



HAL
open science

Volume and entropy of regular timed languages

Eugene Asarin, Aldric Degorre

► **To cite this version:**

Eugene Asarin, Aldric Degorre. Volume and entropy of regular timed languages. 2009. hal-00369812v1

HAL Id: hal-00369812

<https://hal.science/hal-00369812v1>

Preprint submitted on 21 Mar 2009 (v1), last revised 9 Jul 2009 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Volume and entropy of regular timed languages

Eugene Asarin¹ and Aldric Degorre²

¹ LIAFA, Université Paris Diderot / CNRS
case 7014, 75205 Paris Cedex 13, France

Eugene.Asarin@liafa.jussieu.fr

² VERIMAG,

Centre Equation, 2 av. de Vignate, 38610 Gières, France

Aldric.Degorre@imag.fr

Abstract. For timed languages, we define measures of their size: volume for a fixed finite number of events, and entropy (growth rate) as asymptotic measure for an unbounded number of events. These measures can be used for comparison of languages, and the entropy can be viewed as information contents of a timed language. In case of languages of deterministic timed automata, we give exact formulas for volumes. Next we characterize the entropy, using methods of functional analysis, as a logarithm of the leading eigenvalue (spectral radius) of a positive integral operator. We devise several methods to compute the entropy: a symbolic one for so-called “ $1\frac{1}{2}$ -clock” automata, and two numerical ones: one using techniques of functional analysis, another based on discretization.

1 Introduction

Since early 90s timed automata and timed languages are extensively used for modelling and verification of real-time systems, and thoroughly explored from a theoretical standpoint. However two important, and closely related, aspects have never been addressed: quantitative analysis of the size of these languages and of information content of timed words. In this paper we formalize and solve these problems for a large subclass of timed automata.

Recall that a timed word describes a behaviour of a system, taking into account delays between events. A timed languages, which is just a set of timed words represents, may represent all such potential behaviours. Our aim is to measure the size of such a language. For a fixed number n of events we can consider the language as a subset of $\Sigma^n \times \mathbb{R}^n$ (that is of several copies of the space \mathbb{R}^n). A natural measure in this case is just Euclidean volume V_n of this subset. When the number of events is not fixed, we can still consider for each n all the timed words with n events belonging to the language and their volume V_n . It turns out that in most cases V_n asymptotically behaves as $2^{n\mathcal{H}}$ for some constant \mathcal{H} that we call entropy of the language.

The information-theoretic meaning of \mathcal{H} can be stated as follows: for a small ε , if the delays are measured with a finite precision ε , then using the words of the language L with entropy \mathcal{H} one can transmit $\mathcal{H} + \log(1/\varepsilon)$ bits of information

per event (see Thms. 5-6 below for a formalization in terms of Kolmogorov complexity).

There can be several potential applications of these notions:

- The most direct one is capacity estimation for an information transmission channel or for a time-based information flow.
- When one overapproximates a timed language L_1 by a simpler timed language L_2 , using for example some abstractions as in [1], it is important to assess the quality of the approximation. Comparison of entropies of L_1 and L_2 provides such an assessment.
- In model-checking of timed systems it is often interesting to know the size of all the bad behaviours, and/or of the error traces produced by model-checking tools.

In this paper, we explore, and partly solve the following problems: given a deterministic timed automaton, find the volume V_n of the set of accepted words of a given length n and the entropy \mathcal{H} of the whole timed language accepted by the automaton.

1.1 Related Work

For a deterministic finite automaton \mathcal{A} whose states are all accepting (hence it accepts a prefix-closed language), the cardinality of the set L_n of all the accepted words of length n also behaves as $2^{n\mathcal{H}}$ for some entropy \mathcal{H} . This entropy can be found as logarithm of the spectral radius of adjacency matrix of reachable states of \mathcal{A} . This holds also for automata with multiplicities. The reader is referred to [2] for details. Our problems and techniques are inspired by this theory: in fact the main technical tool used to compute the entropy of finite automata is Perron-Frobenius theory for positive matrices, and we will use its extensions to infinite-dimensional operators [3].

In [4] probabilities of some timed languages and densities in the clock space are computed. Our formulae for fixed-length volumes can be seen as specialization of results of these authors to uniform measures. As for full languages, Vicario et al. need a stringent condition of full simultaneous reset of all the clocks at most every k steps, and under such a condition they provide a finite stochastic class graph that allows computing various interesting probabilities. We use a much weaker hypothesis (we require every clock to be reset at most every k steps, but these resets need not be simultaneous), and we obtain only the entropy.

1.2 Paper Organization

This paper is organized as follows. In Sect. 2 we define volumes of fixed-length timed languages and entropy of unbounded-length timed languages. We identify a subclass of deterministic timed automata, whose volumes and entropy are considered in the rest of the paper, and a normal form for such automata. Finally,

we provide an algorithm for computing the volumes of languages of such automata. In Sect. 3 we define a functional space associated to a timed automaton and a positive operator on this space. We rephrase the formulas for the volume in terms of this operator. Next we state the main result of the paper: a characterization of the entropy as the logarithm of the spectral radius of this operator. Such a characterization could seem too abstract but later on, in sections 4-5 we give three practical procedures for approximate computing this spectral radius. First, we show how to solve the eigenvector equation symbolically in case of timed automata with $1\frac{1}{2}$ clocks defined below. Next, for general timed automata we apply a “standard” iterative procedure from [3] and thus obtain an upper and a lower bound for the spectral radius/entropy. These bounds become tighter as we make more iterations. Last, in Sect. 5, also for general timed automata, we devise a procedure which provides upper and lower bounds of the entropy by discretization of the timed automaton. In the same section, and using the same method, we give an interpretation of the entropy of timed regular languages in terms of Kolmogorov complexity. We conclude the paper by some final remarks in Sect. 6. Throughout the paper, the concepts and the techniques are illustrated by several running examples.

