
entropy: $\mathcal{H} = \log \rho(M) \approx 0.84955$.

Context

Timed automata

- A model for verification of real-timed systems
- Invented by Alur and Dill in early 1990s
- Precursors: time Petri nets (Berthomieu)
- Now: an efficient model for verification, supported by tools (UPPAAL)
- A popular research topic ($>$ 8000 citations for papers by Alur and Dill)
  - modeling and verification
  - decidability and algorithmics
  - automata and language theory
  - very recent: dynamics
- Inspired by TA: hybrid automata, data automata, automata on nominal sets

Foreword: timed words and languages

- A word: $u = abbabb$ represents a sequence of events in some Σ.
- A timed word: $w = 0.8a2.66b1.5b0a3.14159b2.71828b$ represents a sequence of events and delays.
- It lives in a timed monoid $\Sigma^* \oplus \mathbb{R}_+$ (but nevermind this!).
- For us it sits in $(\mathbb{R}_+ \times \Sigma)^*$ (words on some infinite alphabet), that is $w = (0.8, a), (2.66, b), (1.5, b), (0, a), (3.14159, b), (2.71828, b)$.
- Geometrically w is a point in several copies of $\mathbb{R}^n$:

$$w = (0.8, 2.66, 1.5, 0, 3.14159, 2.71828) \in \mathbb{R}^6_{abbabb}$$

- A timed language is a set of timed words – examples below.

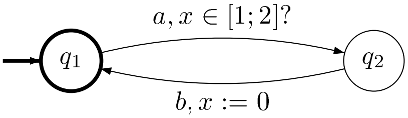
So, what is a TA?

Recipe for making a timed automaton :

- take a finite automaton;
- add some variables $x_1, \ldots, x_n$, called clocks;
- add guards to transitions (e.g. $x_3 < 7$);
- add resets to transitions (e.g. $x_2 := 0$);
- make all clocks run at speed $\dot{x}_i = 1$ everywhere and interpret behaviors in continuous time;
- enjoy!

An example of timed automaton

- Timed automaton $\mathcal{A}$:



- A run:
  $(q_1, 0) \xrightarrow{1.83} (q_1, 1.83) \xrightarrow{a} (q_2, 1.83) \xrightarrow{4.1} (q_2, 5.93) \xrightarrow{b} (q_1, 0) \xrightarrow{1} (q_1, 1) \to \dots$
- Its trace $1.83a4.1b1a$ is a timed word.
- The timed language of the TA: set of all traces starting in q_1, ending in q_1:
  $\{t_1 a s_1 b t_2 a s_2 b \dots t_n a | \forall i. t_i \in [1;2]\}$

Observation: clock value of x: time since the last reset of x.

Outline

Talking about size

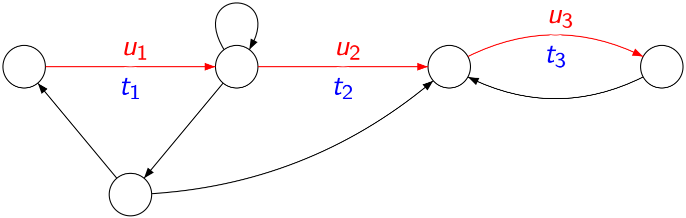
- Timed languages typically are non-countable sets (continuous choice of delays).
- How does one describe the "size" of such an object?
  (and thus translate a nice classical theory to the realm of timed automata / timed shifts $\rightarrow$ extra-motivation).

Talking about size

- Timed languages typically are non-countable sets (continuous choice of delays).
- How does one describe the "size" of such an object?
  (and thus translate a nice classical theory to the realm of timed automata / timed shifts → extra-motivation).

The idea: timed regular languages must be seen as unions of polytopes → instead of counting words, we sum up their volumes.

Volume and Entropy for Timed Languages



- Choice of a timed word $(\vec{t}, u) \in L_n$ = discrete choice of path u (untiming) + continuous choice of delay vector $\vec{t}$ (timing).
- Given u, $L_u = \{\vec{t} \mid (\vec{t}, u) \in L_n\} \subseteq \mathbb{R}^n$ is a polytope (e.g. hypercube, simplex...)
- Measure of L_n, $\mathtt{Vol}(L_n) = \sum_{u \in \Sigma^n} \mathtt{Vol}(L_u)$
- (Rate of volumic) entropy: $\mathcal{H} = \lim \frac{1}{n} \log_2(\mathtt{Vol}(L_n))$

Simple n-volumes

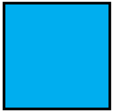
hypercubes

dimension 1

$t_1 \leq d$

Volume d

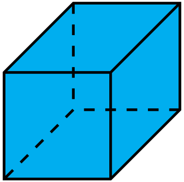
dimension 2



$t_1, t_2 \leq d$

Volume d^2

dimension 3



$t_1, t_2, t_3 \leq d$

Volume d^3

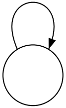
dimension n

?

$t_1, \ldots, t_n \leq d$

Volume d^n

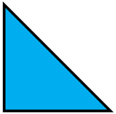
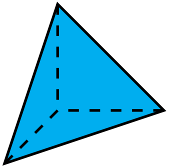
$a, x \leq d / x := 0$



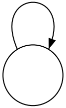
Timed word : $(t_1, a)(t_2, a) \ldots (t_n, a)$

Simple n-volumes

simplices

dimension 1	dimension 2	dimension 3	dimension n
			?
$t_1 \leq 1$	$t_1 + t_2 \leq 1$	$t_1 + t_2 + t_3 \leq 1$	$t_1 + \cdots + t_n \leq 1$
Volume 1	Volume $1/2$	Volume $1/6$	Volume $1/n!$

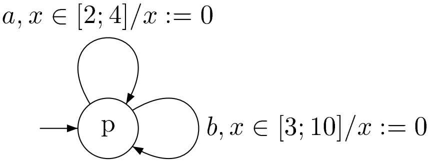
$a, x \leq 1$



Timed word : $(t_1, a)(t_2, a) \ldots (t_n, a)$

Volume and entropy of timed automata

Example 1: rectangles

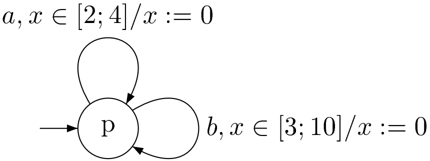


Language:

$L_1 = ([2; 4]a + [3; 10]b)^*$

Volume and entropy of timed automata

Example 1: rectangles

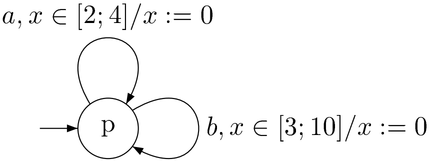


Language:
$L_1 = ([2;4]a + [3;10]b)^*$

- For the untiming *bbab* the set of timings is a 4-**rectangle**: $[3;10] \times [3;10] \times [2;4] \times [3;10]$, its volume $7 \cdot 7 \cdot 2 \cdot 7 = 686$.

