Volumetry of timed languages and applications

Nicolas Basset

LIGM, Université Paris Est & LIAFA, Université Paris Diderot.

Soutenance de thèse

Outline

1. Introduction.
2. Several contributions of the thesis.
3. Conclusion.

Part I

Introduction

The context

Introduced by Alur and Dill in 1990 to model and verify **real-time** properties of **embedded systems** (cars, phones, pacemakers...).

Since then many people works...

- on **practical issues**:
  - succesful implemented tools;
  - used in the industry.
- and also on a common **theoretical challenge**: leveraging results from automata theory to timed automata theory.

The context

Introduced by Alur and Dill in 1990 to model and verify **real-time** properties of **embedded systems** (cars, phones, pacemakers...).

Since then many people works...

- on **practical issues**:
  - succesful implemented tools;
  - used in the industry.
- and also on a common **theoretical challenge**: leveraging results from automata theory to timed automata theory.

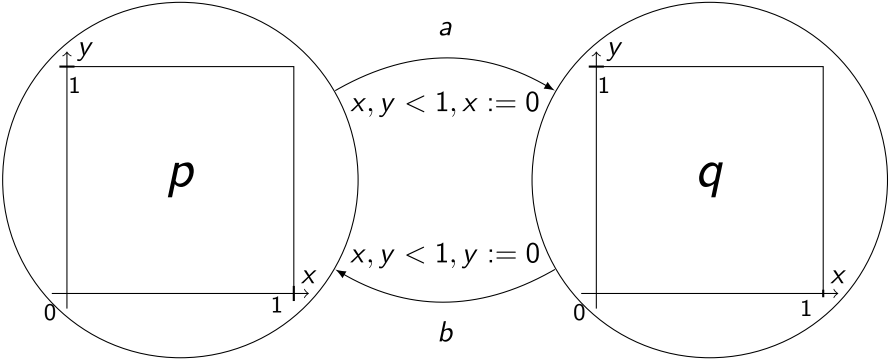
Our research program since 2009 $\subset$ theoretical challenge

Leverage quantitative properties from

- automata theory (e.g. Kleene like system of equations for generating functions);
- symbolic dynamics (e.g. entropy);
- coding theory.

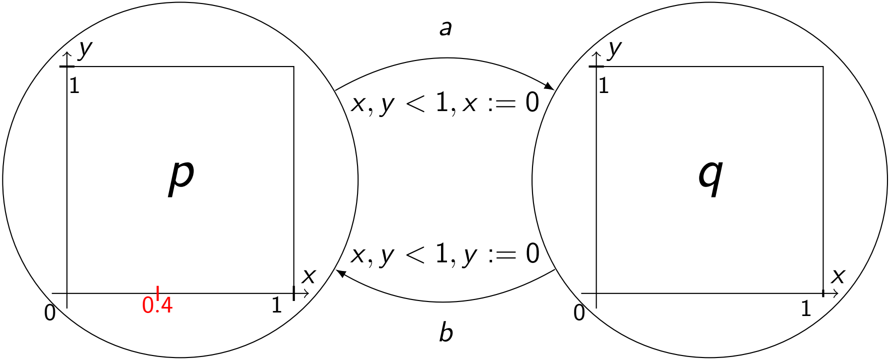
Anatomy of a Timed Automaton

- Vertices (called locations), labelled edges, clocks, guards, resets, generic state of the form $(q, \vec{x})$, intial state (e.g. $(p, (0, 0))$), Final locations



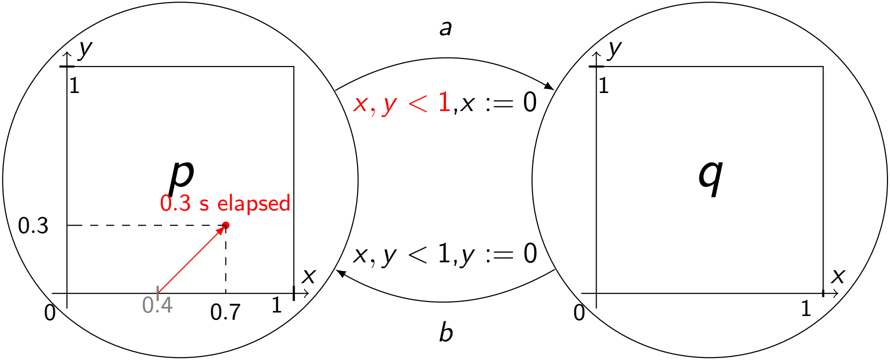
A run of the timed automaton

$\color{red}{(p,(0.4,0))} \xrightarrow{0.3,a} (q,(0,0.3)) \xrightarrow{0.2,b} (p,(0.2,0)) \xrightarrow{0.6,a} (q,(0,0.6))$
(its labelling timed words: $(0.3,a)(0.2,b)(0.6,a)$).



A run of the timed automaton

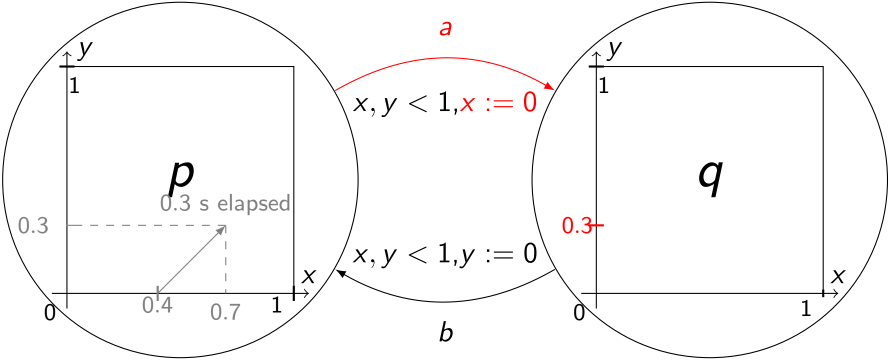
$(p,(0.4,0)) \xrightarrow{0.3,a} (q,(0,0.3)) \xrightarrow{0.2,b} (p,(0.2,0)) \xrightarrow{0.6,a} (q,(0,0.6))$



$x = 0.7 < 1$ and $y = 0.3 < 1$, the guard is satisfied.