2 Problem statement

2.1 Geometry, Volume and Entropy of Timed Languages

A *timed word* of size n over an alphabet Σ is a sequence $w = t_1, a_1, t_2, \dots, t_n, a_n$, where $a_i \in \Sigma, t_i \in \mathbb{R}$ and $0 \leq t_i$. Here t_i represents the delay between the events a_{i-1} and a_i . With such a timed word w of size n we associate its *untiming* $\eta(w) = a_1, \dots, a_n \in \Sigma^n$ (which is just a word), and its *timing* which is a point $\theta(w) = (t_1, \dots, t_n)$ in \mathbb{R}^n . A timed language L is a set of timed words. For a fixed n we define the *n-volume* of L as follows:

$$V_n(L) = \sum_{v \in \Sigma^n} \text{Vol}\{\theta(w) \mid w \in L, \eta(w) = v\},$$

where Vol stands for the standard Euclidean volume in \mathbb{R}^n . In other words, we sum up over all the possible untimings v of size n the volumes of corresponding sets of delays in \mathbb{R}^n . In case of regular timed languages, these sets are polyhedral, and hence their volumes (finite or infinite) are well-defined.

We have associated with every timed language a sequence of n -volumes V_n . We will show in Sect. 2.5 that, for languages of a large class of timed automata, V_n is a computable sequence of rational numbers. However, we would like to find a unique real number characterizing the asymptotic behaviour of V_n as $n \rightarrow \infty$. Typically, as we will see in Sect. 3, V_n depends approximately exponentially on n . We define entropy of the language $L(\mathcal{A})$ as the rate of this dependence.

Formally, for a timed language L we define its *entropy* as follows³:

$$\mathcal{H}(L) = \limsup_{n \rightarrow \infty} \frac{\log_2 V_n}{n}.$$

Remark 1. Many authors consider a slightly different kind of timed words: sequences $w = (a_1, d_1), \dots, (a_n, d_n)$, where $a_i \in \Sigma, d_i \in \mathbb{R}$ and $0 \leq d_1 \leq \dots \leq d_n$, with d_i representing the date of the event a_i . This definition is in fact isomorphic to ours by a change of variables : $t_1 = d_1$ and $t_i = d_i - d_{i-1}$ for $i = 2..n$. It is important for us that this change of variables preserves the n -volume, since it is linear and its matrix has determinant 1. So, choosing date (d_i) or delay (t_i) representation has no influence on language volumes (and entropy). Due to the authors' preferences (justified in [5]) the delays will be used in the sequel.

2.2 Three Examples

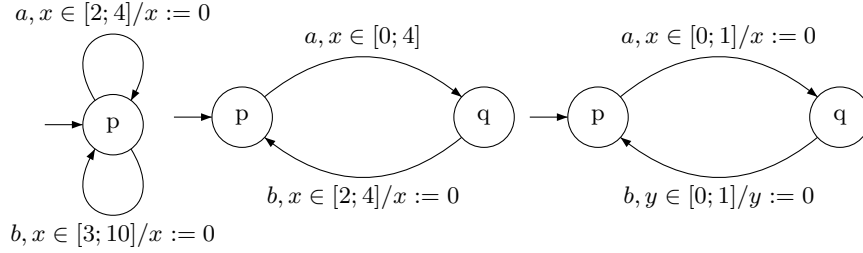


Fig. 1. Three simple timed automata $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$

To illustrate the problem consider the languages recognized by three timed automata on Fig. 1. Two of them can be analysed directly, using definitions and common sense. The third one resists naive analysis, and will be used to illustrate more advanced methods throughout the paper.

Rectangles. Consider the timed language defined by the expression

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

It can be recognized by the first automaton on Fig. 1.

³ In fact, due to Assumption A2, the languages we consider in the paper are prefix-closed, and \limsup is in fact a \lim . It will be stated formally in Corollary 1

For a given untiming $w \in \{a, b\}^n$ containing k letters a and $n - k$ letters b , the set of possible timings is a rectangle in \mathbb{R}^n of a volume $2^k 7^{n-k}$ (notice that there are C_n^k such untimings). Summing up all the volumes we obtain

$$V_n(L_1) = \sum_{k=0}^n C_n^k 2^k 7^{n-k} = (2 + 7)^n = 9^n,$$

and the entropy

$$\mathcal{H}(L_1) = \log 9 \approx 3.17.$$

A Product of Trapezia. Consider the language defined by the automaton \mathcal{A}_2 on Fig. 1, that is containing words of the form $t_1 a s_1 b t_2 a s_2 b \dots t_k a s_k b$ such that $2 \leq t_i + s_i \leq 4$. Since we want prefix-closed languages, the last $s_k b$ can be omitted.

For an even $n = 2k$ the only possible untiming is $(ab)^k$. The set of timings in \mathbb{R}^{2k} is a Cartesian product of k trapezia $2 \leq t_i + s_i \leq 4$. The surface of each trapezium equals $S = 4^2/2 - 2^2/2 = 6$, and the volume $V_{2k}(L_2) = 6^k$.

For an odd $n = 2k + 1$ the same product of trapezia is combined with an interval $0 \leq t_{k+1} \leq 4$, hence the volume is $V_{2k+1}(L_2) = 6^k \cdot 4$.

Thus the entropy

$$\mathcal{H}(L_2) = \log 6/2 \approx 1.29.$$

Our Favourite Example. The language recognized by the automaton \mathcal{A}_3 on Fig. 1 contains the words of the form $t_1 a t_2 b t_3 a t_4 b \dots$ with $t_i + t_{i+1} \in [0; 1]$. Notice that the automaton has two clocks that are never reset together. The geometric form of possible untimings in \mathbb{R}^n is defined by overlapping constraints $t_i + t_{i+1} \in [0; 1]$.

It is not so evident how to compute the volume of this polyhedron. A systematic method is described below in Sect. 2.5. Another solution would be to integrate 1 over the polyhedron, and to rewrite this multiple integral as an iterated one. The resulting formula for the volume would be

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

In the sequel we will also compute the entropy of this language using each of the methods presented in this paper.

2.3 A Subclass of Timed Automata

In the rest of the paper, we compute volumes and entropy for regular timed languages recognized by a subclass of timed automata (TA). We assume that the reader is acquainted with timed automata, otherwise we refer her or him to [6] for details. Here we only fix notations for ingredients of timed automata and state several assumptions that the automata should satisfy. Thus a TA is

a tuple $\mathcal{A} = (Q, \Sigma, C, \Delta, q_0)$. Its elements are respectively the set of locations, the alphabet, the set of clocks, the transition relation, and the initial location (we don't need to specify accepting states due to A2 below). A generic state of \mathcal{A} is a couple (q, \mathbf{x}) of a control location and a vector of clock values. A generic element of Δ is written as $\delta = (q, a, \mathbf{g}, \mathbf{r}, q')$ (a transition from q to q' with label a , guard \mathbf{g} and reset \mathbf{r}). We spare the reader of definitions of a run of \mathcal{A} and of its accepted language.

