Measuring Information in Timed Languages

Eugene Asarin

LIAFA — Univesité Paris Diderot and CNRS

March 6, 2012 — LATA

Credits

People

EA, Aldric Degorre, Dominique Perrin, Nicolas Basset, and Romain Aïssat

Papers

- CONCUR'09
- FORMATS'09
- Basset's Master Thesis
- FSTTCS'10
- FORMATS'11
- a submission to a conference
- but no journal paper yet

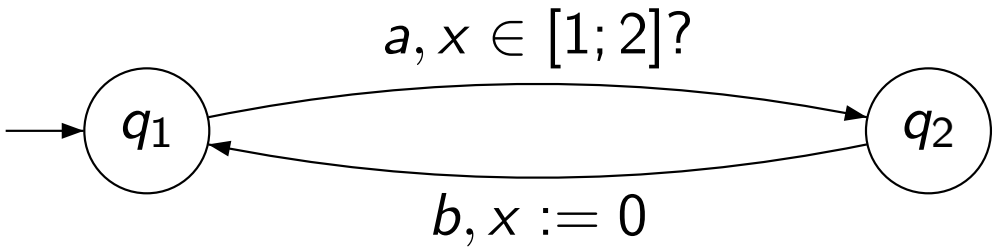
Part I

The context

Outline

Reminder: Timed Languages and Timed Automata [Alur and Dill]

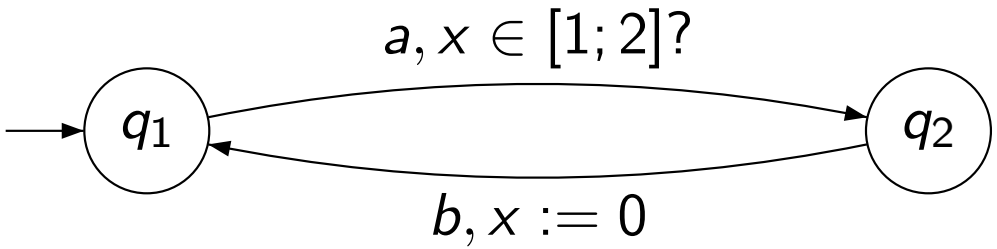
- Timed automaton (we forget to write $\dot{x} = 1$):



- Anatomy: locations, clocks, transitions, labels, guards, resets
- States: $(q, \mathbf{x})$ (infinite-state)

Reminder: Timed Languages and Timed Automata [Alur and Dill]

- Timed automaton (we forget to write $\dot{x} = 1$):

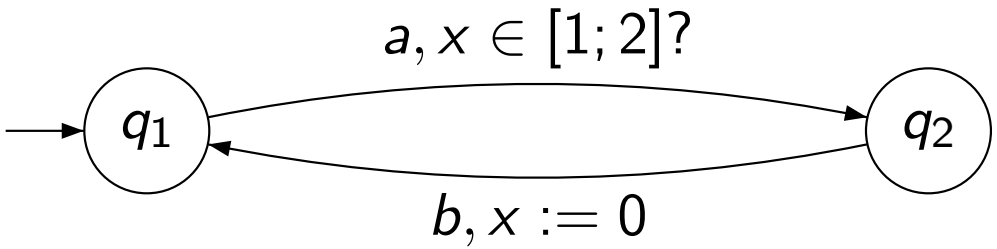


- Its run

$$(q_1, 0) \xrightarrow{1.83} (q_1, 1.83) \xrightarrow{a} (q_2, 1.83) \xrightarrow{4.1} (q_2, 5.93) \xrightarrow{b} (q_1, 0) \xrightarrow{1} (q_1, 1) \xrightarrow{a} (q_2, 1)$$

Reminder: Timed Languages and Timed Automata [Alur and Dill]

- Timed automaton (we forget to write $\dot{x} = 1$):



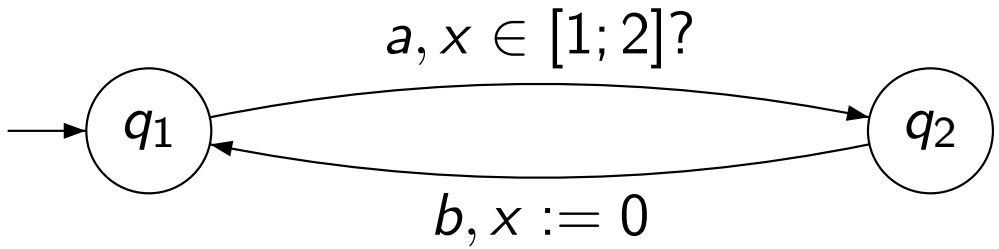
- Its run

$$(q_1, 0) \xrightarrow{1.83} (q_1, 1.83) \xrightarrow{a} (q_2, 1.83) \xrightarrow{4.1} (q_2, 5.93) \xrightarrow{b} (q_1, 0) \xrightarrow{1} (q_1, 1) \xrightarrow{a} (q_2, 1)$$

- Its *trace* $1.83\, a\, 4.1\, b\, 1\, a \in \Sigma^3 \times \mathbb{R}^3$ *a timed word*

Reminder: Timed Languages and Timed Automata [Alur and Dill]

- Timed automaton (we forget to write $\dot{x} = 1$):



- Its run

$$(q_1, 0) \xrightarrow{1.83} (q_1, 1.83) \xrightarrow{a} (q_2, 1.83) \xrightarrow{4.1} (q_2, 5.93) \xrightarrow{b} (q_1, 0) \xrightarrow{1} (q_1, 1) \xrightarrow{a} (q_2, 1)$$

- Its *trace* $1.83\, a\, 4.1\, b\, 1\, a \in \Sigma^3 \times \mathbb{R}^3$ *a timed word*
- Its *timed language*: set of all the traces starting in q_1, ending in q_2:

$$\{t_1\, a\, s_1\, b\, t_2\, a\, s_2\, b \ldots t_n\, a \mid \forall i. t_i \in [1; 2]\}$$

Some simple exercises

Draw timed automata for specifications:

- Request a arrives every 5 minutes.

Some simple exercises

Draw timed automata for specifications:

- Request a arrives every 5 minutes.
- Request a arrives every 5 to 7 minutes.

Some simple exercises

Draw timed automata for specifications:

- Request a arrives every 5 minutes.
- Request a arrives every 5 to 7 minutes.
- a arrives every 5 to 7 minutes; and b arrives every 3 to 10 minutes.

Some simple exercises

Draw timed automata for specifications:

- Request a arrives every 5 minutes.
- Request a arrives every 5 to 7 minutes.
- a arrives every 5 to 7 minutes; and b arrives every 3 to 10 minutes.
- Request a is serviced within 2 minutes by c or rejected within 1 minute by r.

Some simple exercises

Draw timed automata for specifications:

- Request a arrives every 5 minutes.
- Request a arrives every 5 to 7 minutes.
- a arrives every 5 to 7 minutes; and b arrives every 3 to 10 minutes.
- Request a is serviced within 2 minutes by c or rejected within 1 minute by r.
- The same, but a arrives every 5 to 7 minutes.

What I said 10 years ago

From applied theory to basic theory - challenges

- Develop an algebraic theory of timed languages.
- Add topology and handle imprecision.
- Lift the classical theory: shuffle, determinizability, Kleene, logics, Myhill-Nerode, minimization, pumping
- Find new applications to timed texts: speech, music...

What I said 10 years ago

From applied theory to basic theory - challenges

- Develop an algebraic theory of timed languages.
- Add topology and handle imprecision.
- Lift the classical theory: shuffle, determinizability, Kleene, logics, Myhill-Nerode, minimization, pumping
- Find new applications to timed texts: speech, music...

But I missed one topic

Explore size and information measures of timed languages

Why to Measure Size of Timed Languages

Some reasons - speculations

- Mathematics: it is interesting, and generalizes classical notions
- Quantitative verification: size of bad behaviors
- Approximation quality for $L \supset M$
- Information content: see below
- Security: timed information flow

Reminder: Size of (Information in) Languages

Defining entropy

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

Reminder: Size of (Information in) Languages

Defining entropy

- Take a prefix-closed 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. $|w\texttt{.zip}| \approx \mathcal{H}|w|$
- Information content of a typical $w \in L$ (bits/symbol)
- Topological entropy of a subshift.