A run of the timed automaton

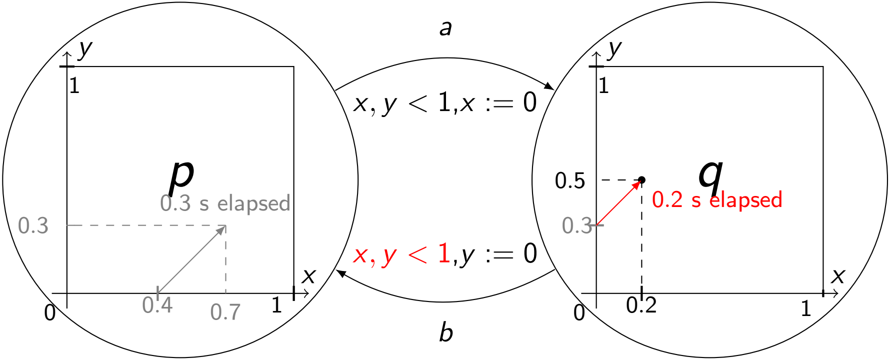
$(p,(0.4,0)) \xrightarrow{0.3,a} (q,(0,0.3)) \xrightarrow{0.2,b} (p,(0.2,0)) \xrightarrow{0.6,a} (q,(0,0.6))$



x is reset while the transition is fire, $y = 0.3$ is unchanged.

A run of the timed automaton

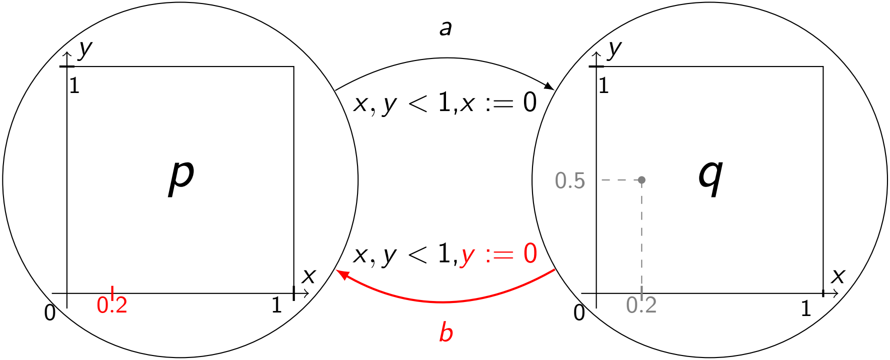
$$(p,(0.4,0)) \xrightarrow{0.3,a} (q,(0,0.3)) \xrightarrow{0.2,b} (p,(0.2,0)) \xrightarrow{0.6,a} (q,(0,0.6))$$



$x = 0.2 < 1$ and $y = 0.5 < 1$, the guard is satisfied.

A run of the timed automaton

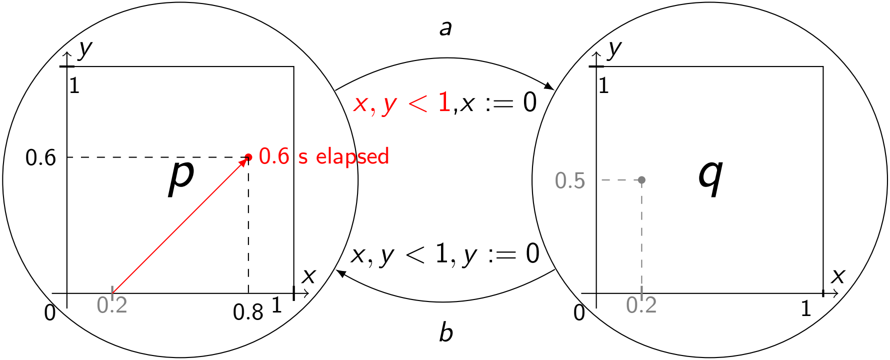
$$(p,(0.4,0)) \xrightarrow{0.3,a} (q,(0,0.3)) \xrightarrow{0.2,b} (p,(0.2,0)) \xrightarrow{0.6,a} (q,(0,0.6))$$



y is reset while the transition is fired, x is unchanged.

A run of the timed automaton

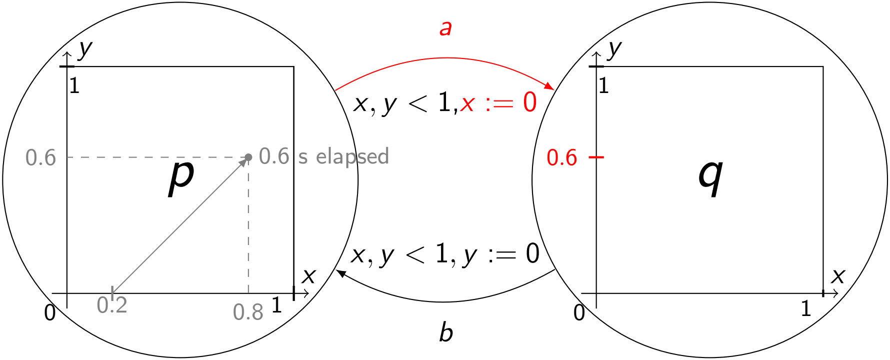
$(p,(0.4,0)) \xrightarrow{0.3,a} (q,(0,0.3)) \xrightarrow{0.2,b} (p,(0.2,0)) \xrightarrow{0.6,a} (q,(0,0.6))$



$x = 0.8 < 1$ and $y = 0.6 < 1$, the guard is satisfied.

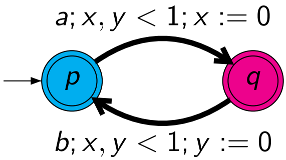
A run of the timed automaton

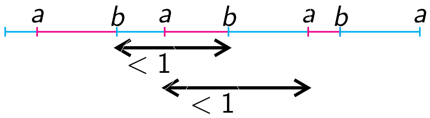
$$(p,(0.4,0)) \xrightarrow{0.3,a} (q,(0,0.3)) \xrightarrow{0.2,b} (p,(0.2,0)) \xrightarrow{0.6,a} (q,(0,0.6))$$



x is reset while the transition is fired, $y = 0.6$ is unchanged.