We call a timed automaton \mathcal{A} *nice* if it satisfies the following Assumptions:

- A1. The automaton \mathcal{A} is deterministic⁴.
- A2. All its states are accepting.
- A3. Guards are rectangular. Every guard upper bounds at least one clock.
- A4. On every run segment of D transitions every clock is reset at least once.

Below we motivate and justify these choices:

- A1: Most of known techniques to compute entropy of regular languages work on deterministic automata. Indeed, these techniques count paths in the automaton, and only in the deterministic case their number coincides with the number of accepted words. The same is true for volumes in timed automata. Unfortunately, for timed automata the determinization is impossible, and A1 is restrictive.
- A2: Prefix-closed languages are natural in the entropy context, and somewhat easier to study. These languages constitute the natural model for the set of behaviours of causal systems.
- A3: If a guard of a feasible transition is infinite, the volume becomes infinite. We conclude that A3 is unavoidable and almost not restrictive.
- A4: We use this rather strong non-Zenoness condition several times in our proofs and constructions. As shows a simple example represented on Fig. 2, if we omit this assumption some anomalies can occur.



Fig. 2. An automaton without resets

The language of this automaton is

$$L = \{t_1 a \dots t_n a \mid 0 \leq \sum t_i \leq 1\},$$

and V_n is the volume of an n -dimensional simplex defined by the constraints $0 \leq \sum t_i \leq 1$, and $0 \leq t_i$. Hence $V_n = 1/n!$ which decreases faster than any

⁴ That is for any two transitions with the same source and the same label m their guards are disjoint

exponent, which is too fine to be distinguished by our methods. Assumption A4 rules out such anomalies.

This assumption is also the most difficult to check. A possible way would be to explore all simple cycles in the region graph and to check that all of those reset every clock.

2.4 Preprocessing Timed Automata

In order to compute volumes V_n and entropy \mathcal{H} of the language of a nice timed automaton, we first transform this automaton in a normal form. This form can be considered as a (timed) variant of the region graph from [6]. We suppose that the reader is familiar with the region equivalence relation (cf. [6]).

We say that a (nice) timed automaton $\mathcal{A} = (Q, \Sigma, C, \delta, q_0)$ is in a *region-split form*, if the following properties hold:

- B1. Each location and each transition of \mathcal{A} is visited by some run (starting at $(q_0, 0)$).
- B2. For every location $q \in Q$ a unique clock region \mathbf{r}_q (called its *entry region*) exists, such that runs of \mathcal{A} enter q only with clock values $\mathbf{x} \in \mathbf{r}_q$, and with all such clock values. For the initial location q_0 its entry region is the singleton $\{0\}$.
- B3. The guard \mathbf{g} of every transition $\delta = (q, a, \mathbf{g}, \tau, q') \in \Delta$ is just one clock region.

Notice, that B2 and B3 imply for every δ that $\tau(\mathbf{g}) = \mathbf{r}_{q'}$.

Proposition 1. *Given a nice timed automaton \mathcal{A} , a region-split nice timed automaton \mathcal{A}' accepting the same language can be constructed.*

Proof (sketch). Let $\mathcal{A} = (Q, \Sigma, C, \Delta, q_0)$ be a nice timed automaton and \mathbf{Reg} the set of its regions. The region-split automaton $\mathcal{A}' = (Q', \Sigma, C, \Delta', q'_0)$ can be constructed as follows:

1. Split every state q into substates corresponding to all possible entry regions. Formally, just take $Q' = Q \times \mathbf{Reg}$.
2. Split every transition from q to q' according to two clock regions: one for the clock values when q is entered, another for clock values when q is left. Formally, for every $\delta = (q, a, \mathbf{g}, \tau, q')$ of \mathcal{A} , and every two clock regions \mathbf{r} and \mathbf{r}' such that \mathbf{r}' is reachable from \mathbf{r} by time progress, and $\mathbf{r}' \subset \mathbf{g}$, we define a new transition of \mathcal{A}'

$$\delta'_{\mathbf{r}\mathbf{r}'} = ((q, \mathbf{r}), a, \mathbf{x} \in \mathbf{r}', \tau, (q', \tau(\mathbf{r}'))).$$

3. Take the initial state $q'_0 = (q_0, \{0\})$.
4. Remove all the states and all the transitions unreachable from the initial state. \square

Notice that due to A3 all the guards of original automaton are bounded w.r.t. some clock. Hence, the same holds for smaller (one-region) guards of \mathcal{A}' , that is the infinite region $[M; \infty)^{|C|}$ never occurs as a guard.

We could work with the region-split automaton, but it has too many useless (degenerate) states and transitions which do not contribute to the volume and the entropy of the language. This justifies the following definition: we say that a region-split nice timed automaton is *fleshy* if the following holds:

B4. For every transition δ its guard \mathbf{g} has no constraints of the form $x = c$ in its definition.

Proposition 2. *Given a region-split nice timed automaton \mathcal{A} accepting a language L , a fleshy region-split nice timed automaton \mathcal{A}' accepting a language $L' \subset L$ with $V_n(L') = V_n(L)$ and $\mathcal{H}(L') = \mathcal{H}(L)$ can be constructed.*

Proof (sketch). The construction is straightforward:

1. remove all non-fleshy transitions;
2. and remove all the states and transitions that became unreachable.

Inclusion $L' \subset L$ is immediate. Every path in \mathcal{A} (of length n) involving a non-fleshy (punctual) transition corresponds to the set of timings in \mathbb{R}^n which is degenerate (its dimension is smaller than n), hence it does not contribute to V_n . \square

From now on, and till the end of Sect. 4 we suppose w.l.o.g. that the nice automaton \mathcal{A} is a fleshy region-split one (see Fig. 3).