(explanation – compression rate vs $\mathcal{H}$)

Fact (Lower bound)

Some word $w \in L_n$ *yields* $|w\,.\,zip| \geq \mathcal{H}n$

Proof.

There are 2^m possible zip-files of size $< m$. If $2^m < \#L_n$ then some $w \in L_n$ has no zip of size $< m$. □

(explanation – compression rate vs $\mathcal{H}$)

Fact (Lower bound)

Some word $w \in L_n$ *yields* $|w\,.\,\texttt{zip}| \geq \mathcal{H}n$

Fact (Upper bound)

For an appropriate compression algorithm $|w\,.\,\texttt{myzip}| \leq \mathcal{H}n$

Proof.

Take $w\,.\,\texttt{myzip}$ = the ordinal number of w (in binary) in lex. order of L_n. □

(explanation – compression rate vs $\mathcal{H}$)
Kolmogorov complexity

Fact (Lower bound)

Some word $w \in L_n$ *yields* $|w\,.\,zip| \geq \mathcal{H}n$

Fact (Upper bound)

For an appropriate compression algorithm $|w\,.\,myzip| \leq \mathcal{H}n$

Corollary (Entropy = Information per symbol)

$$\max\{K(w) \mid w \in L_n\} \approx \mathcal{H}n$$

Reminder: 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 $\mathcal{H} = \log \rho$.

Reminder: Entropy of Regular Languages

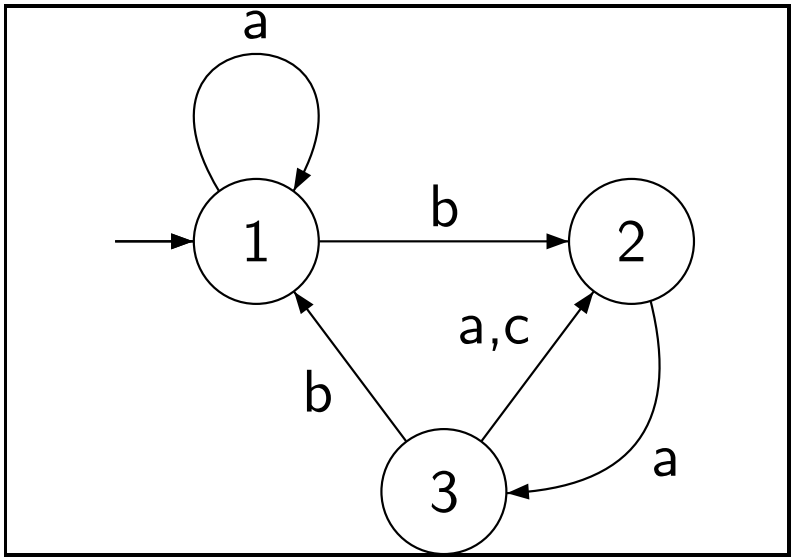
Computing $\mathcal{H}(L(A))$ for a deterministic A

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

Proof

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

Reminder: Entropy of Regular Languages - Example



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

- Languages:
  $\{\varepsilon\}; \{a, b\}; \{aa, ab, ba\}; \{aaa, aab, aba, baa, bab, bac\};$
  $\{aaaa, aaab, aaba, abaa, abab, abac, baaa, bab, baca, baba, babb\}$ ...
- Cardinalities: 1,2,3,6,11, ...
- Spectral radius: $\rho(M) \approx 1.80194$ (ugly cubic roots);
  entropy: $\mathcal{H} = \log \rho(M) \approx 0.84955$.

Part II

Quantitative theory of timed languages

Outline

Towards Main Definitions for Timed Languages

Observations about timed language

- $L_n \subset \Sigma^n \times \mathbb{R}^n$;
- impossible to count L_n;

Towards Main Definitions for Timed Languages

Observations about timed language

- $L_n \subset \Sigma^n \times \mathbb{R}^n$;
- impossible to count L_n;
- L_n is a collection of polyhedra in $\mathbb{R}^n$ (one for each $w \in \Sigma^n$);
- thus we can sum up volumes;

Towards Main Definitions for Timed Languages

Observations about timed language

- $L_n \subset \Sigma^n \times \mathbb{R}^n$;
- impossible to count L_n;
- L_n is a collection of polyhedra in $\mathbb{R}^n$ (one for each $w \in \Sigma^n$);
- thus we can sum up volumes;
- this will give reasonable size measures.

Main Definitions: Volume and Entropy

Notations

- Given $v = t_1 a_1 t_2 a_2 \dots t_n a_n$, we write $v = \mathbf{a} \otimes \mathbf{t}$
  - its *untiming* $\mathbf{a} = a_1 a_2 \dots a_n$
  - its *timing* $\mathbf{t} = t_1 t_2 \dots t_n$
- Volume of $P \in \mathbb{R}^n$ denoted by $\mathrm{Vol}(P)$.

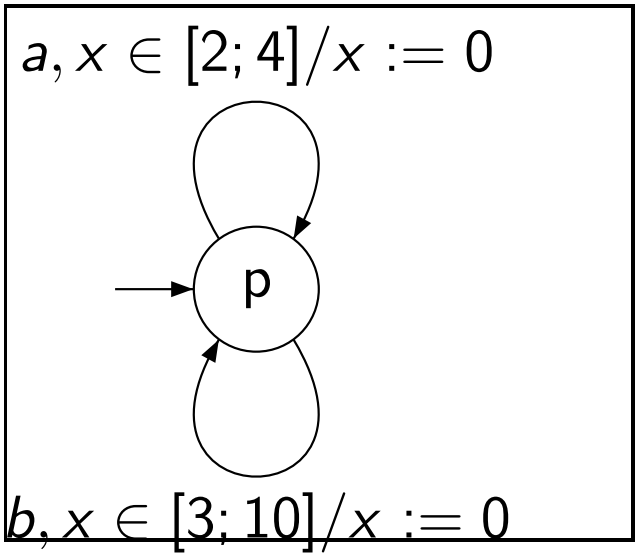
Definition (Volume and entropy of a timed language)

- Volume: sum up volumes of timings for all untimings

$$V_n(L) = \sum_{\mathbf{a} \in \Sigma^n} \mathrm{Vol}\{\mathbf{t} \mid \mathbf{a} \otimes \mathbf{t} \in L\}$$

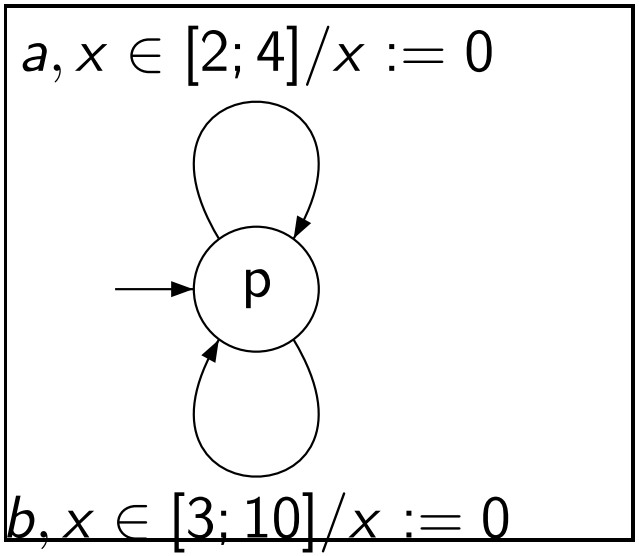
- Entropy: logarithmic growth rate of volume
  $\mathcal{H}(L) = \limsup_{n \to \infty} \frac{\log_2 V_n}{n}$.