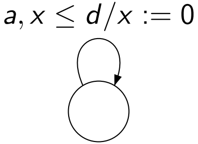
The timed language recognized



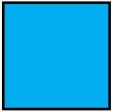
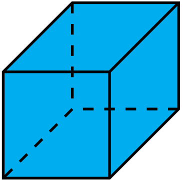


Language: $(t_1, a)(t_2, b)(t_3, a)(t_4, b)\ldots$ such that $t_i + t_{i+1} < 1$.

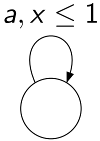
Volume of the hypercube



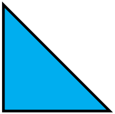
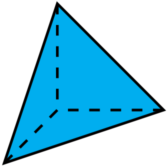
Timed word : $(t_1, a)(t_2, a) \ldots (t_n, a)$

dimension 1	dimension 2	dimension 3	dimension n
			?
$t_1 \leq d$	$t_1, t_2 \leq d$	$t_1, t_2, t_3 \leq d$	$t_1, \ldots, t_n \leq d$
Volume d	Volume d^2	Volume d^3	Volume d^n

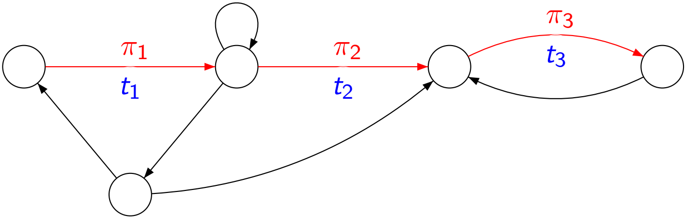
Volume of the simplex



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

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

Measuring timed languages



- Choosing a timed word $(\vec{t}, \pi) \in L_n$ = Discrete choice for the path π + Continuous choice for vector of delays $\vec{t}$.

- Given π, $L_\pi = \{\vec{t} \mid (\vec{t}, \pi) \in L_n\} \subseteq \mathbb{R}^n$ is a polytope (e.g. hypercube, simplex...)

- Measure of L_n: $V_n = \sum_{\pi \in E^n} \mathtt{Vol}(L_\pi)$ (typically $\approx \rho^n$)

- Volumetric entropy : $\mathcal{H} = \log_2 \rho = \lim \frac{1}{n} \log_2(V_n)$

Contributions

I) Characterize a dichotomy between "good" and "bad" TA:
-Thick (good): $\mathcal{H} > -\infty$ Vs. Thin (bad): $\mathcal{H} = -\infty$.

II) Quantify the quality of a discretization $\mathcal{A}_\epsilon$ of a TA $\mathcal{A}$:
-the entropy of $\mathcal{A}_\epsilon$ tends to the entropy of $\mathcal{A}$ when $\epsilon \to 0$.

III) Propose a theory of timed symbolic dynamics and metric mean dimension.

IV) Sketch a coding theory involving timed and discrete languages:
-here entropy plays the key role of information measurement.

V) Design methods to measure precisely a timed language:
-characterize generating functions of volume
$f(z) = \sum_{n\geq 0} V_n z^n$.

VI) Apply these methods to combinatorics of permutations:
-measuring volume of a particular timed languages is counting permutations in a certain class.

VII) Generate the runs of a TA in a most unbiased manner:
-use a maximal entropy approach.

Corresponding articles

I-II) B. and Asarin. **Thin and thick timed regular languages**. In *FORMATS*, 2011

IV) Asarin, B., Béal, Degorre, and Perrin. **Toward a timed theory of channel coding**. In *FORMATS*, 2012.

V) Asarin, B., Degorre, and Perrin. **Generating functions of timed languages**. In *MFCS*, 2012.

VI) B. **Counting and generating permutations using timed language**. To appear *LATIN*, 2014.

VII) B. **A maximal entropy stochastic process for a timed automaton**. In *ICALP*, 2013

Corresponding articles

I-II) B. and Asarin. **Thin and thick timed regular languages**. In *FORMATS*, 2011

IV) Asarin, B., Béal, Degorre, and Perrin. **Toward a timed theory of channel coding**. In *FORMATS*, 2012.

V) Asarin, B., Degorre, and Perrin. **Generating functions of timed languages**. In *MFCS*, 2012.

VI) B. **Counting and generating permutations using timed language**. To appear *LATIN*, 2014.

VII) B. **A maximal entropy stochastic process for a timed automaton**. In *ICALP*, 2013

Not included in the manuscript

Asarin, B., Degorre. **Spectral Gap in Timed Automata**. In *FORMATS*, 2013.

The selection of the day

- Thin & thick dichotomy + discretization.
- Generating functions of timed languages.
- Application to combinatorics of permutations.
- Maximal entropy stochastic process.

Part II

Thin and thick timed regular languages

The thin and thick dichotomy

Recall $\mathcal{H} = \lim \frac{1}{n} \log V_n$.

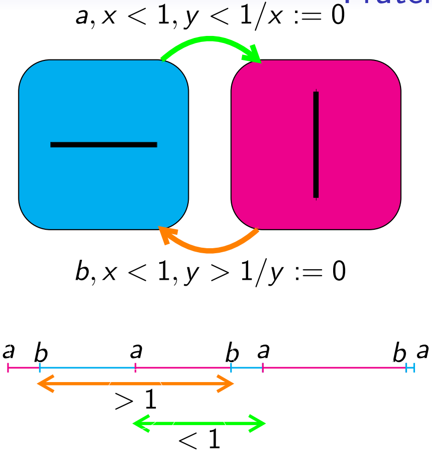
Thin

- $\mathcal{H} = -\infty$ ($V_n << d^n, \ \forall d > 0$.)
- All behaviours are "bad".
- No good discretization.
- States move toward the border of guards.

Thick

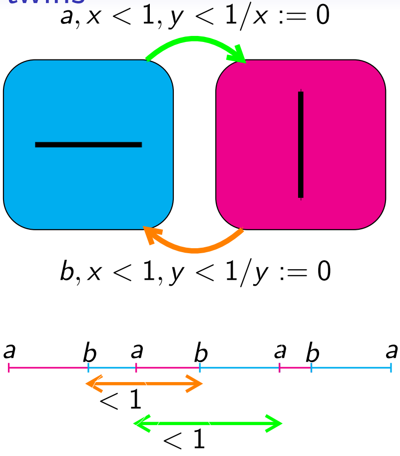
- $\mathcal{H} > -\infty$ ($V_n = d^n \times$ subexponential term.)
- Most of behaviours are "good".
- Forgetfulness
- Quantitative pumping lemma.
- Good discretization.