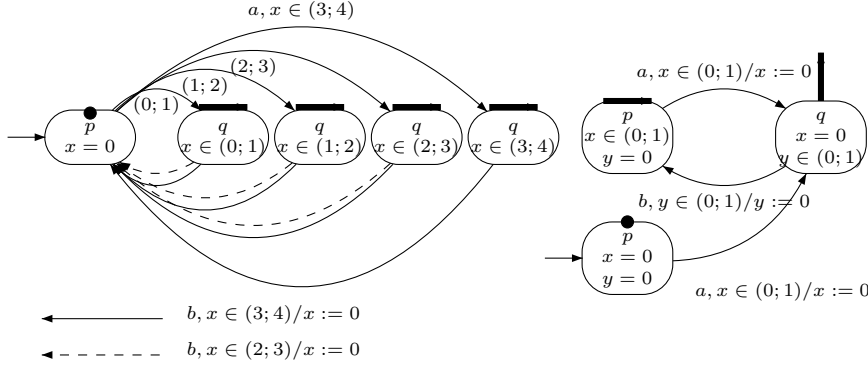


Fig. 3. Fleshy region-split forms of automata \mathcal{A}_2 and \mathcal{A}_3 from Fig. 1. An entry region is drawn over each location.

2.5 Computing Volumes

Given a nice timed automaton (in a fleshy region-split form) \mathcal{A} , we want to compute n -volumes V_n of its language. In order to obtain recurrent equations on these volumes, we need to take into account all possible initial locations and clock configurations in entry regions of these locations. For every state (q, \mathbf{x}) , where $\mathbf{x} \in \mathbf{r}_q$ we define the language $L(q, \mathbf{x})$ of all the timed words corresponding to the runs of the automaton starting at this state. $L_n(q, \mathbf{x})$ is its sublanguage consisting of its words of size n , and $v_n(q, \mathbf{x})$ is the volume of this sublanguage. Hence, the quantity we are interested in, is a value of v_n in the initial state:

$$V_n = v_n(q_0, 0).$$

By definition of runs of a timed automaton we obtain the following equations on languages L_n :

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

Since the automaton is deterministic, the union over transitions (the first \bigcup in the formula) is disjoint. Hence, it is easy to pass to volumes:

$$v_0(q, \mathbf{x}) = 1; \tag{1}$$

$$v_{k+1}(q, \mathbf{x}) = \sum_{(q, a, \mathbf{g}, \mathbf{r}, q') \in \Delta} \int_{\tau: \mathbf{x} + \tau \in \mathbf{g}} v_k(q', \mathbf{r}(\mathbf{x} + \tau)) d\tau. \tag{2}$$

Remark that for a fixed location q , and $\mathbf{x} \in \mathbf{r}_q$, since the guard is just a region, the integration over $\tau: \mathbf{x} + \tau \in \mathbf{g}$ is nothing else than \int_l^u with bounds l and u either constants or of the form $c - x_i$ with c an integer and x_i a clock variable.

These formulas lead to the following structural description of $v_n(q, \mathbf{x})$, which can be proved by a straightforward induction.

Proposition 3. *The function $v_n(q, \mathbf{x})$ restricted to a location q can be expressed by a polynomial of degree n with rational coefficients in variables \mathbf{x} .*

Thus in order to compute the volume V_n one should find by symbolic integration polynomial functions $v_k(q, \mathbf{x})$ for $k = 0..n$, and finally compute $v_n(q_0, 0)$.

Theorem 1. *For a (nice) timed automaton \mathcal{A} the volume V_n is a rational number, computable from \mathcal{A} and n using the algorithm described above.*

3 Operator Approach

In this central section of the paper we develop an approach to volumes and entropy of languages of nice timed automata based on the functional analysis.

We start in 3.1 by identifying a functional space \mathcal{F} containing the volume functions v_n . Next we show that these volume functions can be seen as iterates of some positive integral operator Ψ on this space applied to unit function (Section 3.2). We explore some elementary properties of this operator, and relate them to properties of the timed automaton in 3.3. This makes it possible to apply in 3.4 the theory of positive operators to Ψ and to deduce the main theorem of this paper stating that entropy equals logarithm of the spectral radius of Ψ .

3.1 The Functional Space of a Timed Automaton

In order to use the operator approach we first identify the appropriate functional space \mathcal{F} containing volume functions v_n .

We define S as the disjoint union of all the entry regions of all the states of \mathcal{A} . Formally, $S = \{(q, \mathbf{x}) \mid \mathbf{x} \in \mathbf{r}_q\}$. The elements of the space \mathcal{F} are bounded continuous functions from S to \mathbb{R} . The uniform norm $\|u\| = \sup_{\xi \in S} |u(\xi)|$ can be defined on \mathcal{F} . Thus we obtain a Banach space structure. We can compare two functions in \mathcal{F} pointwise, thus we write $u \leq v$ if $\forall \xi \in S : u(\xi) \leq v(\xi)$. For a function $f \in \mathcal{F}$ we sometimes denote $f(p, x)$ by $f_p(x)$. Thus any function $f \in \mathcal{F}$ can be seen as a finite collection of functions f_p defined on entry regions \mathbf{r}_p of locations of \mathcal{A} . The volume functions v_n (restricted to S) can be considered as elements of \mathcal{F} .

3.2 Volumes Revisited

Let us consider again the recurrent formula (2). It has the following form:

$$v_{k+1} = \Psi v_k,$$

where Ψ is the positive linear operator on \mathcal{F} defined by the equation:

$$\Psi f(q, \mathbf{x}) = \sum_{(q, a, \mathbf{g}, \tau, q') \in \Delta} \int_{\mathbf{x} + \tau \in \mathbf{g}} f(q', \tau(\mathbf{x} + \tau)) d\tau. \quad (3)$$

We have also $v_0 = 1$. Hence

$$v_n = \Psi^n 1,$$

and the problem of computation of volumes and entropy is now phrased as study of iterations of a particular positive bounded linear operator Ψ on the functional space \mathcal{F} . Thus the theory of positive operators can be applied. In two words this theory guarantees, that under some hypotheses, v_n will be close in direction to a positive eigenvector v^* of Ψ , corresponding to its leading eigenvalue ρ . Moreover, values of v_n will grow/decay exponentially like ρ^n . In the next section we will systematically refer to the book [3] when a result concerning positive operators is needed.

3.3 Exploring the Operator Ψ

Before applying the theory of positive operators to Ψ we need to state some elementary properties of this operator.