Example 1: Volume and Entropy



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

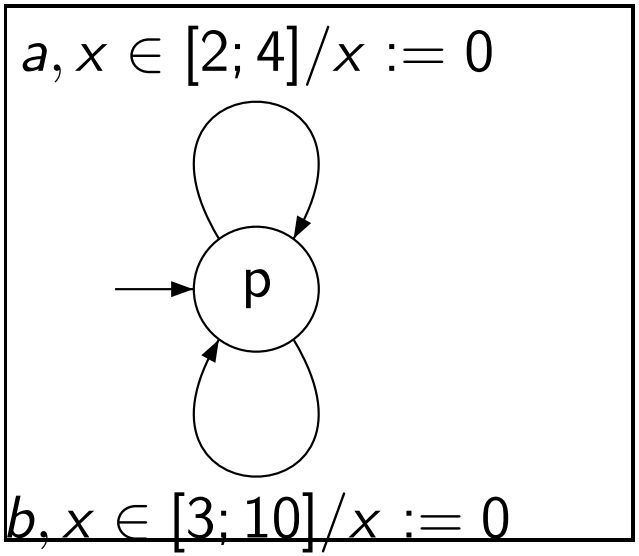
Example 1: Volume and Entropy



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

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

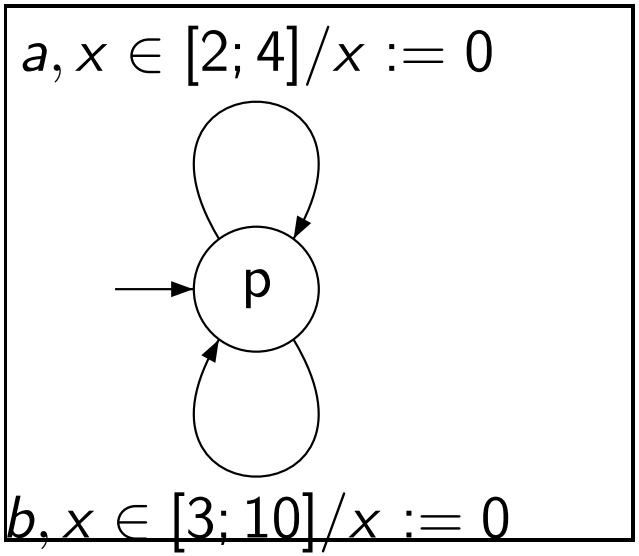
Example 1: Volume and Entropy



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

- For the untiming $bbab$ the set of timings is a 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 $w \in \{a, b\}^n$ with $a \times k; b \times (n-k)$, the set of timings is a rectangle, volume $2^k 7^{n-k}$

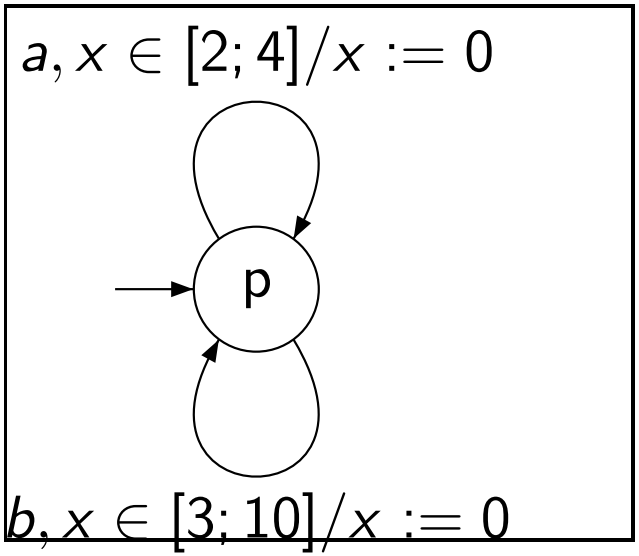
Example 1: Volume and Entropy



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

- For the untiming $bbab$ the set of timings is a 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 $w \in \{a,b\}^n$ with $a \times k; b \times (n-k)$, the set of timings is a rectangle, volume $2^k 7^{n-k}$
- Volume: $V_n(L_1) = \sum_{k=0}^{n} C_n^k 2^k 7^{n-k} = (2+7)^n = 9^n$,

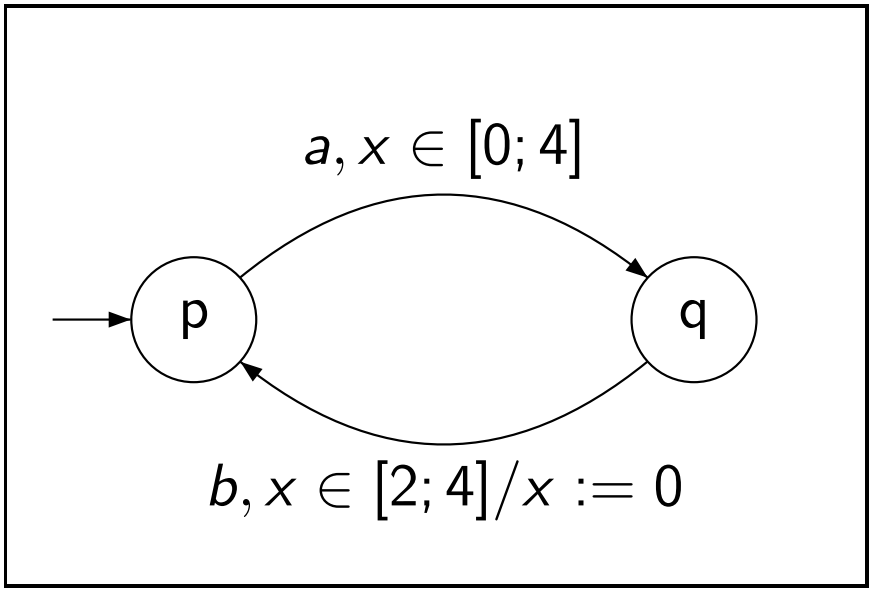
Example 1: Volume and Entropy



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

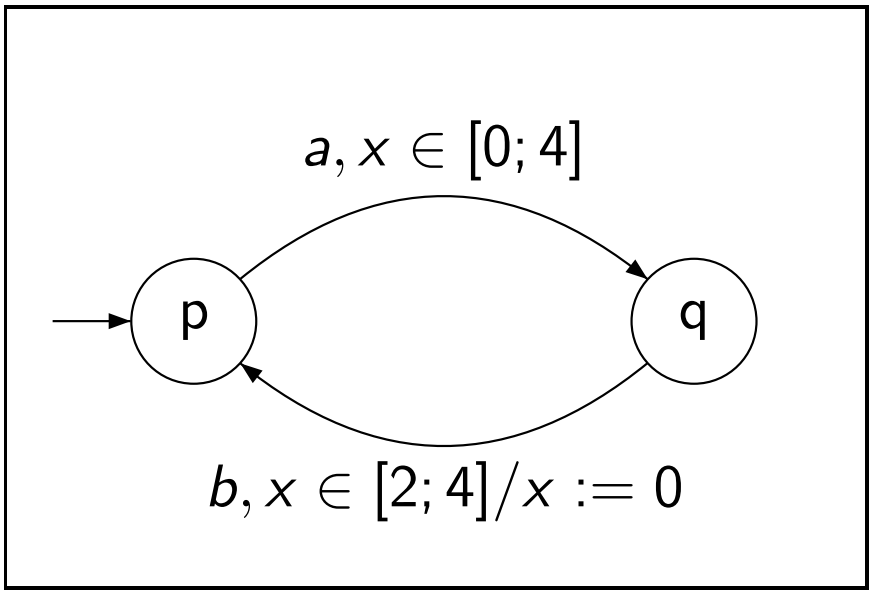
- For the untiming $bbab$ the set of timings is a rectangle $[3;10] \times [3;10] \times [2;4] \times [3;10]$, its volume $7 \cdot 7 \cdot 2 \cdot 7 = 686$
- For an untiming $w \in \{a,b\}^n$ with $a \times k; b \times (n-k)$, the set of timings is a rectangle, volume $2^k 7^{n-k}$
- Volume: $V_n(L_1)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