Volume and entropy of timed automata

Example 1: rectangles

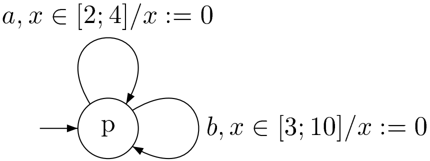


Language:
$L_1 = ([2;4]a + [3;10]b)^*$

- For the untiming *bbab* the set of timings is a 4-**rectangle**: $[3;10] \times [3;10] \times [2;4] \times [3;10]$, its volume $7 \cdot 7 \cdot 2 \cdot 7 = 686$.
- For an untiming in $\{a,b\}^n$ with $a \times k; b \times (n-k)$, the set of timings is a rectangle, volume $2^k 7^{n-k}$

Volume and entropy of timed automata

Example 1: rectangles

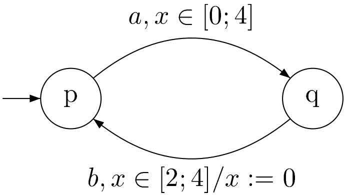


Language:
$L_1 = ([2;4]a + [3;10]b)^*$

- For the untiming *bbab* the set of timings is a 4-**rectangle**: $[3;10] \times [3;10] \times [2;4] \times [3;10]$, its volume $7 \cdot 7 \cdot 2 \cdot 7 = 686$.
- For an untiming in $\{a,b\}^n$ with $a \times k; b \times (n-k)$, the set of timings is a rectangle, volume $2^k 7^{n-k}$
- Volume: $V_n(L_1) = \sum_{k=0}^{n} C_n^k 2^k 7^{n-k} = (2+7)^n = 9^n$,
- Entropy: $\mathcal{H}(L_1) = \log 9 \approx 3.17$.

Volume and entropy of timed automata

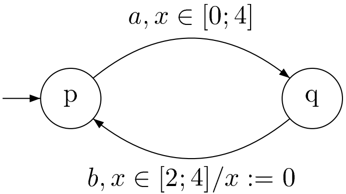
Example 1: trapezia

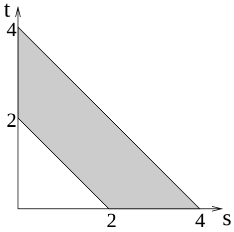


- Language : $t_1 a s_1 b t_2 a s_2 b \ldots t_k a s_k b$ such that $2 \leq t_i + s_i \leq 4$

Volume and entropy of timed automata

Example 1: trapezia

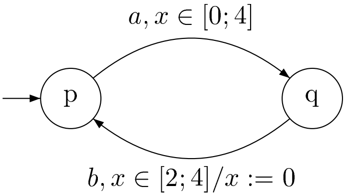


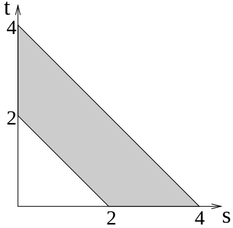


- Language : $t_1 a s_1 b t_2 a s_2 b \dots t_k a s_k b$ such that $2 \leq t_i + s_i \leq 4$
- For the only n-untiming $w = (ab)^{n/2}$ the set of timings is a product of $n/2$ **trapezia**.

Volume and entropy of timed automata

Example 1: trapezia

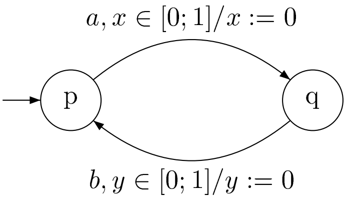




- Language : $t_1 a s_1 b t_2 a s_2 b \dots t_k a s_k b$ such that $2 \leq t_i + s_i \leq 4$
- For the only n-untiming $w = (ab)^{n/2}$ the set of timings is a product of $n/2$ **trapezia**.
- Volume: $V_n(L_2) = 6^{n/2}$,
- Entropy: $\mathcal{H}(L_2) = \log 6/2 \approx 1.29$.

Volume and entropy of timed automata

Example 3: strange polytopes



- Language : $L_3 = \{t_1 a t_2 b t_3 a t_4 b \ldots | t_i + t_{i+1} \in [0;1]\}$
- For the only n-untiming $w = (ab)^{n/2}$ the set of timings is **a strange polytope**.
- Volume: see below
- Entropy: see below

General case: some minor restrictions

For the rest of the paper, all our TAs actually are BDTAs:

Bounded Deterministic Timed Automaton

A BDTA is a timed automaton with following contraints:

1. it is deterministic.
2. its guards are conjunctions of bounded intervals.[a]

[a] We allow "punctual" guards (singletons), in spite of induced pathologies.

Functional Analysis Approach

The first approach is based on results from functional analysis.

Outline

- We find a recurrence for computing volumes.
- Volumes functions = points of some functional space.
- Recurrence = some linear operator Ψ on this space.
- The study of volume and entropy thus reduces to the study of the properties of Ψ

All of this is in [ABD'15].

Recurrence for Languages and Volumes

Idea:

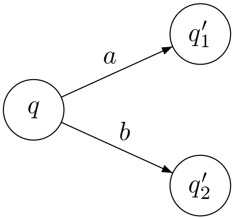
language recurrent equations $\longrightarrow$ volume recurrent equations

Recurrence for Languages and Volumes

Idea:

language recurrent equations $\longrightarrow$ volume recurrent equations

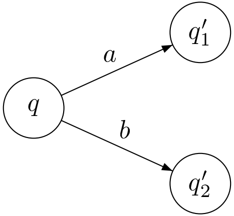
Discrete automata: what n-**language** $L_n(q)$ can you read from state q?



$$L_{k+1}(q) = aL_k(q'_1) + bL_k(q'_2)$$

Recurrence for Languages and Volumes

Discrete automata: what *n*-**language** $L_n(q)$ can you read from state q?



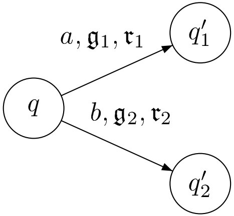
$$L_{k+1}(q) = aL_k(q'_1) + bL_k(q'_2)$$

Language recurrence

$$\begin{aligned} L_0(q) &= \varepsilon; \\ L_{k+1}(q) &= \bigcup_{(q,a,q')\in\Delta} a \cdot L_k(q'). \end{aligned}$$

Recurrence for Languages and Volumes

Timed automata: what n-**language** $L_n(q, \mathbf{x})$ can you read from state $(q, \mathbf{x})$?



$$L_{k+1}(q, \mathbf{x}) = \bigcup_{\mathbf{x}+\tau \in \mathfrak{g}_1} \tau a \cdot L_k(q'_1, \mathfrak{r}(\mathbf{x}+\tau)) + \bigcup_{\mathbf{x}+\tau \in \mathfrak{g}_2} \tau b \cdot L_k(q'_2, \mathfrak{r}_2(\mathbf{x}+\tau))$$

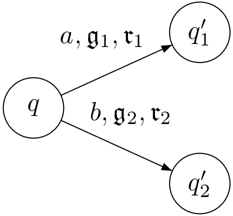
Language recurrence

$$L_0(q, \mathbf{x}) = \varepsilon;$$

$$L_{k+1}(q, \mathbf{x}) = \bigcup_{(q,a,\mathfrak{g},\mathfrak{r},q') \in \Delta} \bigcup_{\tau : \mathbf{x}+\tau \in \mathfrak{g}} \tau a \cdot L_k(q', \mathfrak{r}(\mathbf{x}+\tau)).$$