Fraternal twins



$t_{2n} \to_{n\to+\infty} 0$
$\mathcal{H} = -\infty$

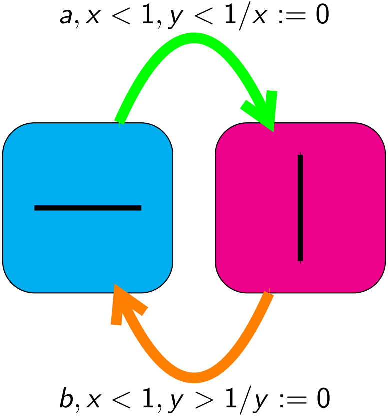
The **THIN** twin

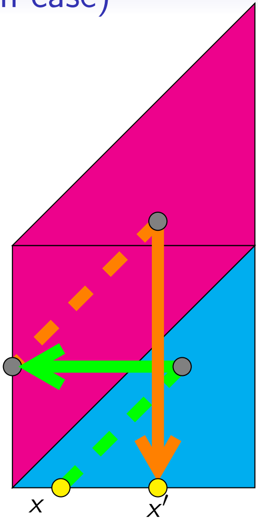


No asymptotical dependance.
$\mathcal{H} = \log_2(2/\pi)$

The **THICK** twin

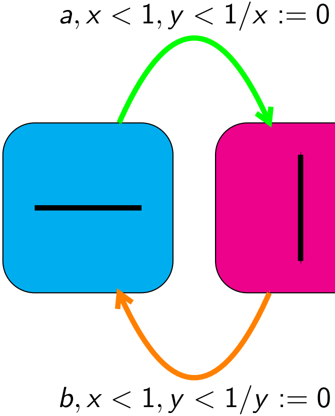
A not forgetful cycle (Thin case)

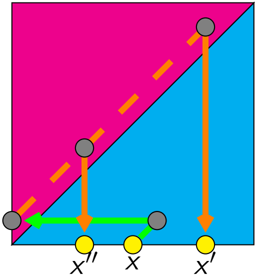




Dependance between x and x': $x' \geq x$

A forgetful cycle (Thick case)





ab is a forgetful cycle.

The characterization of thickness

Theorem

Equivalent characterizations of thick automata:

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

The thin side

1. $\mathcal{H} = -\infty$;
2. dependance between begining and end of each runs;
3. distance to border decreases.

The characterization of thickness

Theorem

Equivalent characterizations of thick automata:

1. $\mathcal{H} > -\infty$;
2. there exists a forgetful cycle;
3. there exists an ε-discrete limit cycle for some $\varepsilon > 0$.
4. robust controller synthesis is possible [SBMR13].

The thin side

1. $\mathcal{H} = -\infty$;
2. dependance between begining and end of each runs;
3. distance to border decreases.

The characterization of thickness

Theorem

Equivalent characterizations of thick automata:

1. $\mathcal{H} > -\infty$;
2. there exists a forgetful cycle;
3. there exists an ε-discrete limit cycle for some $\varepsilon > 0$.
4. robust controller synthesis is possible [SBMR13].

The thin side

1. $\mathcal{H} = -\infty$;
2. dependance between begining and end of each runs;
3. distance to border decreases.

A corollary of [SBMR13]

The thickness property is PSPACE complete

Application to discretization

- Discretization: delay and clock values multiple ε (such that $1/\varepsilon \in \mathbb{N}$).
- Finite alphabet: $\mathbb{A}_\varepsilon = \{0, \varepsilon, 2\varepsilon, \dots, M-\varepsilon, M\} \times \Sigma$ is finite.
- Finite automaton: $\mathcal{A}_\varepsilon$ recognizing $L(\mathcal{A}) \cap \mathbb{A}_\varepsilon^*$.
- ϵ-entropy: $h_\varepsilon = \lim_{n\to+\infty} \log_2(|L_n(\mathcal{A}_\varepsilon)|)/n$.

Theorem

Provided that $\mathcal{H} > -\infty$ the following asymptotic expansion holds:

$$h_\varepsilon = \mathcal{H} + \log_2(1/\varepsilon) + o(1)$$

(and an upper bound on the convergence speed can be derived from the proof).

Corollary

$\mathcal{H}$ is computable.

Part III

Generating functions of timed languages

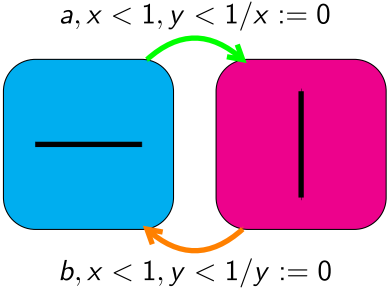
Generating functions of timed languages

Volume generating function:

$$f(z) = \sum_{n \in \mathbb{N}} V_n z^n$$

For the thick twin:

$f(z) = \tan(z) + 1/\cos(z)$
$= 1 + z + \frac{1}{2}z^2 + \frac{1}{3}z^3 + \frac{5}{24}z^4 + \ldots$



Generating functions of timed languages

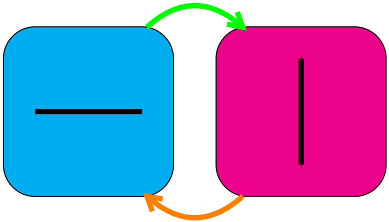
Volume generating function:

$$f(z) = \sum_{n \in \mathbb{N}} V_n z^n$$

For the thick twin:

$f(z) = \tan(z) + 1/\cos(z)$

$= 1 + z + \frac{1}{2}z^2 + \frac{1}{3}z^3 + \frac{5}{24}z^4 + \dots$



Main idea: consider starting state $s = (q, \vec{x})$ as a parameter.

- $\mathbb{L}(s)$: languages starting from s.
- $v_k(s) = \mathtt{Vol}(\mathbb{L}_k(s))$ for $k \in \mathbb{N}$.
- $f_z(s) = \sum_{k \in \mathbb{N}} v_k(s) z^k$.
- Take value in the initial state s_0: $f(z) = f_z(s_0)$.