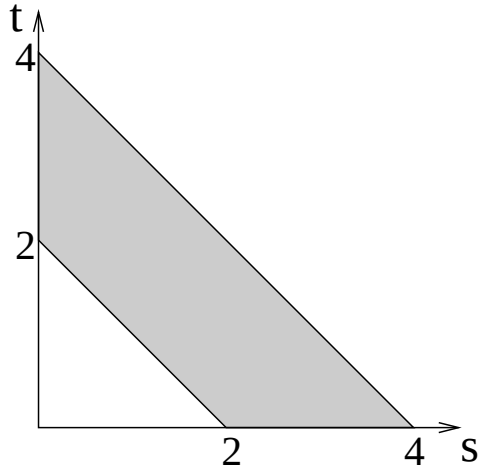
Example 2: Volume and Entropy



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

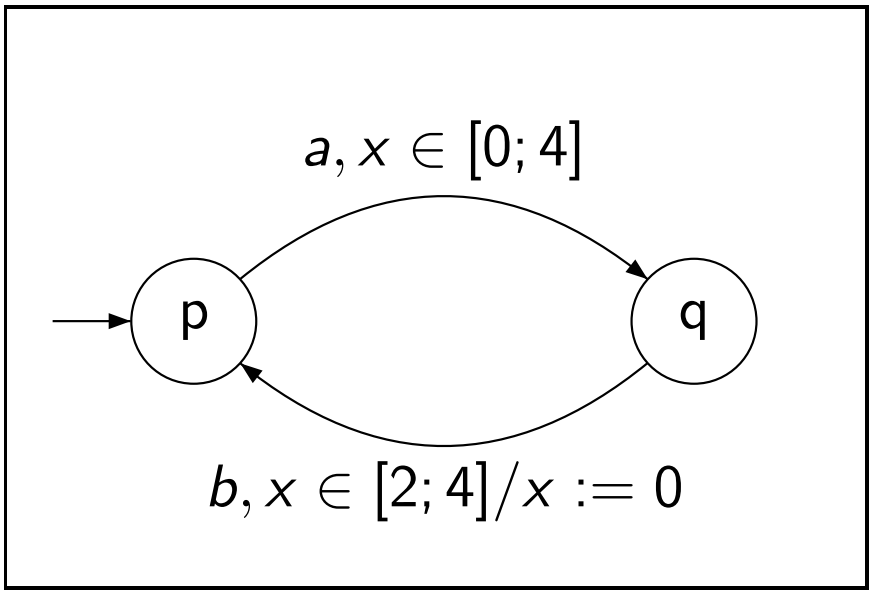
Example 2: Volume and Entropy

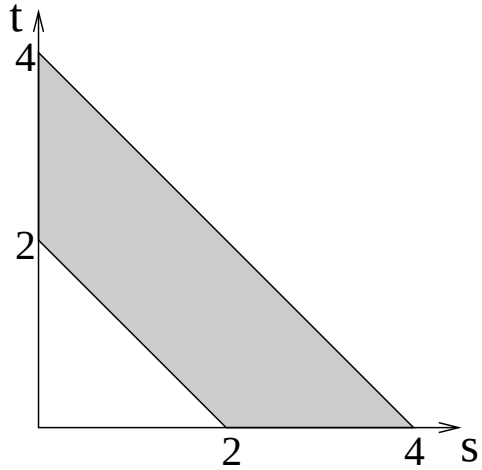




- Language: $t_1 a s_1 b t_2 a s_2 b \ldots t_k a s_k b$ such that $2 \leq t_i + s_i \leq 4$
- For the only n-untiming $w = (ab)^{n/2}$ the set of timings is a product of $n/2$ trapezia.
- Volume: $V_n(L_2) = 6^{n/2}$,

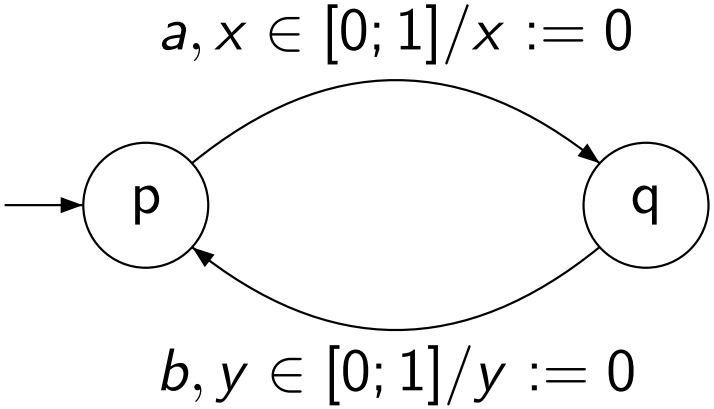
Example 2: Volume and Entropy





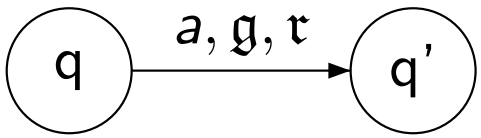
- Language: $t_1 a s_1 b t_2 a s_2 b \ldots t_k a s_k b$ such that $2 \leq t_i + s_i \leq 4$
- For the only n-untiming $w = (ab)^{n/2}$ the set of timings is a product of $n/2$ trapezia.
- Volume: $V_n(L_2) = 6^{n/2}$,
- Entropy: $\mathcal{H}(L_2) = \log 6/2 \approx 1.29$.

Example 3: Volume and Entropy



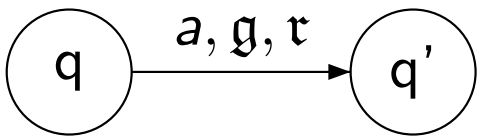
- Language: $t_1 a t_2 b t_3 a t_4 b \ldots$ such that $t_i + t_{i+1} \in [0;1]$
- For the only n-untiming $w = (ab)^{n/2}$ the set of timings is a strange polyhedron.
- Volume: see below
- Entropy: see below

Recurrence for Languages and Volumes



Let $(q, x) \stackrel{\tau a}{\to} (q', x')$
then u recognized from $(q', x') \Rightarrow$
$\tau a u$ recognized from $(q, \mathbf{x})$.

Recurrence for Languages and Volumes



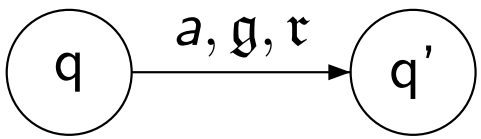
Let $(q,x) \stackrel{\tau a}{\to} (q',x')$
then u recognized from $(q',x') \Rightarrow$
$\tau a u$ recognized from $(q,\mathbf{x})$.

We introduce the languages $L_n(q,\mathbf{x})$ (where $L_n = L_n(q_0,0)$):

Language recurrence

$$\begin{aligned} L_0(q,\mathbf{x}) &= \varepsilon; \\ L_{k+1}(q,\mathbf{x}) &= \bigcup_{(q,x)\stackrel{\tau a}{\to}(q',x')} \tau a L_k(q',x'). \end{aligned}$$

Recurrence for Languages and Volumes



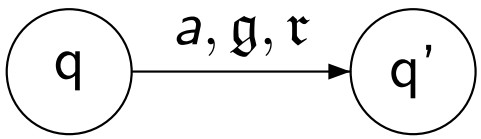
Let $(q, x) \stackrel{\tau a}{\to} (q', x')$
then u recognized from $(q', x') \Rightarrow$
$\tau a u$ recognized from $(q, \mathbf{x})$.

We introduce the languages $L_n(q, \mathbf{x})$ (where $L_n = L_n(q_0, 0)$):

Language recurrence (detailed)

$$
\begin{aligned}
L_0(q, \mathbf{x}) &= \varepsilon; \\
L_{k+1}(q, \mathbf{x}) &= \bigcup_{(q,a,\mathfrak{g},\mathfrak{r},q') \in \Delta} \; \bigcup_{\tau : \mathbf{x}+\tau \in \mathfrak{g}} \tau a L_k(q', \mathfrak{r}(\mathbf{x}+\tau)).
\end{aligned}
$$

Recurrence for Languages and Volumes



Let $(q, x) \xrightarrow{\tau a} (q', x')$
then u recognized from $(q', x') \Rightarrow$
$\tau a u$ recognized from $(q, \mathbf{x})$.

We deduce the n-volume $V_n = v_n(q_0, 0)$, where:

Volume recurrence

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

First Theorem

Theorem (Volume is computable)

V_n is a rational number. It 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

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

First Theorem

Theorem (Volume is computable)

V_n is a rational number. It 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

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

We have a beautiful formula for this sequence

but the slide is too small (©Fermat).