Recurrence for Languages and Volumes

Deterministic timed automata: what n-**volume** $V_n(q, \mathbf{x})$ does $L_n(q, \mathbf{x})$ have?



$$v_{k+1}(q, \mathbf{x}) = \int_{\mathbf{x}+\tau \in \mathfrak{g}_1} v_k(q'_1, \mathfrak{r}(\mathbf{x}+\tau))d\tau + \int_{\mathbf{x}+\tau \in \mathfrak{g}_2} v_k(q'_2, \mathfrak{r}_2(\mathbf{x}+\tau))d\tau$$

Volume recurrence

$$v_0(q, \mathbf{x}) = 1;$$
$$v_{k+1}(q, \mathbf{x}) = \sum_{(q,a,\mathfrak{g},\mathfrak{r},q') \in \Delta} \int_{\tau : \mathbf{x}+\tau \in \mathfrak{g}} v_k(q', \mathfrak{r}(\mathbf{x}+\tau))\, d\tau.$$

First Theorem

Theorem (Volume is computable)

v_n is polynomial on each clock region.
$V_n(= v_n(q_0, \mathbf{0}))$ is a rational number.
They can be computed using the recurrence above.

Example (Volume of L_3)

The volume for our running example is

$$V_n(L_3) = \int_0^1 dt_1 \int_0^{1-t_1} dt_2 \int_0^{1-t_2} dt_3 \ldots \int_0^{1-t_{n-1}} dt_n$$

That is[a]

$$1; \frac{1}{2}; \frac{1}{3}; \frac{5}{24}; \frac{2}{15}; \frac{61}{720}; \frac{17}{315}; \frac{277}{8064}; \ldots$$

[a] ... which also happens to be the coefficients of the Taylor expansion of $(\sin x + 1)/\cos x - 1$!

Reconsidering the Recurrence for Volumes

Volume recurrence formula

$$\begin{aligned} v_0(q,\mathbf{x}) &= 1; \\ v_{k+1}(q,\mathbf{x}) &= \sum_{(q,a,\mathfrak{g},\mathfrak{r},q')\in\Delta} \int_{\tau:\mathbf{x}+\tau\in\mathfrak{g}} v_k(q',\mathfrak{r}(\mathbf{x}+\tau))\,d\tau. \end{aligned}$$

Can we use these equations to compute entropy?

Reconsidering the Recurrence for Volumes

Volume recurrence formula

$$\begin{aligned} v_0(q,\mathbf{x}) &= 1; \\ v_{k+1}(q,\mathbf{x}) &= \sum_{(q,a,\mathfrak{g},\mathfrak{r},q')\in\Delta} \int_{\tau:\mathbf{x}+\tau\in\mathfrak{g}} v_k(q',\mathfrak{r}(\mathbf{x}+\tau))\,d\tau. \end{aligned}$$

Volume recurrence – in 12 symbols

Same formulas, shorter version:

$$\begin{aligned} v_0 &= 1; \\ v_{k+1} &= \Psi v_k, \end{aligned}$$

where Ψ is a positive linear operator on some functional space.

Toward the Main Theorem

We want to find $\mathcal{H}$ by studying (the iterates of) Ψ.

Ψ's nice properties

- Trivial: Ψ is a linear, bounded, positive operator on a Banach space. (Ψ lives in $\mathcal{F} = C(Q \times [0; M]^n)$)
- If $\mathcal{A}$ is strongly connected of period p and $\mathcal{H} > -\infty$[a], then Ψ^p has a spectral gap.
- Results from functional analysis apply (cf. [Krasnosel'skij, Lifshits, Sobolev 89]).
- $\Rightarrow \Psi^k f \sim \rho^k f^*$ (Gelfand). For us: $v_k(q, \mathbf{x}) \sim \rho^k f^*(q, \mathbf{x})$.

[a] [AB'11]: $\mathcal{H} > -\infty$ can be checked in time exponential to the number of clocks.

Spectral gap

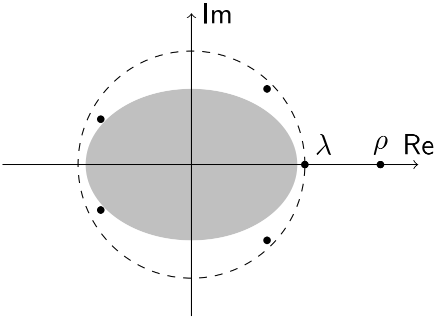


Figure: Spectrum of an operator having a gap.

Main theorem

Theorem (Main result of [ABD'15])

For a BDTA A, either $\rho(\Psi) = 0$ *(and* $\mathcal{H} = -\infty$*) or* $\mathcal{H} = \log \rho(\Psi)$.

Main theorem

Theorem (Main result of [ABD'15])

For a BDTA A, either $\rho(\Psi) = 0$ *(and* $\mathcal{H} = -\infty$*) or* $\mathcal{H} = \log \rho(\Psi)$*.*

$\mathcal{H} = \log \rho(\Psi) \to$ Are We Done?

Yes – we have a characterization of the entropy.
No – how do we know the maximal λ such that $\Psi f = \lambda f$?

- An awful integral equation ...
- How to get a number?

Main theorem

Theorem (Main result of [ABD'15])

For a BDTA A, either $\rho(\Psi) = 0$ *(and* $\mathcal{H} = -\infty$*) or* $\mathcal{H} = \log \rho(\Psi)$.

$\mathcal{H} = \log \rho(\Psi) \rightarrow$ Are We Done?

Yes – we have a characterization of the entropy.
No – how do we know the maximal λ such that $\Psi f = \lambda f$?

- An awful integral equation . . .
- How to get a number?

$\rightarrow$ reduction to ODE in a particular case

$\rightarrow$ iterative method of approximation for the general case

The Easy Case : $1\frac{1}{2}$ Clocks

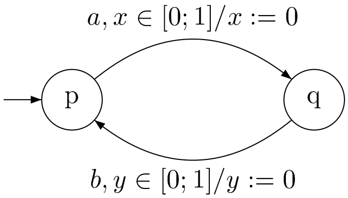
Definition ($1\frac{1}{2}$ clocks timed automata)

BDTA is $1\frac{1}{2}$ clocks $\Leftrightarrow$ after every transition at most one clock $\neq 0$.

Then $v(q, x)$ has 1-dim argument $\Rightarrow$ linear ODE: all is easy.

The Easy Case : $1\frac{1}{2}$ Clocks

Case of our favorite example



- Integral equation: $\lambda f(x) = \Psi f(x)$ with $\Psi f(x) = \int_0^{1-x} f(s)\,ds$.
- Derived twice: $\lambda^2 f''(x) = -f(x)$, with $f(1) = 0, f'(0) = 0$.
- We find: $\lambda = 2/\pi$; $f^*(x) = \cos(\frac{x\pi}{2})$
- $\Rightarrow$ entropy: $\mathcal{H} = \log(2/\pi) \approx -0.6515$

The Easy Case : $1\frac{1}{2}$ Clocks

General case

Lemma

The solutions of $\Psi v = \lambda v$ *are the solutions of the differential equation* $\lambda Y' = AY$ *satisfying* $Y(1/2) = \begin{pmatrix} X \\ X \end{pmatrix}$ *with* $M_\lambda X = 0$.