Languages and volume equations, operator Ψ [AD09]

- States $s = (q, \vec{x}) \in \mathbb{S}$
- Timed letter $\alpha = (t, \delta) \in \mathbb{A}$
- Successor operation $s \xrightarrow{\alpha} s_\alpha$ (with $s_\alpha \in \mathbb{S} \cup \{\bot\}$).
- Integral over timed letter $\int_{\alpha \in \mathbb{A}} .d\alpha = \sum_{\delta \in \Delta} \int_0^M .dt$

Recursive equations for languages and volumes

Languages equations	volumes equations
$\mathbb{L}_0(s) = \varepsilon$ if s is final	$v_0(s) = 1_F(s)$
$\mathbb{L}_{k+1}(s) = \bigcup_{\alpha \in \mathbb{A}} \alpha \mathbb{L}_k(s_\alpha)$ with $\mathbb{L}_k(\bot) = \emptyset$	$v_{k+1}(s) = \int_{\alpha \in \mathbb{A}} v_k(s_\alpha)\, d\alpha$ with $v_k(\bot) = 0$

Volumes and operator Ψ

$$\Psi(g)(s) =_{def} \int_{\mathbb{A}} g(s_\alpha)\, d\alpha \quad \text{with } g(\bot) = 0.$$

$$v_n = \Psi(v_{n-1}) = \Psi^n(1_F).$$

Form languages equations to generating functions

Recall

Languages equations	volumes equations
$\mathbb{L}_0(s) = \varepsilon$ if s is final	$v_0(s) = 1_F(s)$
$\mathbb{L}_{k+1}(s) = \bigcup_{\alpha \in \mathbb{A}} \alpha \mathbb{L}_k(s_\alpha)$ with $\mathbb{L}_k(\bot) = \emptyset$	$v_{k+1}(s) = \int_{\alpha \in \mathbb{A}} v_k(s_\alpha)\, d\alpha$ with $v_k(\bot) = 0$

Implies $f_z(s) = \sum_{k \in \mathbb{N}} v_k(s) z^k$ fixed point of:

$$f_z(s) = z \int_{\alpha \in \mathbb{A}} f_z(s_\alpha)\, d\alpha + 1_F(s). \tag{1}$$

More succinctly: $f_z = z \Psi f_z + 1_F$.

Form languages equations to generating functions

Recall

Languages equations	volumes equations
$\mathbb{L}_0(s) = \varepsilon$ if s is final	$v_0(s) = 1_F(s)$
$\mathbb{L}_{k+1}(s) = \bigcup_{\alpha\in\mathbb{A}} \alpha\mathbb{L}_k(s_\alpha)$ with $\mathbb{L}_k(\bot) = \emptyset$	$v_{k+1}(s) = \int_{\alpha\in\mathbb{A}} v_k(s_\alpha)\, d\alpha$ with $v_k(\bot) = 0$

Implies $f_z(s) = \sum_{k\in\mathbb{N}} v_k(s) z^k$ fixed point of:

$$f_z(s) = z \int_{\alpha\in\mathbb{A}} f_z(s_\alpha)\, d\alpha + 1_F(s). \qquad (1)$$

More succinctly: $f_z = z\Psi f_z + 1_F$.

Remark

(1) can be obtained directly from:

$$\mathbb{L}(s) = \bigcup_{\alpha\in\mathbb{A}} \alpha\mathbb{L}(s_\alpha) + (\varepsilon \ \text{ if } s \text{ is final}).$$

Characterization of the generating function.

Entropy and spectral radius of Ψ [AD09]

$$\mathcal{H}(\mathcal{A}) = \log_2(\rho)$$

with spectral radius ρ = maximal eigenvalue of Ψ (i.e. $\Psi v = \rho v$).

Theorem

- The convergence radius of f is $1/\rho = 2^{-\mathcal{H}} > 0$.
- Within the disc of convergence (i.e. $z < 1/\rho$), the function f_z is the unique solution of the integral equation:

$$f_z = z\Psi f_z + 1_F.$$

Can we compute a closed form formula of f_z from this equation?

Characterization of the generating function.

Entropy and spectral radius of Ψ [AD09]

$$\mathcal{H}(\mathcal{A}) = \log_2(\rho)$$

with spectral radius $\rho =$ maximal eigenvalue of Ψ (i.e. $\Psi v = \rho v$).

Theorem

- The convergence radius of f is $1/\rho = 2^{-\mathcal{H}} > 0$.
- Within the disc of convergence (i.e. $z < 1/\rho$), the function f_z is the unique solution of the integral equation:

$$f_z = z\Psi f_z + 1_F.$$

Can we compute a closed form formula of f_z from this equation?

- In general too difficult: involve variables $\vec{x} \in \mathbb{R}^d$ with $d > 1$.

Characterization of the generating function.

Entropy and spectral radius of Ψ [AD09]

$$\mathcal{H}(\mathcal{A}) = \log_2(\rho)$$

with spectral radius $\rho =$ maximal eigenvalue of Ψ (i.e. $\Psi v = \rho v$).

Theorem

- The convergence radius of f is $1/\rho = 2^{-\mathcal{H}} > 0$.
- Within the disc of convergence (i.e. $z < 1/\rho$), the function f_z is the unique solution of the integral equation:

$$f_z = z\Psi f_z + 1_F.$$

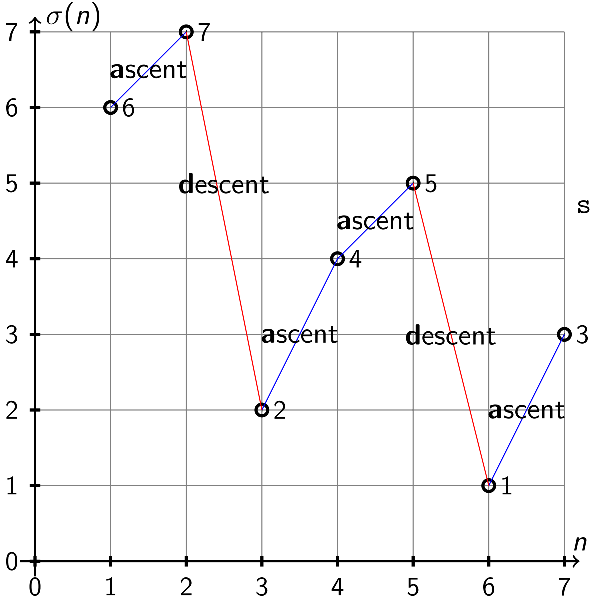
Can we compute a closed form formula of f_z from this equation?