We start by rewriting (3) as an operator on \mathcal{F} and separating all its summands.

$$(\Psi f)_q(\mathbf{x}) = \sum_{\delta=(q,\dots,q') \in \Delta} (\psi_\delta f_{q'})_q(\mathbf{x}). \quad (4)$$

For $\delta = (q, a, \mathbf{g}, \mathbf{r}, q')$ the operator ψ_δ acts from the space $C(\mathbf{r}_{q'})$ of bounded continuous functions on the target region to the space $C(\mathbf{r}_q)$ of functions on the source region. It is defined by the integral:

$$\psi_\delta f(\mathbf{x}) = \int_{\mathbf{x}+\tau \in \mathbf{g}} f(\mathbf{r}(\mathbf{x} + \tau)) d\tau.$$

We enumerate some properties of ψ that will be used in the sequel:

Proposition 4. *Consider ψ_δ associated with a transition δ . Then, for any $f \in C(\mathbf{r}_q)$:*

1. *If $f \geq 0$ and f is not identically 0 then $\psi_\delta f$ is not identically 0.*
2. *$\|\psi_\delta f\| \leq \|f\|$ (in other words, $\|\psi_\delta\| \leq 1$).*
3. *If δ resets x_i then $\psi_\delta f$ is continuously differentiable by x_i and $\|\frac{\partial}{\partial x_i} \psi_\delta f\| \leq \|f\|$*
4. *If δ does not reset x_i and f is continuously differentiable by x_i , then $\psi_\delta f$ is continuously differentiable by x_i and $\|\frac{\partial}{\partial x_i} \psi_\delta f\| \leq \|f\| + \|\frac{\partial}{\partial x_i} f\|$.*

Proof.

(1) Let $\mathbf{x}_1 \in \mathbf{r}_{q'}$ be such that $f(\mathbf{x}_1) > 0$.

By B2 and B3, we know that there exists $\mathbf{x}_2 \in \mathbf{g}$ such that $\mathbf{r}(\mathbf{x}_2) = \mathbf{x}_1$. As $\mathbf{x}_2 \in \mathbf{g}$, there also exists $\mathbf{x}_3 \in \mathbf{r}_q$ and $\tau \in \mathbb{R}_{\geq 0}$ verifying $\mathbf{x}_2 = \mathbf{x}_3 + \tau_0$.

Furthermore, because δ is fleshy, there exists τ_1 and τ_2 , $\tau_1 < \tau_2$, such that for every $\tau \in [\tau_1, \tau_2]$, $\mathbf{x}_3 + \tau \in \mathbf{g}$.

Put together, the integration interval of $\psi_\delta f(\mathbf{x}_3) = \int_{\mathbf{x}_3+\tau \in \mathbf{g}} f(\mathbf{r}(\mathbf{x}_3 + \tau)) d\tau$ contains a value τ_0 , for which the integrated function is positive, and includes $[\tau_1, \tau_2]$, thus is neither empty or a singleton. The integrated function being non-negative and continuous, its integral, $\psi_\delta f(\mathbf{x}_3)$, is positive.

(2) In $\psi_\delta f(\mathbf{x}) = \int_{\mathbf{x}+\tau \in \mathbf{g}} f(\mathbf{r}(\mathbf{x}+\tau)) d\tau$, we estimate $|f|$ from above by the constant $\|B\|$ and the length of the integration interval by 1, as it is included in the region \mathbf{g} . This gives us the requested bound.

(3) As \mathbf{r} resets x_i , $f(q', \mathbf{r}(\mathbf{x} + \tau))$ does not depend on x_i , and thus $\psi_\delta(q, x) = \int_{\mathbf{x}+\tau \in \mathbf{g}} f(q', \mathbf{r}(\mathbf{x} + \tau)) d\tau$ is differentiable by x_i . Its derivative is:

$$\frac{\partial}{\partial x_i} \psi_\delta f(q, \mathbf{x}) = \frac{\partial}{\partial x_i} \int_{\mathbf{x}+\tau \in \mathbf{g}} f(q', \mathbf{r}(\mathbf{x} + \tau)) d\tau \quad (5)$$

$$= \pm(f(q', \mathbf{r}(\mathbf{x} + \tau_{max})) - f(q', \mathbf{r}(\mathbf{x} + \tau_{min}))). \quad (6)$$

The choice of + or – sign in the line (6) and the bounds τ_{max} and τ_{min} depend on the form of the guard.

First observe the latter term is a sum of continuous functions and, as such, is continuous. Furthermore this term is bounded in absolute value by $\|f\|$. Thus we prove $\frac{\partial}{\partial x_i} \psi_\delta f(q, \mathbf{x}) \leq \|f\|$.

(4) As f is differentiable by x_i , then so is $\psi_\delta f(q, \mathbf{x}) = \int_{\mathbf{x}+\tau \in \mathfrak{g}} f(q', \mathbf{r}(\mathbf{x} + \tau)) d\tau$. Let us differentiate it:

$$\begin{aligned} \frac{\partial}{\partial x_i} \psi_\delta f(q, \mathbf{x}) &= \frac{\partial}{\partial x_i} \int_{\mathbf{x}+\tau \in \mathfrak{g}} f(q', \mathbf{r}(\mathbf{x} + \tau)) d\tau \\ \frac{\partial}{\partial x_i} \psi_\delta f(q, \mathbf{x}) &= \pm (f(q', \mathbf{r}(\mathbf{x} + \tau_{max})) - f(q', \mathbf{r}(\mathbf{x} + \tau_{min}))) \\ &\quad + \int_{\mathbf{x}+\tau \in \mathfrak{g}} \frac{\partial}{\partial x_i} f(q', \mathbf{r}(\mathbf{x} + \tau)) d\tau. \end{aligned}$$

The resulting expression is still continuous in x_i . Indeed the newly added term in the latest equality is an integral of a continuous function that does not depend on x_i on an interval that continuously depends on x_i .

We already stated that $|(f(q', \mathbf{r}(\mathbf{x} + \tau_{max})) - f(q', \mathbf{r}(\mathbf{x} + \tau_{min})))|$ is smaller than $\|f\|$. Also in $\int_{\mathbf{x}+\tau \in \mathfrak{g}} \frac{\partial}{\partial x_i} f(q', \mathbf{r}(\mathbf{x} + \tau)) d\tau$, we can estimate the integrated function from above by the norm $\|\frac{\partial}{\partial x_i} f\|$. As the integration interval is smaller than one, the integral is smaller than this norm too. Hence the required inequality holds: $|\frac{\partial}{\partial x_i} \psi_\delta f(q, \mathbf{x})| \leq \|f\| + \|\frac{\partial}{\partial x_i} f\|$. \square

Sometimes we need to use powers of operator Ψ . Iterating (4) we obtain that

$$(\Psi^k f)_p(\mathbf{x}) = \sum_{\delta_1 \dots \delta_D \text{ from } p \text{ to } p'} (\psi_{\delta_1} \dots \psi_{\delta_D} f_{p'}) (\mathbf{x}).$$

Using this formula and proposition 4 we obtain:

Proposition 5. *Consider operator Ψ .*

1. *If $f \geq 0$ not zero on p' and there is a path of length k from p to p' then $\Psi^k f$ is not identically zero on p .*
2. *For D defined in assumption A_4 there exists a constant $E \in \mathbb{R}$ such that for any $f \in \mathcal{F}$ the following estimate hold:*

$$\forall i : \left\| \frac{\partial}{\partial x_i} \Psi^D f \right\| \leq E \|f\|.$$

Proof.