The details:

- Y is the vector of volume functions (slightly transformed)
- A can be derived directly from $\mathcal{A}$
- M_λ is slightly more involved (contains $\int_{-1/2}^{0} \exp \frac{t}{\lambda} A dt$)

The ODE has non-zero solution iff $det\ M_\lambda = 0$. Thus:

Theorem

For $1\frac{1}{2}$*-clocks BDTA,* $\mathcal{H} = \log \max\{|\lambda| \mid det\ M_\lambda = 0\}$.

General case: Iteration method for positive operators

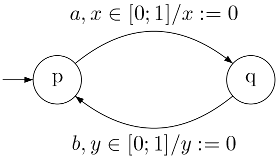
Theorem (Iteration)

For a strongly connected BDTA of period p with $\mathcal{H} > -\infty$,
$\rho_n = \|v_{(n+1)p}\| / \|v_{np}\| \to_{n\to\infty} \rho$ *with exponential speed.*

(Recall: $v_n = \Psi^n v_0$ and Ψ has a spectral gap. Thus $v_n \simeq \rho^n v_0$.)

Iteration method for positive operators

Applied to our favorite example...



n	$v_n(x)$	$\|v_n\|$	ρ_{n-1}
0	1	1	
1	$1-x$	1	1
2	$1-x-(1-x)^2/2$	1/2	0.5
3	$(1-x)/2-(1-x)^3/6$	1/3	0.6667
4	$(1-x)/3+(1-x)^4/24-(1-x)^3/6$	5/24	0.6250
5	$\frac{5}{24}(1-x)+(1-x)^5/120-(1-x)^3/12$	2/15	0.6400
6	$\frac{2}{15}(1-x)-(1-x)^6/720+(1-x)^5/120-(1-x)^3/18$	61/720	0.6354
7	$\frac{61}{720}(1-x)-(1-x)^7/5040+(1-x)^5/240-\frac{5}{144}(1-x)^3$	17/315	0.6370
8	$\frac{17}{315}(1-x)+(1-x)^8/40320-(1-x)^7/5040+(1-x)^5/360-(1-x)^3/45$	277/8064	0.63648

Table: Iterating the operator for $\mathcal{A}_3$ ($\mathcal{H}=\log(2/\pi)\approx\log 0.6366\approx -0.6515$)

Discretization Approach

The second approach is based on brute force discretization of timed automata.

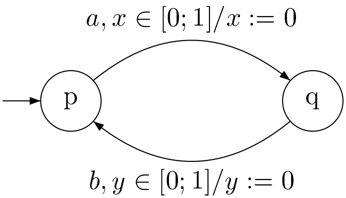
Outline

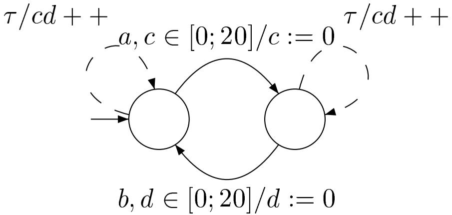
- We take a BDTA $\mathcal{A}$ (and remove punctual guards).
- We fix a discretization step ε.
- We transform $\mathcal{A}$ into a finite automaton $\mathcal{A}_\varepsilon$ on alphabet $\Sigma \cup \{\tau\}$ that approximates its behaviors up to precision ε.
- We use classical methods to compute the entropy of $\mathcal{A}_\varepsilon$.
- Finally we deduce the entropy of $\mathcal{A}$.

This approach is described in [AD'09].

Discretizing Timed Automata

An example of such a discretization:





More details:

- Take the BDTA $\mathcal{A}$. Fix $\varepsilon > 0$.
- Replace every clock x by a counter $c \approx x/\varepsilon$.
- Add to every state a $\tau, c{+}{+}$-loop (ε-time progress).
- Bounded counters $\implies$ finite state space.

Counting Words and Computing Entropy

- L^{ε}: language of the discretized automaton
  $=$ set of ε-samples of L
- $V_n(L_n) \approx \#L_n^{\varepsilon} \cdot \varepsilon^n$ (i.e. #samples · Vol(ε-ball))
- So we take the logarithm and...

Counting Words and Computing Entropy

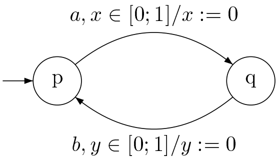
- L^ε: language of the discretized automaton
  $=$ set of ε-samples of L
- $V_n(L_n) \approx \#L_n^\varepsilon \cdot \varepsilon^n$ (i.e. #samples · Vol(ε-ball))
- So we take the logarithm and...

Theorem

Computing Entropy by Discretization [AD'09, AB'11]

$$\mathcal{H}(L) - \mathcal{H}_{discrete}(L^\varepsilon) - \log(\varepsilon) = o(1)$$

Discretization of L_3



Applying the method to the 3rd example,

- for $\varepsilon = 0.1$, we find

$$\mathcal{H} \in [\log 0.62; \log 0.653] \subset (-0.69; -0.61)$$

- and for $\varepsilon = 0.01$,

$$\mathcal{H} \in [\log 0.6334; \log 0.63981] \subset (-0.659; -0.644).$$

(reminder: $\mathcal{H} = \log(2/\pi) \approx -0.6515$)

Information theory
(Links with and applications to...)

- Volumic entropy: several information theoretical characterizations: ε-entropy (see above), Kolmogorov complexity (next slide), ...
- A concrete application: channel coding

$\rightarrow$ we generalize the classical theory of constrained channel coding for timed sources and/or timed channels.

Kolmogorov Complexity of Timed Words

Definition

Kolmogorov complexity of a word w [Kolmogorov 65]:

$$K(w) = \min \# \text{ of instructions to define } w$$

Theorem

For L a timed regular language,

$$\max_{w \in L_n} \min_{d(v,w)<\varepsilon} K(v) \approx n(\mathcal{H}(L) - \log \varepsilon)$$

Proof idea: close to discretization theorem.

The bottom line: entropy is linked to the worst case complexity of the best ε-approximation a word in L_n.

Typical problems of channel coding

Given...

- a source: $S \subseteq A^*$ (e.g. possible message, contents of a file, etc.);
- a channel: $C \subseteq A'^*$ (e.g. what can be transmit by telegraph, written on a DVD, etc.).

In this paradigm: no noise, no probability.

Questions

- Is it possible to transmit any source message via the channel?
- What would be the transmission speed?
- How to encode the message before and to decode it after transmission?

Coding: a definition

Definition ($\phi : S \to C$, encoding with rate $\alpha \in \mathbb{Q}$)

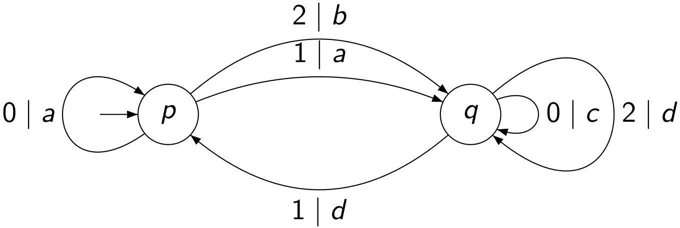
- it is of rate α, i.e. $\alpha = \frac{|w|}{|\phi(w)|}$;
- it is injective,

Coding: a definition

Definition ($\phi : S \to C$, encoding with rate $\alpha \in \mathbb{Q}$)

- it is of rate α, i.e. $\alpha = \frac{|w|}{|\phi(w)|}$;
- it is almost injective with delay d, i.e. if $|w| = |w'|$ and $|u| = |u'| = d$ then $\phi(wu) = \phi(w'u') \Rightarrow w = w'$.