Reconsidering the Recurrence for Volumes

Recall the 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

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

with Ψ a positive linear operator on ...

The Banach Space of a Timed Automaton

Defining a space where v_k lives, and Ψ acts.

First attempt

$C(Q \times [0; M]^n)$: continuous real-valued functions of $(q, \mathbf{x})$
(functions on the state space of A)

The Banach Space of a Timed Automaton

Defining a space where v_k lives, and Ψ acts.

First attempt

$C(Q \times [0; M]^n)$: continuous real-valued functions of $(q, \mathbf{x})$
(functions on the state space of A)

A better one

Let $S = \{q\} \times r_q$ with r_q the entry region of q. $\mathcal{F} = C(S)$
(functions on the useful part of the state space of A)

The Operator and its Properties

The Operator

We know that

- $v_k = \Psi^k 1$
- Where $\Psi : \mathcal{F} \to \mathcal{F}$
- $\Psi f(q, \mathbf{x}) = \sum_{(q,a,\mathfrak{g},\mathfrak{r},q') \in \Delta} \int_{\tau : \mathbf{x}+\tau \in \mathfrak{g}} f(q', \mathfrak{r}(\mathbf{x}+\tau))\, d\tau$

Properties of Ψ

- A linear bounded positive operator (easy).
- Ψ^k is compact (if every clock is reset every k steps).
- Perron-Frobenius theory applies.

The Main Quantitative Theorem

An explanation

- We know that $v_k = \Psi^k 1$
- Perron-Frobenius says: $\Psi^k v \sim \rho^k v^*$, where ρ a spectral radius of Ψ, and v^* its eigenvector
- $\rho(\Psi)$ has some nice properties.

The Main Quantitative Theorem

An explanation

- We know that $v_k = \Psi^k 1$
- Perron-Frobenius says: $\Psi^k v \sim \rho^k v^*$, where ρ a spectral radius of Ψ, and v^* its eigenvector
- $\rho(\Psi)$ has some nice properties.

Theorem (Entropy Theorem)

For a deterministic (non-Zeno) timed automaton A the entropy of its language:

$$\mathcal{H} = \log \rho(\Psi)$$

Are We Done?

Yes – we have a characterization of the entropy.

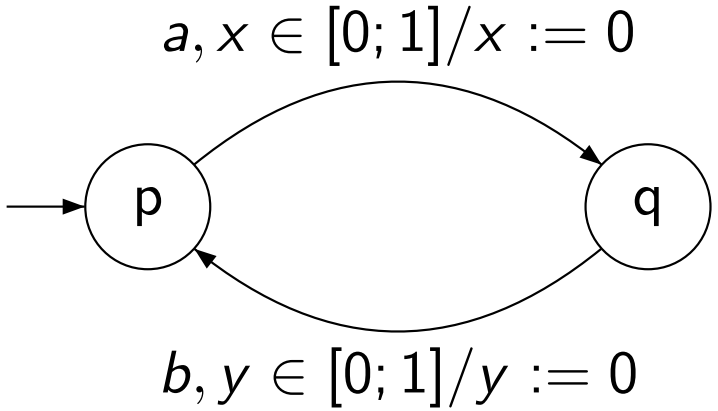
Are We Done?

Yes – we have a characterization of the entropy.
No – it is rather implicit: we need the maximal λ s.t.

$$\Psi f = \lambda f$$

has a nontrivial solution.
An awful integral equation ...
How to get a number?

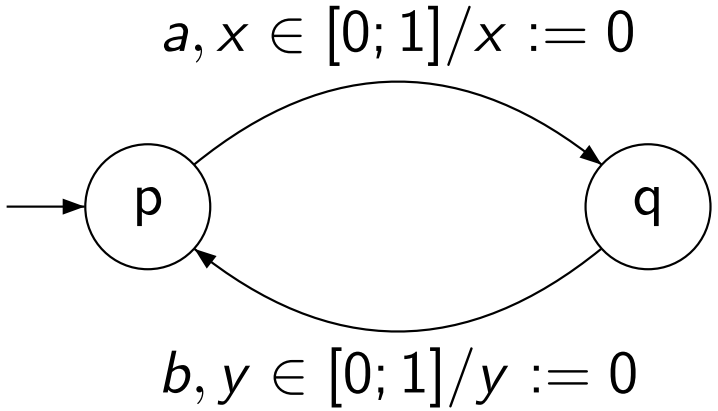
The Easy Case: $1\frac{1}{2}$ Clocks



Definition

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

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

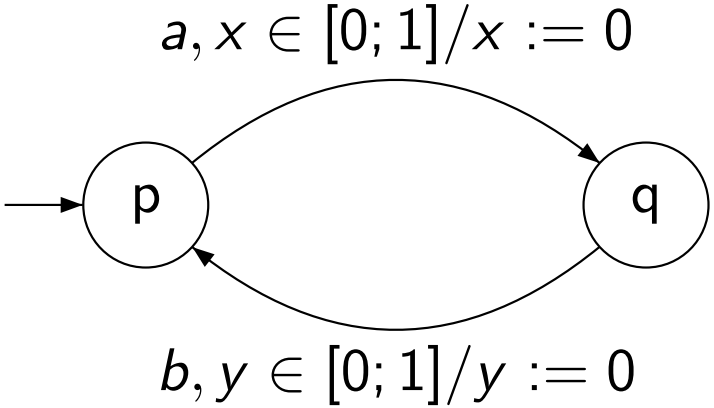


Definition

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

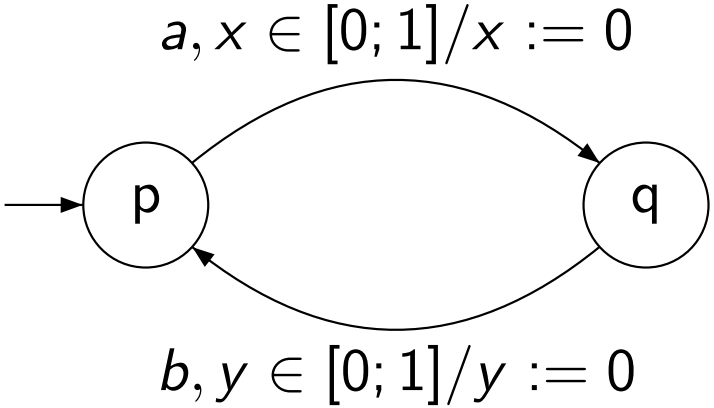
Then $v(q, x)$ has 1-dim argument $\Rightarrow$ all is easy.

Example: $1\frac{1}{2}$ Clocks



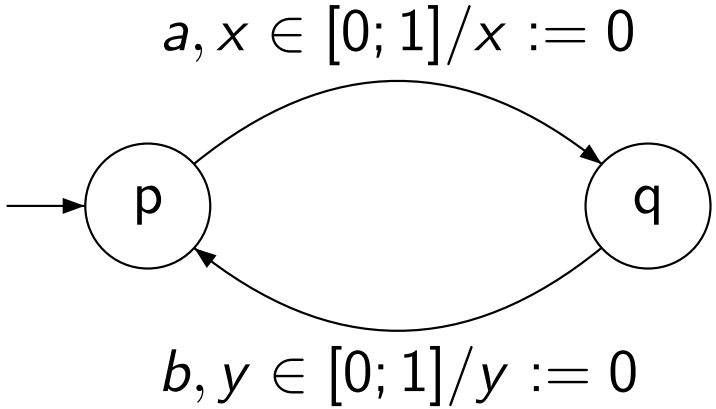
- The operator $\Psi f(x) = \int_0^{1-x} f(s)\, ds$.
- The integral equation $\lambda v(x) = \int_0^{1-x} v(t) dt$

Example: $1\frac{1}{2}$ Clocks



- The operator $\Psi f(x) = \int_0^{1-x} f(s)\, ds$.
- The integral equation $\lambda v(x) = \int_0^{1-x} v(t) dt$
- After derivation $\lambda v'(x) = -v(1-x)$. Still a problem.