- (1) This is a straightforward induction using Prop. 4-1.
- (2) For some x_i , and a location p , the following equality holds:

$$\frac{\partial}{\partial x_i} (\Psi^D f)_p(\mathbf{x}) = \sum_{\delta_1 \dots \delta_D \text{ from } p \text{ to } p'} \frac{\partial}{\partial x_i} (\psi_{\delta_1} \dots \psi_{\delta_D} f_{p'}) (\mathbf{x}).$$

Let us consider one term (one path) of this sum. By hypothesis, in this path, there is a first transition δ_k , $1 \leq k \leq D$, such that δ_k resets x_i .

By Prop. 4-3, $\psi_{\delta_k} \dots \psi_{\delta_D} f_{p'}$ is continuously differentiable by x_i . By induction and using Prop. 4-4, it follows that $\psi_{\delta_1} \dots \psi_{\delta_k} \dots \psi_{\delta_D} f_{p'}$ is also continuously differentiable by x_i .

Now we differentiate this term. For every j , $1 \leq j \leq D$, iterating Prop. 4-2 $D - j$ times, we obtain $\|\psi_{\delta_j} \dots \psi_{\delta_D} f_{p'}\| \leq \|f\|$. Thus, by Prop. 4-3, we have $\left\| \frac{\partial}{\partial x_i} \psi_{\delta_k} \dots \psi_{\delta_D} f_{p'} \right\| \leq \|f\|$. It follows by induction on the path, using Prop. 4-4, that $\left\| \frac{\partial}{\partial x_i} \psi_{\delta_1} \dots \psi_{\delta_k} \dots \psi_{\delta_D} f_{p'} \right\| \leq k \|f\|$.

Now, if we come back to the sum, we have, at least, the following bound: $\left\| \frac{\partial}{\partial x_i} (\Psi^D f)_p \right\| \leq |\Delta|^D D \|f\|$, true for every p , therefore $\left\| \frac{\partial}{\partial x_i} \Psi^D f \right\| \leq |\Delta|^D D \|f\|$.

And finally we deduce an important property of Ψ^D .

Proposition 6. *For D defined in assumption A4 the operator Ψ^D is compact on \mathcal{F} .*

Proof. Recall that D is the lowest integer such that in every path of length D , every clock is reset at least once.

Consider \mathcal{B} the unit ball of \mathcal{F} . Let us prove that $\Psi^D \mathcal{B}$ is a compact set. This set is clearly bounded. It follows from Prop. 5-2, that the whole set $\Psi^D \mathcal{B}$ is Lipschitz continuous with constant $|c|E$ where $|c|$ is the dimension of the clock space. Hence it is equicontinuous, and, by Arzela-Ascoli theorem it is compact. \square

3.4 Applying Operators

The main result of this paper can now be stated.

Theorem 2. *For any nice timed automaton \mathcal{A} the entropy \mathcal{H} of its language coincides with logarithm of the spectral radius of the Ψ operator defined on \mathcal{F} .*

Proof. Denote by ρ the spectral radius of Ψ .

Lemma 1. *If $\rho > 0$ then it is an eigenvalue of Ψ with corresponding eigenvector $v^* \geq 0$.*

Proof (of Lemma). According to Thm. 9.4 of [3] the statement holds for every positive linear operator with a compact power. Thus the result follows immediately from Prop. 6. \square

Lemma 2. *If $\rho > 0$ then the eigenvector corresponding to the spectral radius satisfies $v^*(q_0, 0) > 0$.*

Proof (of Lemma). Let (p, \mathbf{x}) be a state for which v^* is positive. Consider a path from $(q_0, 0)$ to (p, \mathbf{x}) , and let k be its length. By Prop. 5-1, the function $\Psi^k v^*$ is not identically zero on the region of $(q_0, 0)$. Since this region is a singleton, this means that $(\Psi^k v^*)(q_0, 0) > 0$. Since v^* is an eigenvector, we rewrite this as $\rho^k v^*(q_0, 0) > 0$, and the statement is immediate. \square

We can now prove the theorem. First notice that

$$V_n = v_n(q_0; 0) \leq \|v_n\| = \|\Psi^n \mathbf{1}\| \leq \|\Psi^n\|.$$

Taking logarithm and dividing by n we obtain

$$\log V_n/n \leq \log \|\Psi^n\|/n.$$

The limit of the right-hand side is $\log \rho$ due to Gelfand's formula for spectral radius:

$$\rho = \lim_{n \rightarrow \infty} \|\Psi^n\|^{1/n}.$$

We conclude that

$$\mathcal{H} = \limsup_{n \rightarrow \infty} \log V_n/n \leq \log \rho.$$

This upper bound for the entropy is proved.

In the case when $\rho > 0$ we also have to prove the lower bound. In this case an eigenvector $v^* > 0$ with norm 1 exists. Then

$$v^* \leq 1.$$

For any natural n we can apply the positive operator Ψ^n to the previous inequality. Using the fact that v^* is an eigenvector and the formula for v_n we obtain:

$$\rho^n v^* \leq v_n.$$

Taking the values of the functions in the initial state we get:

$$\rho^n v^*(q_0; 0) \leq V_n.$$

Hence, by the previous lemma, denoting the positive number $v^*(q_0; 0)$ by δ :

$$\rho^n \delta \leq V_n.$$

Taking logarithm and dividing by n we obtain

$$\log \rho + \log \delta/n \leq \log V_n/n.$$

The limit of the left-hand side is $\log \rho$. We conclude that

$$\log \rho \leq \liminf_{n \rightarrow \infty} \log V_n/n = \mathcal{H}.$$

□

The following result is immediate from the proof of the Theorem.