A coding realized by a transducer with delay 1



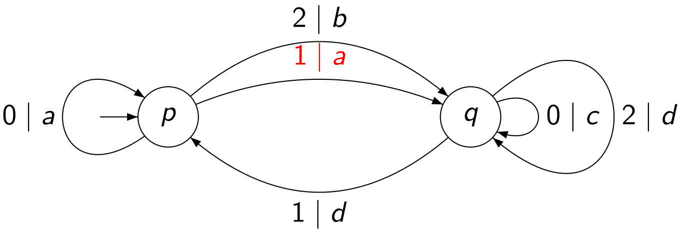
Coding: $1021 \mapsto acdd$.

Decoding: $acdd \mapsto$ 102.(1 or 2).

Properties of the transducer

- Deterministic on its input.
- Deterministic on its output with delay $d = 1$.

A coding realized by a transducer with delay 1

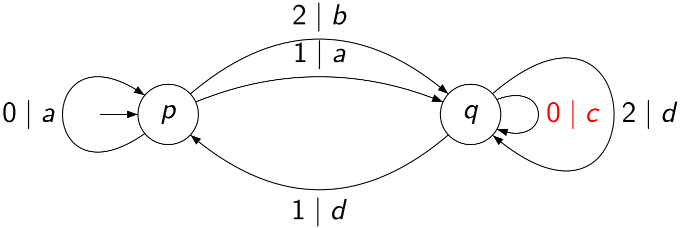


Coding: 1021↦*acdd*.
Decoding: *acdd*↦ 102.(1 or 2).

Properties of the transducer

- Deterministic on its input.
- Deterministic on its output with delay $d = 1$.

A coding realized by a transducer with delay 1



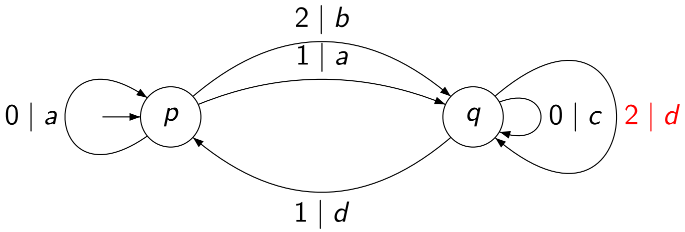
Coding: 1021↦$acdd$.

Decoding: $acdd$↦ 102.(1 or 2).

Properties of the transducer

- Deterministic on its input.
- Deterministic on its output with delay $d = 1$.

A coding realized by a transducer with delay 1

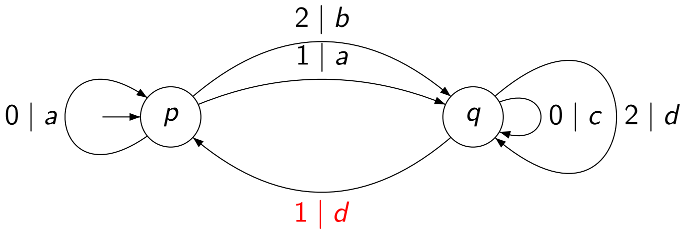


Coding: 1021 $\mapsto$ *acdd*.
Decoding: *acdd* $\mapsto$ 102.(1 or 2).

Properties of the transducer

- Deterministic on its input.
- Deterministic on its output with delay $d = 1$.

A coding realized by a transducer with delay 1



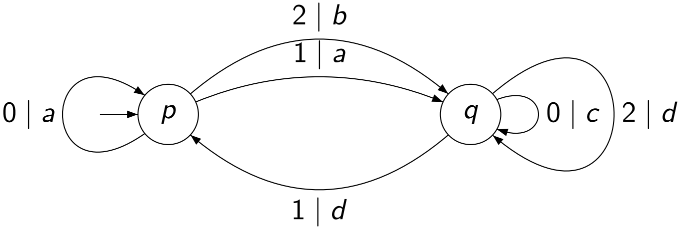
Coding: 1021 $\mapsto acdd$.

Decoding: $acdd \mapsto$ 102.(1 or 2).

Properties of the transducer

- Deterministic on its input.
- Deterministic on its output with delay $d = 1$.

A coding realized by a transducer with delay 1



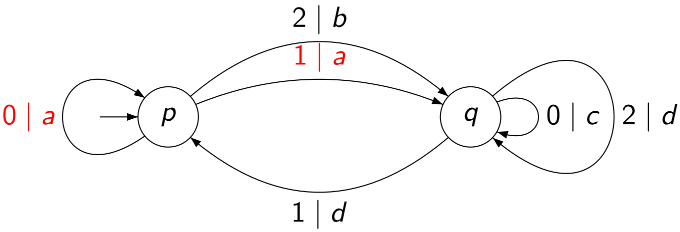
Coding: $1021 \mapsto acdd$.

Decoding: $acdd \mapsto 102.(1 \text{ or } 2)$.

Properties of the transducer

- Deterministic on its input.
- Deterministic on its output with delay $d = 1$.

A coding realized by a transducer with delay 1



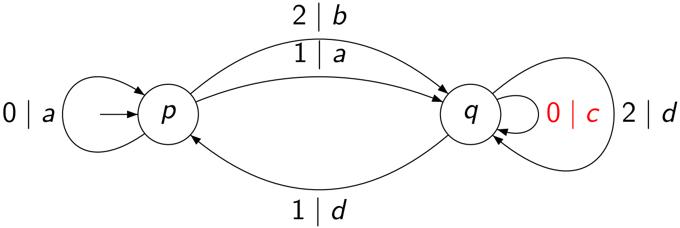
Coding: $1021 \mapsto acdd$.

Decoding: $acdd \mapsto 102.(1 \text{ or } 2)$.

Properties of the transducer

- Deterministic on its input.
- Deterministic on its output with delay $d = 1$.

A coding realized by a transducer with delay 1



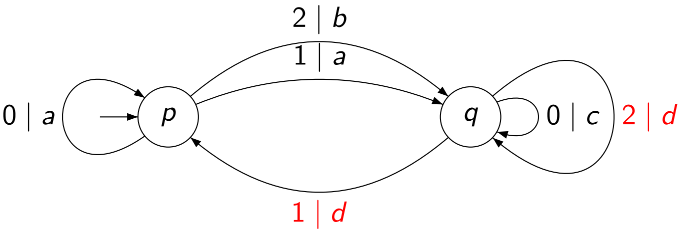
Coding: 1021 $\mapsto$ $acdd$.

Decoding: $acdd \mapsto$ 102.(1 or 2).

Properties of the transducer

- Deterministic on its input.
- Deterministic on its output with delay $d = 1$.