Example: $1\frac{1}{2}$ Clocks



- The operator $\Psi f(x) = \int_0^{1-x} f(s)\, ds$.
- The integral equation $\lambda v(x) = \int_0^{1-x} v(t) dt$
- After derivation $\lambda v'(x) = -v(1-x)$. Still a problem.
- Once again $\lambda^2 v''(x) = -v(x)$. Also $v(1) = 0, v'(0) = 0$
- Hence $\lambda = 2/\pi; v(x) = \cos(\frac{x\pi}{2})$
- The entropy: $\log(2/\pi) \approx -0.6515$

The Procedure: $1\frac{1}{2}$ Clocks

1. Write the integral eigenvalue equation (I) with one variable.
2. Derivate (I) w.r.t. x and get a differential equation (D).
3. Instantiate (I) at 0, and obtain a boundary condition (B).
4. Solve (D) with boundary condition (B).
5. Take $\rho = \max\{\lambda |$ a non-0 solution exists$\}$.
6. Return $\mathcal{H}(L(A)) = \log \rho$.

Iteration Method for Positive Operators

Theorem (Iteration)

If $\alpha v_m \leq v_{m+1} \leq \beta v_m$, *then* $\alpha \leq \rho \leq \beta$.

Iteration Method for Positive Operators

Theorem (Iteration)

If $\alpha v_m \leq v_{m+1} \leq \beta v_m$, *then* $\alpha \leq \rho \leq \beta$.

Method

To compute entropy

- Choose m and compute v_m and v_{m+1} (piecewise polynomial functions).
- Compute $\alpha = \min(v_{m+1}/v_m)$ and $\beta = \max(v_{m+1}/v_m)$.
- Conclude that $\mathcal{H} \in [\log \alpha; \log \beta]$.

Applying to the Running Example

For L_3 the operator is:

$$\Psi f(x) = \int_0^{1-x} f(s)\, ds.$$

Applying to the Running Example

For L_3 the operator is:

$$\Psi f(x) = \int_0^{1-x} f(s)\, ds.$$

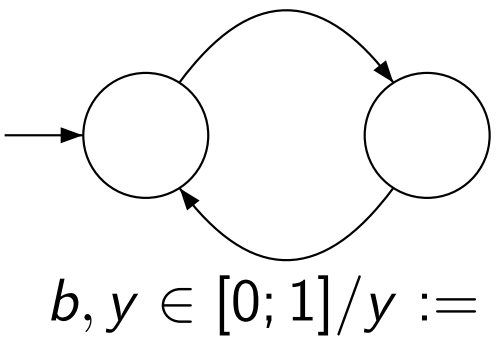
m	$v_m(x)$	α	β	$\log \alpha$	$\log \beta$
0	1	0	1		
1	$1-x$	0.5	1	-1	0
2	$1-x-1/2\,(1-x)^2$	0.5	0.667	-1	-0.584
3	$1/2\,(1-x)-1/6\,(1-x)^3$	0.625	0.667	-0.679	-0.584
4	$1/3\,(1-x)+1/24\,(1-x)^4-1/6\,(1-x)^3$	0.625	0.641	-0.679	-0.643
5	$\frac{5}{24}\,(1-x)+\frac{1}{120}\,(1-x)^5-1/12\,(1-x)^3$	0.6354	0.641	-0.6543	-0.643
6	$\frac{2}{15}\,(1-x)-\frac{1}{720}\,(1-x)^6+\frac{1}{120}\,(1-x)^5-\frac{1}{18}\,(1-x)^3$	0.6354	0.6371	-0.6543	-0.6506
7	$\frac{61}{720}\,(1-x)-\frac{1}{5040}\,(1-x)^7+\frac{1}{240}\,(1-x)^5-\frac{5}{144}\,(1-x)^3$	0.6364	0.6371	-0.6518	-0.6506

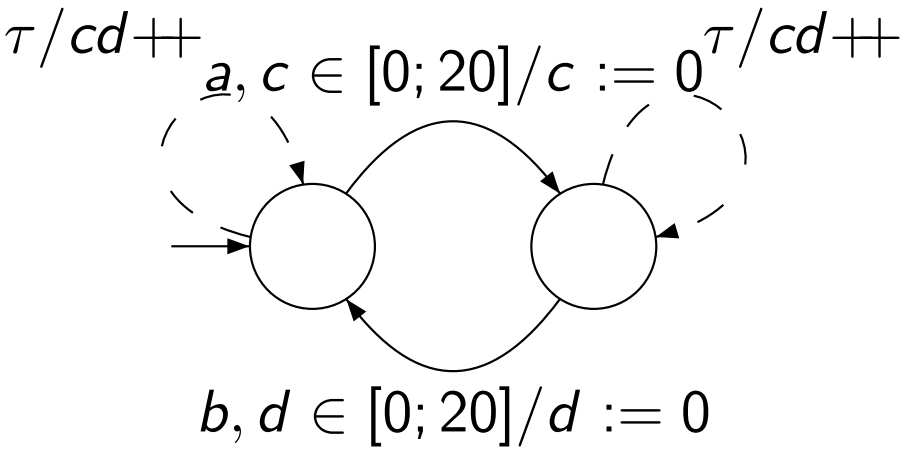
Conclusion: $-0.6518 < \mathcal{H} < -0.6506$

Discretizing Timed Automata

Brute force approach - first steps

- Take a timed automaton A. Fix $\varepsilon > 0$.
- Replace every clock x by a counter $c \approx x/\varepsilon$.
- Add to every state a $\tau, c{+}{+}$-loop (time progress ε).
- Bounded counter automaton $\rightarrow$ finite state automaton A^ε.
- A^ε accepts the discretized language L^ε over $\Sigma \cup \{\tau\}$.





Counting Words and Computing Entropy

- Let L_n^ε = all the words in L_ε with n events ($\neq \tau$).
- Then L_n^ε is an ε-mesh in L_n
- Hence $\#L_n^\varepsilon \approx \mathrm{Vol}(L_n)/\varepsilon^n$

Theorem

Computing Entropy by Discretization

$$\mathcal{H}(L) \approx \mathcal{H}(L^\varepsilon) + \log(\varepsilon)$$

with error bound $O\left(\varepsilon^{1/3}\left(\log \frac{1}{\varepsilon}\right)^{2/3}\right)$

Remark. Don't forget to eliminate τs carefully.

Kolmogorov Complexity of Timed Words and Timed Languages

Preliminary considerations

Given a timed language L

- We want to know information content of $w \in L$

Kolmogorov Complexity of Timed Words and Timed Languages

Preliminary considerations

Given a timed language L

- We want to know information content of $w \in L$
- $\min K(w)$ - too small - take a word that always takes a.
- $\max K(w) = \infty$ - take an uncomputable duration

Kolmogorov Complexity of Timed Words and Timed Languages

Preliminary considerations

Given a timed language L

- We want to know information content of $w \in L$
- $\min K(w)$ - too small - take a word that always takes a.
- $\max K(w) = \infty$ - take an uncomputable duration
- Idea: fix a precision ϵ, find maximal $\max K(w)$ up to ϵ

Kolmogorov Complexity of Timed Words and Timed Languages

Preliminary considerations

Given a timed language L

- We want to know information content of $w \in L$
- $\min K(w)$ - too small - take a word that always takes a.
- $\max K(w) = \infty$ - take an uncomputable duration
- Idea: fix a precision ϵ, find maximal $\max K(w)$ up to ϵ

Definition

Maximal information content of n-event words in L up to ϵ

$$K(L, n, \epsilon) = \max_{w \in L} \min_{d(v,w) < \epsilon} K(v)$$

Kolmogorov Complexity and Entropy

Theorem

$$K(L, n, \epsilon) \approx n(\mathcal{H} - \log \epsilon)$$

interpretation

Entropy = information per symbol.