- In general too difficult: involve variables $\vec{x} \in \mathbb{R}^d$ with $d > 1$.
- For some subclass of timed automata, yes!

Part IV

Application to enumerative combinatorics of permutations

Signature of a permutation



Signature:
$\mathrm{sg}(\sigma) =$ **adaada**

Two problem statements

Given a regular language $L \subseteq \{\mathbf{a}, \mathbf{d}\}^*$, we are interested in

$$\mathrm{sg}^{-1}(L) = \{\sigma \mid \mathrm{sg}(\sigma) \in L\}.$$

Problem 1: Enumeration

Design an algorithm that compute a closed form formula for the exponential generating function:

$$g_L(z) = \sum_{n \geq 1} \alpha_n(L) \frac{z^n}{n!}$$

where $\alpha_n(L) = |\{\sigma \in \mathfrak{S}_n \mid \mathrm{sg}(\sigma) \in L\}|$

Problem 2: Uniform sampling

Construct a uniform random sampler for $\{\sigma \in \mathfrak{S}_n \mid \mathrm{sg}(\sigma) \in L\}$. That is $Prob(output = \sigma) = \frac{1}{\alpha_n(L)}$.

Solving the two problems

We describe a GOOD class of TA such that

- computing a closed form formula for the volume generating function is easy;
- uniform sampling of timed words is easy;

... and a linear time transformation of automata:

$$\begin{array}{ccc} \text{DFA on } \{\mathbf{a}, \mathbf{d}\} & \to & \texttt{GOOD} \\ \mathcal{A} & \mapsto & \mathcal{A}' \end{array}$$

such that

- $g_L(z) = \sum_{n \in \mathbb{N}} V_n(\mathcal{A}')z^n$.
- Uniform sampling in $L_n(\mathcal{A}')$ + some easy computations

  = Uniform sampling of permutations with signature in L.

Solving the two problems

We describe a GOOD class of TA such that

- computing a closed form formula for the volume generating function is easy; mainly computing an inversion and exponential of a matrix of side $O(|\mathcal{A}|)$
- uniform sampling of timed words is easy;

... and a linear time transformation of automata:

$$\begin{array}{ccc} \text{DFA on } \{\mathbf{a}, \mathbf{d}\} & \rightarrow & \texttt{GOOD} \\ \mathcal{A} & \mapsto & \mathcal{A}' \end{array}$$

such that

- $g_L(z) = \sum_{n \in \mathbb{N}} V_n(\mathcal{A}') z^n$.
- Uniform sampling in $L_n(\mathcal{A}')$ + some easy computations

$=$ Uniform sampling of permutations with signature in L.

Solving the two problems

We describe a GOOD class of TA such that

- computing a closed form formula for the volume generating function is easy; mainly computing an inversion and exponential of a matrix of side $O(|\mathcal{A}|)$
- uniform sampling of timed words is easy; generation of each n-length timed word in $O(n)$ + preprocessing time $O(|\mathcal{A}|n^2)$

... and a linear time transformation of automata:

$$\begin{array}{ccc} \text{DFA on } \{\mathbf{a},\mathbf{d}\} & \rightarrow & \texttt{GOOD} \\ \mathcal{A} & \mapsto & \mathcal{A}' \end{array}$$

such that

- $g_L(z) = \sum_{n\in\mathbb{N}} V_n(\mathcal{A}')z^n$.
- Uniform sampling in $L_n(\mathcal{A}')$ + some easy computations

$=$ Uniform sampling of permutations with signature in L.

Solving the two problems

We describe a GOOD class of TA such that

- computing a closed form formula for the volume generating function is easy; mainly computing an inversion and exponential of a matrix of side $O(|\mathcal{A}|)$
- uniform sampling of timed words is easy; generation of each n-length timed word in $O(n)$ + preprocessing time $O(|\mathcal{A}|n^2)$

... and a linear time transformation of automata:

$$\begin{array}{ccc} \text{DFA on } \{\mathbf{a}, \mathbf{d}\} & \to & \texttt{GOOD} \\ \mathcal{A} & \mapsto & \mathcal{A}' \end{array}$$

such that

- $g_L(z) = \sum_{n\in\mathbb{N}} V_n(\mathcal{A}')z^n$.
- Uniform sampling in $L_n(\mathcal{A}')$ + some easy computations (transformation in $O(n)$ + a sort $O(n \log n)$) = Uniform sampling of permutations with signature in L.

Part V

A maximal entropy stochastic process for a TA

Practical problem statement

Generate **quickly** and as **uniformly** as possible runs of a timed automaton.

Theoretical problem statement

Design a natural stochastic process for a timed automaton.

The solution

- is based on a maximal entropy approach,
- lift to the timed case the maximal entropy Markov chain of a finite graph described by Shannon and Parry.

Possible applications in ...

verification:

proportional model checking e.g. more than 65 per cent of the runs satisfy a formula with confidence $\geq 99\%$;

enumerative combinatorics (see above):

fast (quasi) uniform generation in certain classes of permutations;

information theory:

compression of timed words in a timed regular language.

Stochastic Process Over Runs (SPOR) of a TA

A SPOR of $\mathcal{A}$ is defined by

- initial density on states, $p_0 : \mathbb{S} \to \mathbb{R}^+$;
- conditional density on timed transitions knowing a state s:$p(.|s), \mathbb{A} \to \mathbb{R}^+$.

It induces a density of probability on $\mathtt{Runs}_n$

- Chain rules $p_n(s_0 \xrightarrow{\alpha_0} s_1 \ldots \xrightarrow{\alpha_{n-1}} s_n) = p_0(s_0)p(\alpha_0|s_0)\cdots p(\alpha_{n-1}|s_{n-1})$.
- $(s_0, \alpha_0)(s_1, \alpha_1)\ldots(s_n, \alpha_n)\ldots$ stochastic process (Discrete time continuous state Markov Chain).