A coding realized by a transducer with delay 1



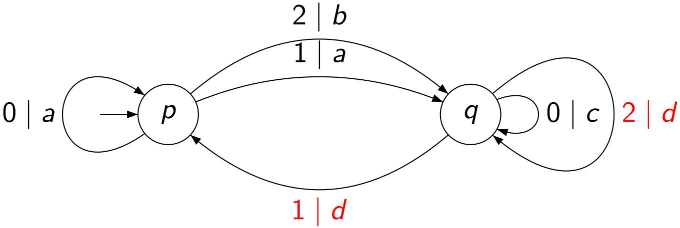
Coding: $1021 \mapsto acdd$.

Decoding: $acdd \mapsto 102.(1 \text{ or } 2)$.

Properties of the transducer

- Deterministic on its input.
- Deterministic on its output with delay $d = 1$.

A coding realized by a transducer with delay 1



Coding: 1021↦*acdd*.

Decoding: *acdd*↦ 102.(1 or 2).

Properties of the transducer

- Deterministic on its input.
- Deterministic on its output with delay $d = 1$.

Finite state coding theorem

Proposition

Let S and C be factorial and extensible languages. If an (S, C)-encoding with rate α exists, then (II) holds.

Information Inequality

$$\alpha \mathcal{H}(S) \leq \mathcal{H}(C), \qquad \text{(II)}$$

Theorem

If S and C are sofic[a] and strong (II) holds, then there exists an (S, C)-encoding realized by a finite-state transducer.

[a] regular+…

The optimal rate…

…is $\alpha \leq \frac{\mathcal{H}(C)}{\mathcal{H}(S)}$.

Problem I: timed source, discrete channel, approximate transmission

Usually timed words are stored in text files.

Subtitle file: SubRip .srt file example (Wikipedia)

```
1
00:00:20,000 --> 00:00:24,400
Altocumulus clouds occur between six thousand

2
00:00:24,600 --> 00:00:27,800
and twenty thousand feet above ground level.
```

What is the optimal encoding for that type of data?

Problem I: timed source S, discrete channel C

Definition (Encoding $\phi : S \to C$: precision ε, rate α, delay d)

- it is of rate α, i.e. $\alpha = \frac{|w|}{|\phi(w)|}$;
- "injective" with precision ε and delay d i.e.

$$\forall n \in \mathbb{N}, w, w' \in A^n : \ \phi(w) = \phi(w') \Rightarrow \mathtt{dist}(w, w') < \varepsilon.$$

if $|w| = |w'|$, $|u| = |u'| = d$ and $\phi(wu) = \phi(w'u')$ then $\mathtt{dist}(w, w') < \varepsilon$.

Example

$S = ([0, 1] \times \{a, b\})^*$, $C = (ASCII)^*$.
Encoding: truncation to 2 digits.

$$(1/3, a)(0.338, a)(ln(2), b) \mapsto 33a33a69b.$$

Rate $\alpha = 1/3$, delay $d = 0$, precision $\varepsilon = 0.01$.

Theorem for Problem I (timed source, discrete channel)

Information Inequality

$$\alpha(\mathcal{H}(S) + \log_2(1/\varepsilon)) \leq \mathcal{H}(C) \qquad \text{(II)}$$

Proposition

If an encoding with rate α and precision ε exists then (II) holds

Theorem

For regular languages S (timed) and C (untimed), if some strong version of (II) holds then an S-C encoding can be realized by a real-time transducer.

Problem II: timed source, timed channel, exact transmission, rate 1

Definition (Encoding $\phi : S \to C$ with delay d)

- it is length preserving (rate 1): $|\phi(w)| = |w|$,
- it is almost injective (with delay d),
- no time scaling.

Theorem for Problem II (timed source, timed channel)

Information Inequality

$$\mathcal{H}(S) \leq \mathcal{H}(C). \tag{II}$$

Proposition

If an encoding exists then (II) holds.

Theorem

If strong (II) holds then an encoding from S to C can be realized by a real time transducer.

A failure: timed source, timed channel, exact transmission, rate $\neq 1$

Definition (Encoding $\phi : S \to C$ with delay d and rate α)

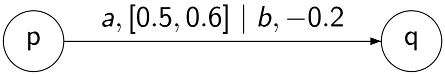
- it is of rate α, i.e. $\alpha = \frac{|w|}{|\phi(w)|}$;
- it is almost injective (with delay d),
- no time scaling.

The results: whatever the entropies of $\mathcal{H}(S)$, $\mathcal{H}(C)$

- If $\alpha > 1$ then no coding exists.
- If $\alpha < 1$ then there is always a coding

Sketch of construction of the real time transducers

A transition of a real time transducer:

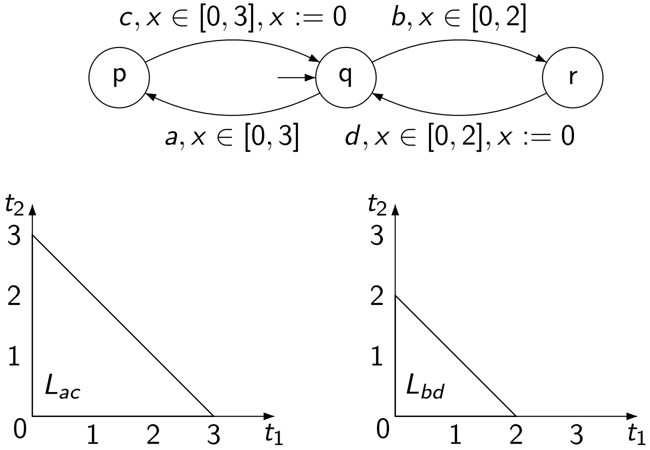


(clock $x \in [0.5, 0.6]$; output $x - 0.2$)
Example: $(a, 0.54321\textit{etc.}) \mapsto (a, 0.34321\textit{etc.})$

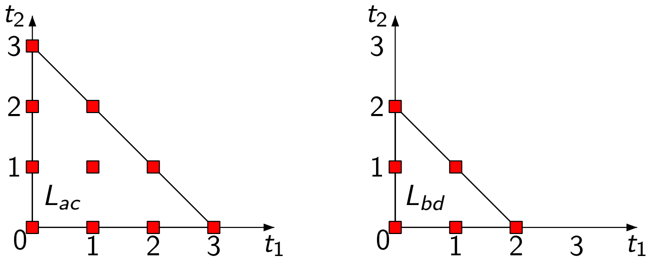
Properties of the real-time transducer

- Real time = one clock always reset (very simple timed automaton/transducer).
- Guards multiple of a fixed discretization step $\varepsilon = 0.1$.
- Exact transmission, no approximation (same *etc.*).