Corollary 1. *For any nice timed automaton \mathcal{A} the lim sup in the definition of the entropy is in fact a limit, that is $\mathcal{H} = \lim_{n \rightarrow \infty} \log V_n/n$.*

4 Computing the Entropy

The characterization of \mathcal{H} in Theorem 2 solves the main problem explored in this paper, but its concrete application requires computing the spectral radius of an integral operator Ψ , and this is not straightforward. In 4.1 we solve this problem for a subclass of automata by reduction to differential equations. As for the general situation, in 4.3 we give an iterative procedure which approximates the spectral radius and the entropy with a guaranteed precision.

4.1 Case of “ $1\frac{1}{2}$ Clock” Automata

Consider now the subclass of (nice fleshy region-split) automata with entry regions of all the locations having dimension 0 or 1. In other words, in such automata every discrete transition resets every clock but one (or every clock). We call this class $1\frac{1}{2}$ clock automata.

For $1\frac{1}{2}$ clock automata, the operator Ψ can be written more explicitly. Let us call x the clock that is not reset when coming into a new location. Now \mathcal{F} is the space of functions f mapping every $(q, x) \in Q \times (0, 1)$ to $f_q(x) \in \mathbb{R}$ which, by a slight abuse of notation, we will assimilate to the space $(0, 1) \rightarrow \mathbb{R}^{|Q|}$ (where f stands for the vector $(f_q)_{q \in Q}$).

We can observe that from the entry region of every state, regions of two types are alternatively visited: regions where x is greater than the other clocks, and regions where it is the contrary. For guards \mathbf{g} that are regions of the first type (a), $x + \tau \in \mathbf{g}$ is equivalent to $\tau \in (0, 1 - x)$, and for the other type (b), to $\tau \in (1 - x, 1)$.

Furthermore, the reset function \mathbf{r} can behave in three different ways: either it resets every clock but one that is not x (1), or it resets every clock but x (2), or it resets every clock (3).

Those two criteria can be combined in 6 different ways, partitioning the set of transitions starting from q in as many sets: $\Delta_{qa1}, \Delta_{qb1}, \Delta_{qa2}, \Delta_{qb2}, \Delta_{qa3}$ and Δ_{qb3} , such that Ψ can now be written the following way:

$$\begin{aligned} \Psi f(q, x) = & \sum_{\delta \in \Delta_{qa1}} \int_0^{1-x} f(q', x + \tau) d\tau & + & \sum_{\delta \in \Delta_{qb1}} \int_{-x}^0 f(q', x + \tau) d\tau \\ & + \sum_{\delta \in \Delta_{qa2}} \int_0^{1-x} f(q', \tau) d\tau & + & \sum_{\delta \in \Delta_{qb2}} \int_{1-x}^1 f(q', \tau) d\tau \\ & + \sum_{\delta \in \Delta_{qa3}} (1-x)f(q', 0) & + & \sum_{\delta \in \Delta_{qb3}} xf(q', 0). \end{aligned}$$

Now we define the square matrices D_{ij} such that the operator can be written as follows:

$$\begin{aligned} \Psi f(x) = & D_{a1} \int_0^{1-x} f(x + \tau) d\tau + D_{b1} \int_{-x}^0 f(x + \tau) d\tau \\ & + D_{a2} \int_0^{1-x} f(\tau) d\tau + D_{b2} \int_{1-x}^1 f(\tau) d\tau \\ & + D_{a3}(1-x)f(0) + D_{b3}xf(0). \end{aligned}$$

This is the explicit formula for Ψ we had been looking for. Now, computing the entropy of the language of the automaton using Thm. 2 involves finding the leading eigenvalue of Ψ , that is the greatest $\lambda \in \mathbb{R}$ such that for some function $f \in \mathcal{F}$:

$$\Psi f = \lambda f \quad (7)$$

We will solve this by transforming this equality into a differential equation. For this, first we differentiate the previous expression once, yielding:

$$\lambda f'(x) = (D_{b_1} - D_{a_1})f(x) + (D_{b_2} - D_{a_2})f(1-x) + (D_{b_3} - D_{a_3})f(0). \quad (8)$$

Before we continue, we can remark that both (7) and (8) are homogeneous equations. Hence their solution sets are vector spaces. As the first equation implies the second, its solution space is a subspace of the differentiated equation.

Now take a solution f^* of the second system, replace f by f^* in the system, and integrate it. We find an equation which is the equivalent to the integral equation, up to the translation of one member by a constant vector. Thus, if by replacing f by f^* , the integral equation holds for $x = 0$, it also holds for all $x \in [0, 1]$.

This means that the solutions to the first equation are exactly those of the second for which the integral equation holds at $x = 0$:

$$\lambda f(0) = (D_{a_1} + D_{a_2}) \int_0^1 f(\tau) d\tau + D_{a_3} f(0). \quad (9)$$

Now we solve the second equation by introducing the functions u and w as follows: $u(x) = f(x) + f(1-x)$ and $w(x) = f(x) - f(1-x)$. This removes the cumbersome dependency between f and $f(1-\cdot)$ and enables us to rewrite the previous equation as a differential system:

$$\begin{cases} \lambda u'(x) = & Aw(x) \\ \lambda w'(x) = & Bu(x) + C(u(0) + w(0)), \\ w(\frac{1}{2}) = & 0 \end{cases}, \quad (10)$$

where $A \triangleq D_{b_1} - D_{a_1} - D_{b_2} + D_{a_2}$, $B \triangleq D_{b_1} - D_{a_1} + D_{b_2} - D_{a_2}$ and $C \triangleq D_{b_3} - D_{a_3}$.

Note that due to the properties of the functions u and w , this system has to be considered on the interval $[0, \frac{1}{2}]$ only, and $w(\frac{1}{2}) = 0$ is the consequence of the definition of w . This rewriting is without loss of information, as the original equation (8) can be recovered by adding those two equations term by term.

System (10) implies:

$$\begin{cases} \lambda^2 w''(x) = BAw(x) \\ \lambda u'(x) = Aw(x) \\ w(\frac{1}{2}) = 0. \end{cases} \quad (11)$$

The first equation of (11) is homogeneous and has a solution space of dimension $2n$, but using the fact that $w(\frac{1}{2}) = 0$, this allows us to consider only n independent solutions w_i .