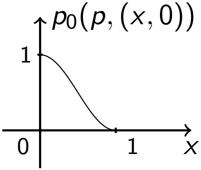
Problem statement

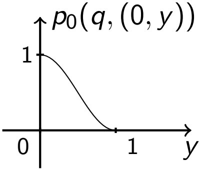
Describe a SPOR such that $p_n(r) \approx 1/\mathtt{Vol}(\mathtt{Runs}_n)$ for "almost" every run r.

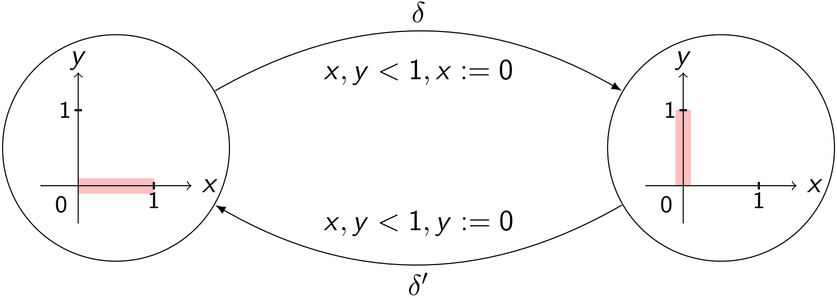
Running a SPOR (the initial density on states)

Initial probability on states:

$\int_0^1 p_0(p,(x,0))dx + \int_0^1 p_0(q,(0,y))dy = 1.$

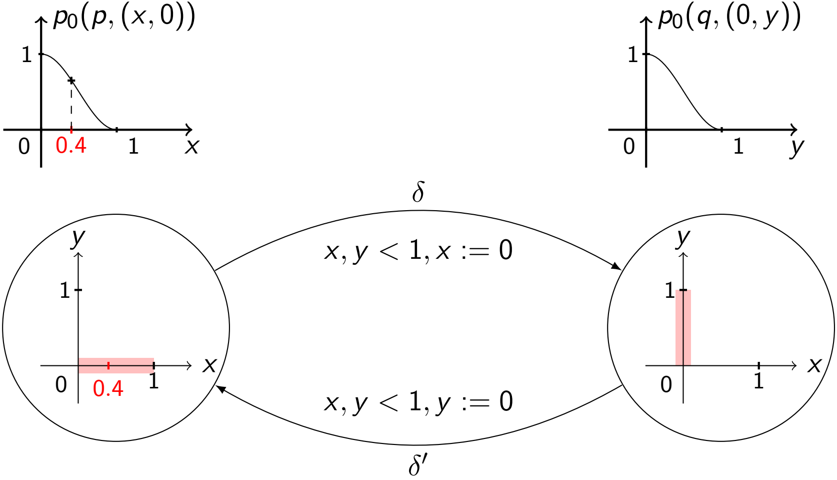






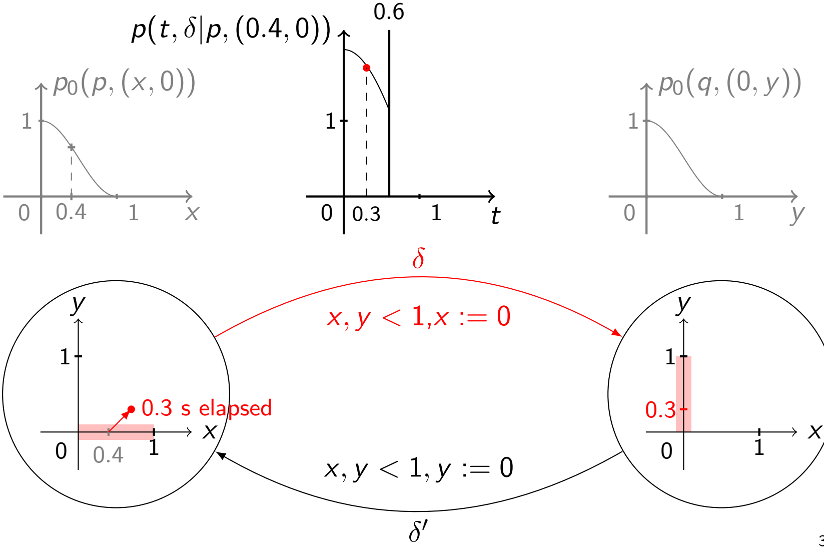
Running a SPOR (Choosing the starting state)

$(p, (0.4, 0))$



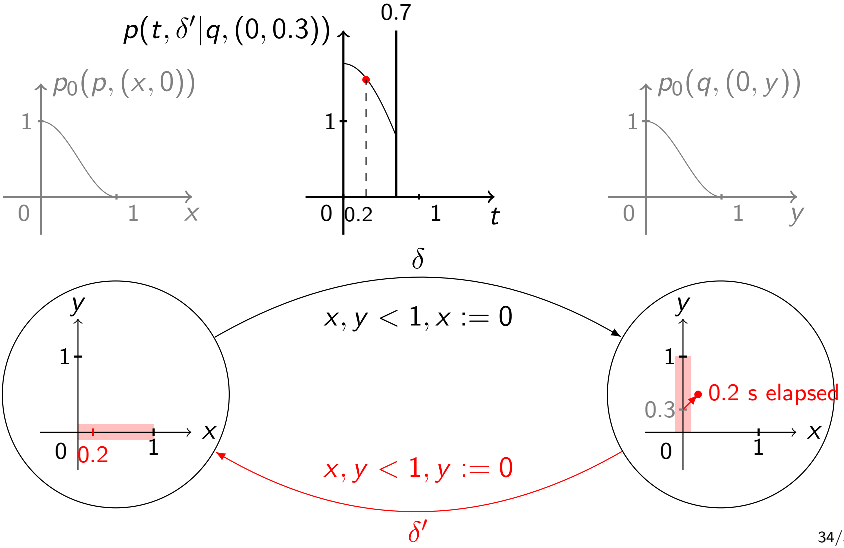
Running a SPOR (Choosing the timed transition)