Discretization and real-time approximation

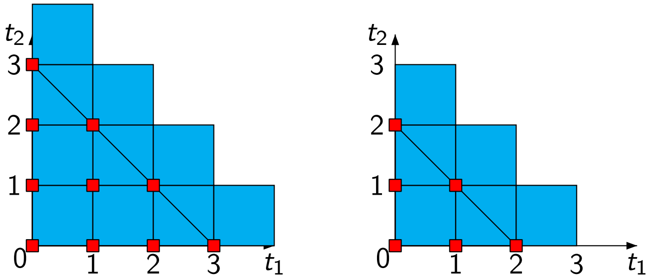


Discretization and real-time approximation



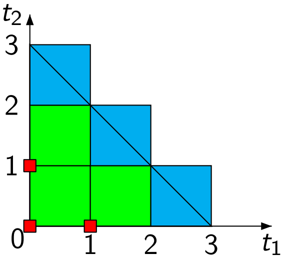
Discretisation L_ε with $\varepsilon = 1$

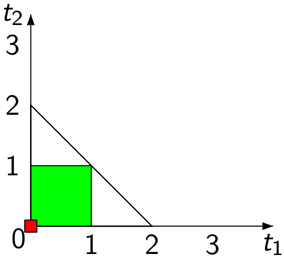
Discretization and real-time approximation



Discretisation L_ε^+ with $\varepsilon = 1$ Over-approximation $L \subseteq \mathcal{B}_\varepsilon^{NE}(L_\varepsilon^+)$

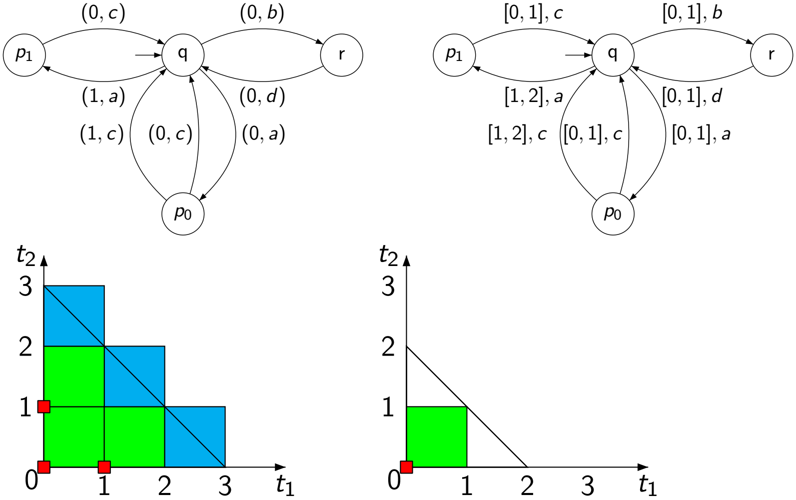
Discretization and real-time approximation





Discretisation L_ε^- with $\varepsilon = 1$ Under-approximation $\mathcal{B}_\varepsilon^{NE}(L_\varepsilon^-) \subseteq L$

Realised by DFA and real-time automaton



Discretisation L^-_ε with $\varepsilon = 1$ Under-approximation $\mathcal{B}^{NE}_\varepsilon(L^-_\varepsilon) \subseteq L$

Reduction to the discrete case

3-step reduction scheme

1. discretize the timed languages S, C with a sampling rate ε to obtain $S_\varepsilon^+, C_\varepsilon^-$; ensure II: $h(S_\varepsilon^+) < h(C_\varepsilon^-)$
2. use classical coding theorem: build coding $S_\varepsilon^+ \to C_\varepsilon^-$;
3. go back to timed languages by taking 1 cube for each discrete points.

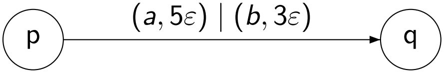
Finally :

$$S \subseteq \mathcal{B}_\varepsilon^{NE}(S_\varepsilon^+) \to \mathcal{B}_\varepsilon^{NE}(C_\varepsilon^-) \subseteq C$$

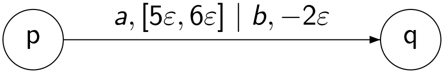
of S_ε and C_ε.

Sketch of construction of the real time transducers

- A transition of the discrete transducer between S_ε^+ and C_ε^-:



- The corresponding transition of the real time transducer:



Summary

- Definition of volume and entropy for TA
- Recurrent formula for volume $\Longrightarrow$ computable
- A symbolic algorithm to compute $\mathcal{H}$ for $1\frac{1}{2}$ clocks
- 2 algorithms to approximate $\mathcal{H}$: using operators or discretization
- Links to other entropies (discretization) and information theory (Kolmogorov complexity, timed coding).

Other applications

Mostly N. Basset's works:

- Eigenvectors of operator Ψ can be used to add "natural"[1] probabilities to timed automata (generalization of Shannon-Parry measure)
  $\rightarrow$ quasi-uniform statistical model checking.
- Computing volumes is linked to counting permutations of a certain kind.

[1] i.e. maximal entropy

Future work

- Entropy/unit of time (actually ongoing work)
- Efficient algorithms (zone based, ...)
- More applications.
- Extensions (hybrid automata, ...)

Relevant publications

This talk is based on:

- Main source: [ABD15] E. Asarin, N. Basset, A. Degorre. Entropy of regular timed languages. Information and Computation 241, 2015.
- Discretization aspects: [AD'09] E. Asarin, A. Degorre. Volume and entropy of regular timed languages: Discretization Approach. Concur'09.
- Channel coding: [ABBDP'12] E. Asarin, N. Basset, M.-P. Béal, A. Degorre, D. Perrin. Toward a Timed Theory of Channel Coding. Formats'12.

Not presented here

Various directions explored by us:

- E. Asarin, A. Degorre. Two Size Measures for Timed Languages. FSTTCS'10.
- E. Asarin, N. Basset, A. Degorre. Generating Functions of Timed Languages Generating functions. MFCS'12.
- N. Basset. Maximal entropy timed stochastic process. ICALP'13.
- N. Basset. Counting and Generating Permutations Using Timed Languages. LATIN'14.

Thank you!

Questions?

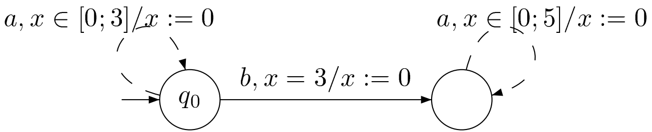
Punctual guards should be fine!

- This time we do not accept 0 (or $-\infty$) as a meaningful answer for the size of a degenerated automaton.
- However we want to keep punctual guards.

What can we do?

- Remark 1: the operator Ψ will always yield volume 0 for degenerated runs.
- Remark 2: discretization approach gives non-zero answers, but how to interpret it in an example such as (next slide), where it adds up meters to square meters ?

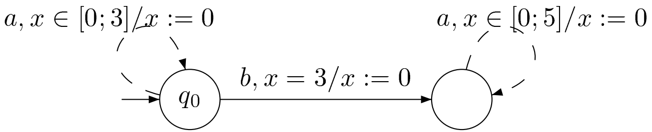
A bothering example



Left or right?

- a^*, set $[0,3]^n$, volume 3^n, entropy $\log 3$ (i.e. 3 sec/symbol)
- ba^*, set $3 \times [0,5]^n$, volume 0, entropy $-\infty$ (but 5 sec/symbol)
- Something is wrong.

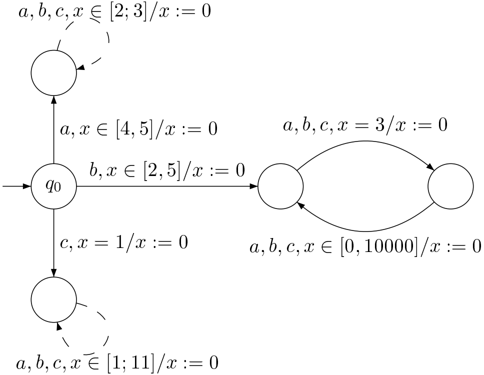
A bothering example



Left or right?