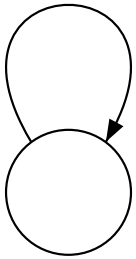
Part III

Qualitative Theory

Outline

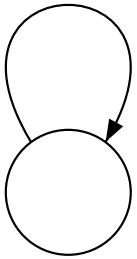
Zeno vs non-Zeno timed automata

$a, x \leq 1$



- $(t_1, a)(t_2, a) \dots (t_n, a)$ with $t_1 + \dots + t_n \leq 1$
- Arbitrary big number of events in bounded time.
- $t_n \to_{n \to +\infty} 0$
- No good discretization: No long words with delays $t_i \in \epsilon \mathbb{N}_{>0}$, $(t_1 + \dots + t_n > n\epsilon > 1)$.

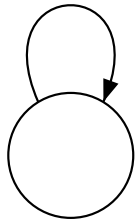
$a, x \leq 1 / x := 0$



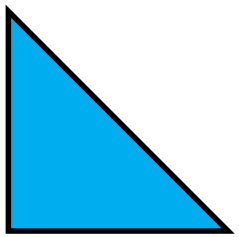
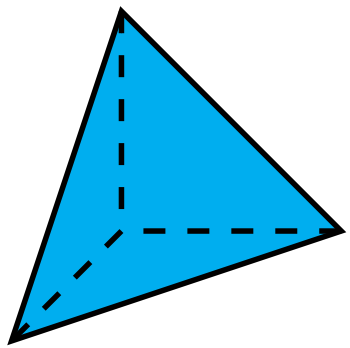
- $(t_1, a)(t_2, a) \dots (t_n, a)$ with $t_1, \dots, t_n \leq 1$.
- Duration of words are not bounded.
- Independant choice of delays $t_1, \dots, t_n$.
- Good discretization e.g. $(0.6, a)^* \subseteq L$

The simplex (Zeno and thin)

$a, x \leq 1$

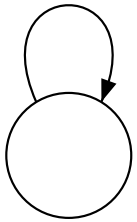


Timed word: $(t_1, a)(t_2, a) \dots (t_n, a)$

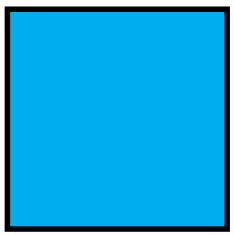
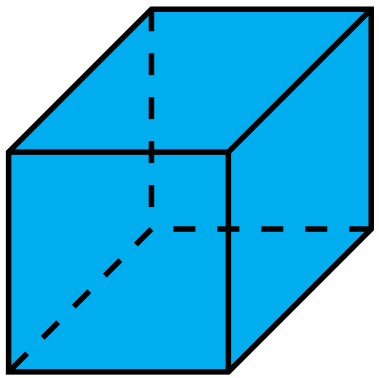
dimension 1	dimension 2	dimension 3	dimension n
			?
$t_1 \leq 1$	$t_1 + t_2 \leq 1$	$t_1 + t_2 + t_3 \leq 1$	$t_1 + \dots + t_n \leq 1$
Volume 1	Volume $\frac{1}{2}$	Volume $\frac{1}{6}$	Volume $\frac{1}{n!}$, $\mathcal{H} = -\infty$.

The hypercube (non-Zeno and thick)

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



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

dimension 1	dimension 2	dimension 3	dimension n
			?
$t_1 \leq d$	$t_1, t_2 \leq d$	$t_1, t_2, t_3 \leq d$	$t_1, \dots, t_n \leq d$
Volume d	Volume d^2	Volume d^3	Volume d^n, $\mathcal{H} = \log d$

Defining thin and thick

Recall $\mathcal{H} = \lim \frac{1}{n} \log \texttt{Vol} L_n$.

Definition

Thin and thick

- Thin languages: $\mathcal{H} = -\infty$ (i.e. $\texttt{Vol}(L_n) <$ any exponent)
- Thick languages: $\mathcal{H} > -\infty$ ($\texttt{Vol}(L_n) \approx$ an exponent)

Example

Hypercube and simplex

- The simplex is thin ($\frac{1}{n!}$ very small)
- The hypercube is thick (d^n an exponent)

Thin and thick alternative - informally

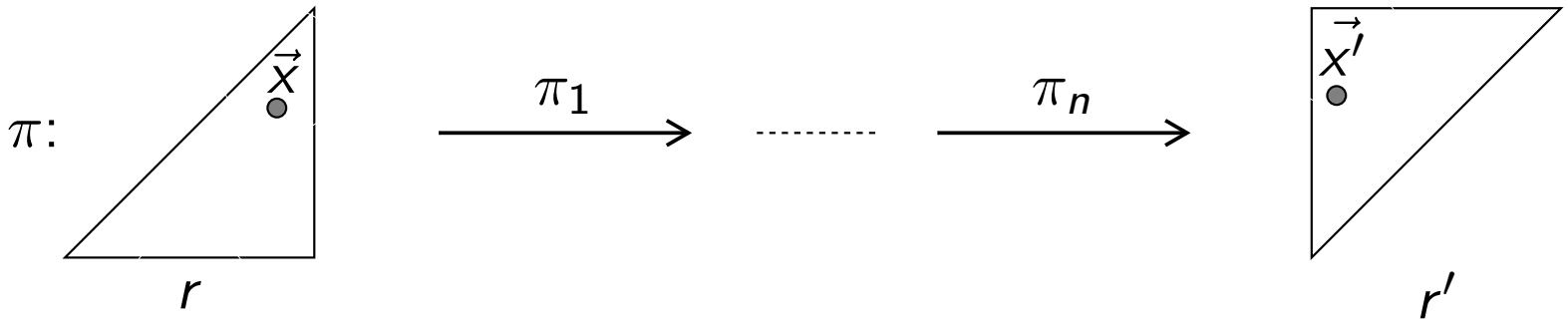
Thin

- All behaviours are "bad".
- No good discretization.
- States move toward the border of guards.

Thick

- Most of behaviours are "good".
- Forgetfulness: independence between beginning and end of a word.
- Pumping lemma.
- Good discretization.

Reachability relation and forgetfulness



Reachability relation along π

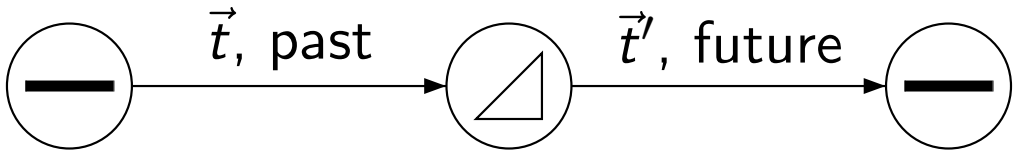
$$\texttt{Reach}(\pi) = \{(\vec{x}, \vec{x'}) \mid \exists t_1 \dots t_n,\ \vec{x} \xrightarrow{(t_1,\pi_1)(t_2,\pi_2)\dots(t_n,\pi_n)} \vec{x'}\}$$

Forgetful path

Forgetful path π: $\texttt{Reach}(\pi) = \mathbf{r} \times \mathbf{r'}$.
Ability to go from any state of $\mathbf{r}$ to any of $\mathbf{r'}$.

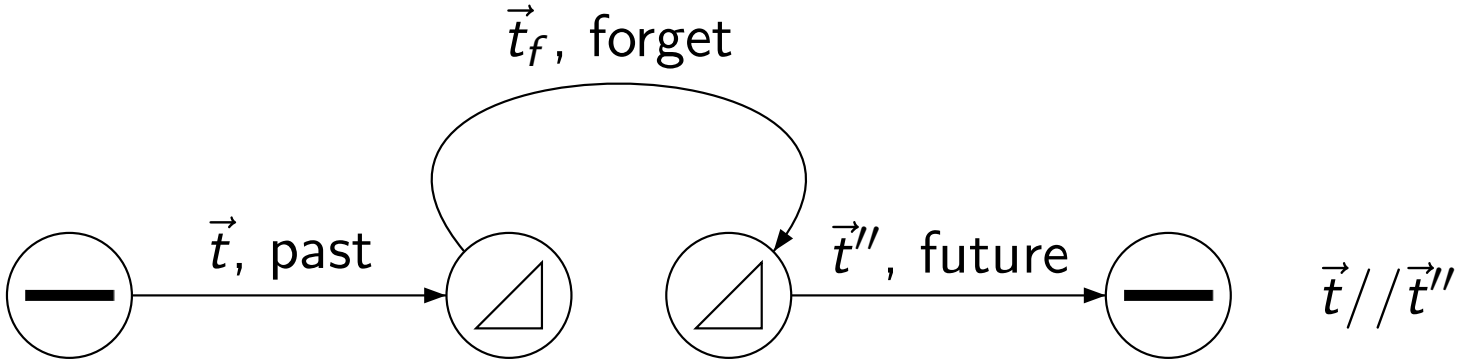
Forgetful cycle: $\mathbf{r'} = \mathbf{r}$

Why forgetful?