$(p,(0.4,0)) \xrightarrow{0.3,\delta} (q,(0,0.3))$



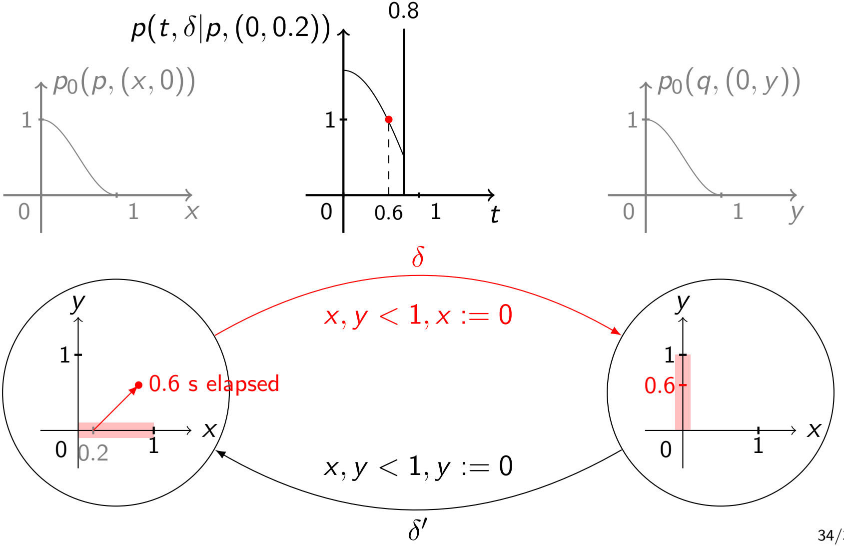
Running a SPOR (Choosing the timed transition)

$$(p,(0.4,0)) \xrightarrow{0.3,\delta} (q,(0,0.3)) \xrightarrow{0.2,\delta'} (p,(0.2,0))$$



Running a SPOR (Choosing the timed transition)

$$(p,(0.4,0)) \xrightarrow{0.3,\delta} (q,(0,0.3)) \xrightarrow{0.2,\delta'} (p,(0.2,0)) \xrightarrow{0.6,\delta} (q,(0.6,0))$$



Maximal entropy; main contributions

- Entropy of a SPOR Y on $\mathcal{A}$ (adapted from Shannon's continuous entropy):

$$h(Y) = \lim_{n \to +\infty} -\frac{1}{n} \int_{\texttt{Runs}_n} p_n(r) \log_2 p_n(r) dr \leq \mathcal{H}(\mathcal{A})$$

Maximal entropy; main contributions

- Entropy of a SPOR Y on $\mathcal{A}$ (adapted from Shannon's continuous entropy):

$$h(Y) = \lim_{n \to +\infty} -\frac{1}{n} \int_{\mathtt{Runs}_n} p_n(r) \log_2 p_n(r) dr \leq \mathcal{H}(\mathcal{A})$$

Theorem 1: Maximal entropy

There exists an ergodic Y^* of maximal entropy: $h(Y^*) = \mathcal{H}(\mathcal{A})$ (constructed later).

Maximal entropy; main contributions

- Entropy of a SPOR Y on $\mathcal{A}$ (adapted from Shannon's continuous entropy):

$$h(Y) = \lim_{n\to+\infty} -\frac{1}{n}\int_{\mathtt{Runs}_n} p_n(r)\log_2 p_n(r)dr \leq \mathcal{H}(\mathcal{A})$$

Theorem 1: Maximal entropy

There exists an ergodic Y^* of maximal entropy: $h(Y^*) = \mathcal{H}(\mathcal{A})$ (constructed later).

Theorem 2: Asymptotic equipartition property

Y^* satisfies $-\frac{1}{n}\log_2 p_n(r) \to h(Y^*)$ almost surely.

This solves the problem of quasi uniform generation of runs: $p_n(r) \approx 2^{-nh(Y*)} = 2^{-n\mathcal{H}(\mathcal{A})} \approx 1/\mathtt{Vol}(\mathtt{Runs}_n)$.

The construction of the maximal entropy SPOR Y^*

1. Take positive eigenfunctions $v, w \in L^2(\mathbb{S})$ for $\rho(\Psi)$:
   - $\Psi v = \rho v$;
   - $\Psi^* w = \rho w$;

   these eigenfunctions are **positive** a.e. and **unique** up to a scalar constant by virtue of a Perron-Frobenius like theorem.
2. Apply normalizing condition: $\langle w, v \rangle = \int_{\mathbb{S}} w(s)v(s)ds = 1$.
3. Define a SPOR Y^* with

$$
\begin{aligned}
p_0^*(s) &= w(s)v(s); \\
p^*(\alpha|s) &= \frac{v(s_\alpha)}{\rho v(s)} \quad \text{with } s \xrightarrow{\alpha} s_\alpha.
\end{aligned}
$$

The construction of the maximal entropy SPOR Y^*

1. Take positive eigenfunctions $v, w \in L^2(\mathbb{S})$ for $\rho(\Psi)$:
   - $\Psi v = \rho v$;
   - $\Psi^* w = \rho w$;

   these eigenfunctions are **positive** a.e. and **unique** up to a scalar constant by virtue of a Perron-Frobenius like theorem.
2. Apply normalizing condition: $\langle w, v \rangle = \int_{\mathbb{S}} w(s)v(s)ds = 1$.
3. Define a SPOR Y^* with

$$\begin{aligned} p_0^*(s) &= w(s)v(s); \\ p^*(\alpha|s) &= \frac{v(s_\alpha)}{\rho v(s)} \quad \text{with } s \xrightarrow{\alpha} s_\alpha. \end{aligned}$$

Fast quasi uniform sampling

With Y^*, a **quasi uniform sampling** of runs of $\mathcal{A}$ can be achieved in **linear** time.

Part VI

Conclusion

In this thesis,

volume and entropy-based studies of timed languages allowed us to

- distinguish between robust and non robust TA;
- initiate constrained channel information theory;
- contribute to simulation of timed automata and uniform generation of timed words;
- define a natural stochastic process for a timed automaton;
- formulate timed automata theory in terms of symbolic dynamics;
- design general methods for combinatorics of permutations.

To do

- Implement the combinatorics algorithms in computer algebra system like Sage.
- Uniform generation of timed words in the whole class of TA.
- Implement entropy-based tools (requires zones instead of regions).

Possible extension to...

- timed context free language;
- hybrid systems and probabilistic models.

Further work:

- Proportional model checking: quantify the set of runs satisfying a formula.
- Compression methods based on entropy.
- Entropy of natural languages (seen as timed languages).
- Empirical estimation of the entropy.