- a^*, set $[0,3]^n$, dimension n, n-volume 3^n
- ba^*, set $3 \times [0,5]^n$, dimension $n-1$, $(n-1)$-volume 3^n
- Who does win?

Another embarassing example



a, b or c?

- $a\Sigma^*$, dimension n, volume 3^{n-1};
- $b\Sigma^*$, dimension $(n + 1/2)$, volume $300^{n-1} \cdot 3$;
- $c\Sigma^*$, dimension $n - 1$, volume 30^{n-1};
- Choose your champion.

Key to solution

Information measure: inspired by Kolmogorov-Tikhomirov ε-entropy.

- $L_n \to$ set of disjoint timing polyhedra
- metric for spaces of every dimension
- Size = cardinality of the ε-net of this set $\simeq \sum_m V_m(P_n^m)\varepsilon^{-m}$

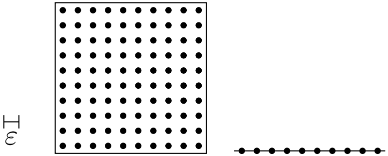


Figure: Adding meters to square meters: two polyhedra and their minimal ε-partitions.

Solution

We define the corresponding entropy:

Definition (ε-entropy)

$$\mathfrak{h}_\varepsilon(L_n) = \log \sum_m V_m(P_n^m)\varepsilon^{-m}$$

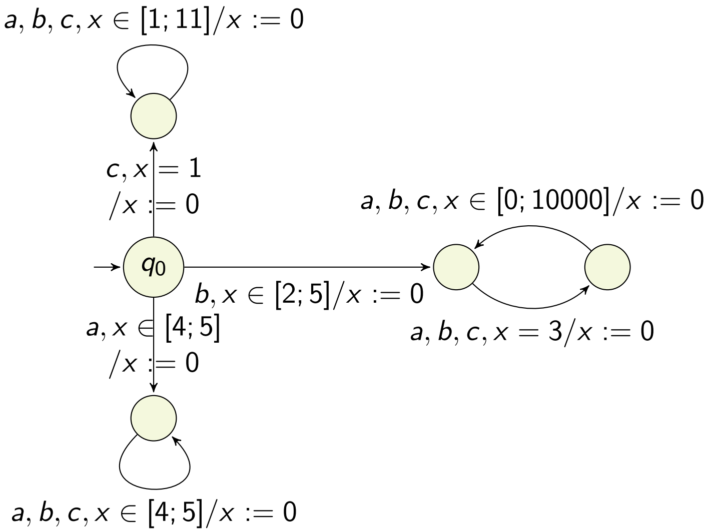
With such a definition, the following holds (for "some" $\simeq$) :

$$\mathfrak{h}_\varepsilon(L_n) \simeq n(-\alpha \log \varepsilon + \mathcal{H}_\alpha)$$

Explanation : when $n \to \infty$ and $\varepsilon \to 0$, only terms of "maximal" dimension do matter.

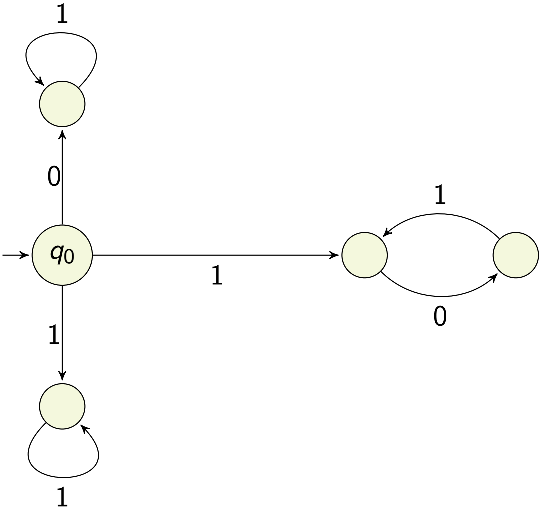
- $\alpha = \lim_{n\to\infty} \dim L_n/n$: mean dimension of L (*à la* Gromov)
- $\mathcal{H}_\alpha$: volumic entropy, i.e. logarithmic asymptotic growth of the (αn)-volume

Mean dimension



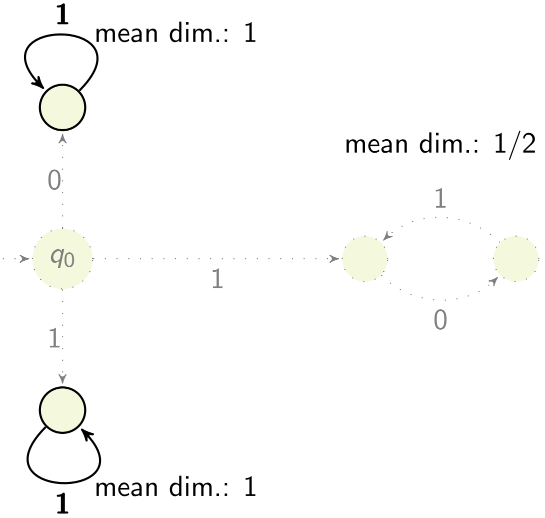
A timed automaton...

Mean dimension



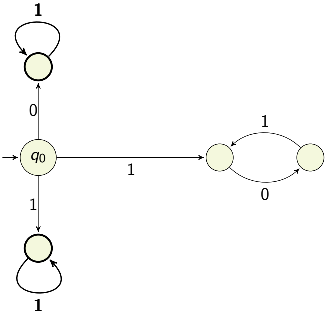
Let's keep only the dimension of guards!

Mean dimension



We find 2 critical cycles, with mean dim.$= 1$.

Mean dimension



$$\leftrightarrow \Phi = \begin{pmatrix} 0 & 1 & 1 & -\infty & -\infty \\ -\infty & \mathbf{1} & -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty & -\infty & 0 \\ -\infty & -\infty & -\infty & \mathbf{1} & -\infty \\ -\infty & -\infty & 1 & -\infty & -\infty \end{pmatrix}$$

- Max dim. $p \to_n q = (\Phi^n)_{pq}$ (in max-plus algebra)
- dim $L_n = \max_{q \in Q} (\Phi^n)_{q_0 q} = \rho(\Phi) n +$constant ($\rho$: max-plus spectral radius)

Lemma

Mean dimension of L: $\alpha = \rho(\Phi)$

Volumic entropy

What about $\mathcal{H}_{vol}$?

- $\mathcal{H}_{vol}$: volume growth of critical paths in the dimension graph

$\mathcal{H}_{vol}$ can be computed using similar techniques as $\mathcal{H}$ before (full-dimension entropy), restricting the operator Ψ to critical components of the automaton.

Related topics

- Volume generating functions: allow manipulating heterogenous n-volumes in the same operator $\rightarrow$ generalization of symbolic method to a larger class of automata.
- Entropy rate with respect to time: volumes of different dimension naturally appear for a same total duration. How do we sum them? (ongoing work)