$\vec{t}$ and $\vec{t}'$ are dependant.
e.g. Zeno phenomenon

Forget the past to build a new future!



$\vec{t} // \vec{t}''$

Main Qualitative Theorem

Thin-Thick theorem

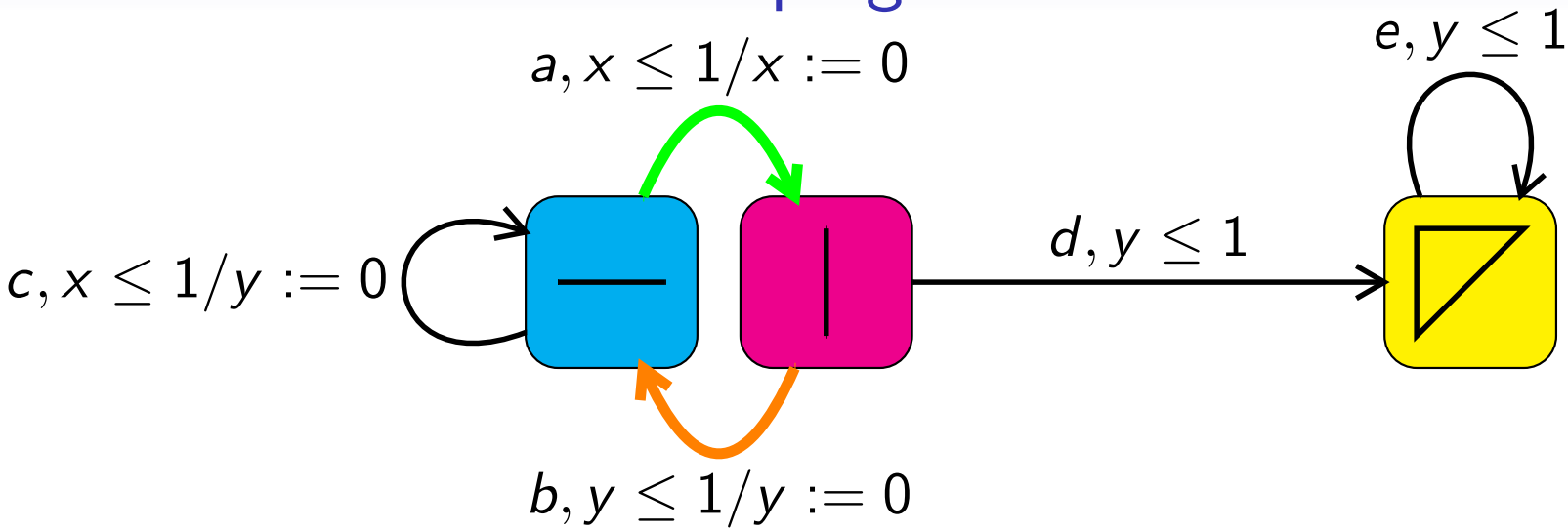
Equivalent definitions of thick languages:

1. $\mathcal{H} > -\infty$;
2. there exists a forgetful cycle;
3. there exists an ε-discrete strong limit cycle with $\varepsilon > 0$.

The thin side (less formal)

1. $\mathcal{H} = -\infty$;
2. dependance between beginning and end;
3. distance to border decreases, volume $\sim 1/n!$.

Pumping lemma



Discrete case pumping:

Every long word contains a pumpable cycle: $uv^*w \in L$

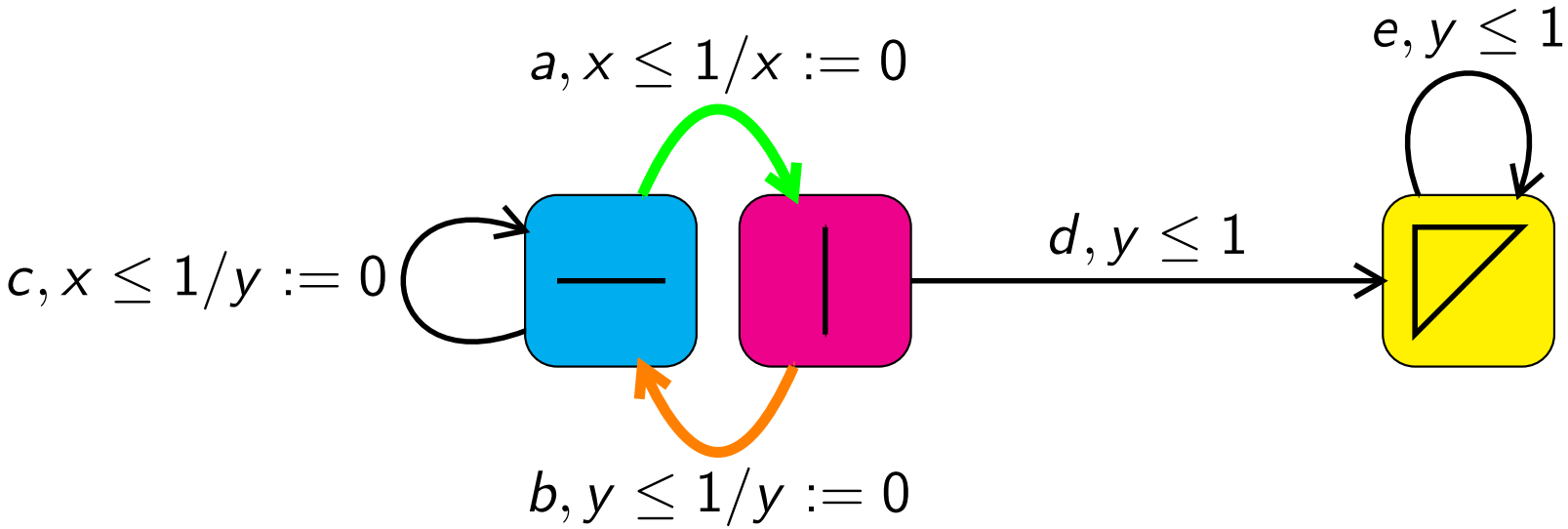
Timed case - no pumping:

Some long word contain no pumpable cycle (no fixed timing for c in bc^*cadee)

Timed thick case - no pumping:

Almost all long word contain a forgetful pumpable cycle (any timing for e.g. $bcca$ in $bccadee$).

Thick path in thick automata



thin and thick SCCs

$\mathcal{H}(\mathcal{A}) = \max \mathcal{H}(\mathcal{A}_i)$ with $\mathcal{A}_i$ the SCCs of $\mathcal{A}$

Most of events of a thick path are in thick SCCs

Every long thick path spends more than 95% time in thick SCC.

Part IV

Conclusion

Outline

Summary

- Definition of volume and entropy
- Recurrent formula for volume
- A Banach space and an operator associated with a TA
- Entropy = log of its spectral radius
- A symbolic algorithm to compute $\mathcal{H}$ for 1.5 clocks
- An iterative semi-numeric algorithm to compute $\mathcal{H}$
- A brute force discretization algorithm to compute $\mathcal{H}$ with error bounds.
- Links to Kolmogorov complexity of timed words
- $\mathcal{H} > -\infty \Rightarrow$ nice features of the automaton

Ongoing Work

We also have

- Link to dynamical systems, other entropies
- SCC decompositions
- A more precise entropy measure for punctual transitions.
- A rudimentary implementation.
- submitted: explicit formulas for volume sequences.

Future Work

We want

- An information theory for discrete/analog channels
- Error bounds/convergence proofs for operator methods
- Practical algorithmics
- Implementation
- Entropy per time unit.
- Applications