Using the second equation, we get $u(x) = \frac{1}{\lambda} \int_0^x Aw(t)dt + u_0$, for every solution w of the first equation and every $u_0 \in \mathbb{R}^n$. Thus (11) yields a solution space of dimension $2n$.

Now having a solution (u, w) to (11) implies that $\lambda^2 w''(x) = BAw(x)$, which implies $\lambda w'(x) = \frac{1}{\lambda} \int_0^x BAw(x) + \lambda w'(0)$ and thus $\lambda w'(x) = \int_0^x Bu'(x) + w'(0) = Bu(x) + Bu(0) + \lambda w'(0)$

Therefore (u, w) is also a solution to (10) if and only if $C(u(0) + w(0)) = Bu(0) + \lambda w'(0)$. Coming back to (8), $f \triangleq \frac{u+w}{2}$ is solution to this system if and only if

$$\lambda(f(0) - f(1)) = 2Cf(0) - B(f(0) + f(1)),$$

or again

$$(\lambda + B - 2C)f(0) = (\lambda - B)f(1). \quad (12)$$

Now to sum up, the space of the solutions of (11) is of dimension $2n$, thus so is the space S of the functions $f = \frac{u+w}{2}$ such that (u, w) is solution to (11). This allows us to write every such f like this: FM where F is a $n \times 2n$ matrix whose columns are a basis of S , and M is a vertical vector of \mathbb{R}^{2n} .

Every such $f = FM$ is solution of (7) if and only if both (9) and (12) hold, which we rewrite here, replacing f by FM :

$$\begin{cases} \lambda F(0)M = ((D_{a1} + D_{a2})(\int_0^1 F(\tau)d\tau) + D_{a3}F(0))M \\ (\lambda + B - 2C)F(0)M = (\lambda - B)F(1)M \\ (F(0) - ((D_{a1} + D_{a2})(\int_0^1 F(\tau)d\tau) + D_{a3}F(0)))M = 0 \\ ((\lambda + B - 2C)F(0) - (\lambda - B)F(1))M = 0. \end{cases}$$

Considering this as an equation on M , this homogeneous linear system has non-zero solutions if and only if

$$\det \begin{pmatrix} F(0) - ((D_{a1} + D_{a2})(\int_0^1 F(\tau)d\tau) + D_{a3}F(0)) \\ (\lambda + B - 2C)F(0) - (\lambda - B)F(1) \end{pmatrix} = 0$$

This is a transcendental equation on λ (as $F(x)$ has coefficients which are polynomials of complex exponentials of $\frac{x}{\lambda}$) that can be solved numerically, and which we know to have a maximal real solution, which is also the spectral radius of Ψ (Lem. 1). The logarithm of this value is the entropy of the language (Thm. 2).

Now we can sum up all those computations and present the whole algorithm for automata with $1\frac{1}{2}$ clocks in Table 1.

4.2 Application to the Running Example

We propose to compute the entropy of the language of the third automaton of Fig. 1. Let us remark that this automaton is a “ $1\frac{1}{2}$ clocks” and we can use the method we just described.

| |
|---|
| <p>1. Transform \mathcal{A} into the fleshy region-split form and check that it has $1\frac{1}{2}$ clock.</p> <p>2. Compute the matrices D_{ij} as follows: for p and q two locations, $D_{ij_{pq}}$ is the number of transitions $\delta = (p, a, \mathbf{g}, \tau, q) \in \Delta$ such that</p> <ul style="list-style-type: none"> – $i = a$ and \mathbf{g} is a region where x is the greatest clock, – or $i = b$ and \mathbf{g} is a region where x is not the greatest clock <p>and</p> <ul style="list-style-type: none"> – $i = 1$ and τ resets every clock but x, – or $i = 2$ and τ resets every clock but one that is not x, – or $i = 3$ and τ resets every clock. <p>3. Compute $A \triangleq D_{b1} - D_{a1} - D_{b2} + D_{a2}$, $B \triangleq D_{b1} - D_{a1} + D_{b2} - D_{a2}$ and $C \triangleq D_{b3} - D_{a3}$.</p> <p>4. Compute the general solution (u, w) to (11).</p> <p>5. Deduce the general solution FM to (8).</p> <p>6. Find the greatest root ρ of</p> $\det \begin{pmatrix} F(0) - ((D_{a1} + D_{a2})(\int_0^1 F(\tau)d\tau) + \Delta_{a3}F(0)) \\ (\lambda + B - 2C)F(0) - (\lambda - B)F(1) \end{pmatrix}$ <p>where the unknown is λ.</p> <p>7. Then we have $\mathcal{H}(L(\mathcal{A})) = \log \rho$.</p> |
|---|

Table 1. Symbolic algorithm: computing \mathcal{H} for $1\frac{1}{2}$ clocks

The region-split form of this automaton is presented on Fig. 3.

As two non-initial states actually have the same role, the volume of a path of length $n \in \mathbb{N}$ is the same function v_n in both states.

Thus v_n is characterized by:

$$\begin{cases} v_0(x) = 1 \\ \forall n \in \mathbb{N}, v_{n+1}(x) = (\Psi v_n)(x) \triangleq \int_0^{1-x} v_n(t) dt. \end{cases}$$

Using Lem. 1, we know that $\rho(\Psi)$ is an eigenvalue and Thm. 2 says its logarithm is also the entropy of the language. Let us compute $\max\{\lambda \in \mathbb{R} \mid \lambda v = \Psi v\}$.

$$\lambda v(x) = \int_0^{1-x} v(t) dt \tag{13}$$

$$\text{implies} \quad \lambda v'(x) = -v(1-x) \tag{14}$$

$$\text{implies} \quad \lambda^2 v''(x) = -v(x) \tag{15}$$

We know the solutions have the form $v(x) = \alpha \sin(\frac{x}{\lambda}) + \beta \cos(\frac{x}{\lambda})$. Using (13) we find $v(1) = 0$. We inject this in (14) for $x = 0$ and deduce $\alpha = 0$. Thus $v(x) = \beta \cos(\frac{x}{\lambda})$ and $\cos(\frac{1}{\lambda}) = 0$. This implies that every solution can be written $\lambda = \frac{2}{(2k+1)\pi}$ for some $k \in \mathbb{Z}$. The highest of those is $\lambda = \frac{2}{\pi}$, and we can verify

that $v(x) = \cos(\frac{x\pi}{2})$ satisfies $\frac{2}{\pi}v = \Psi v$. Therefore the entropy of this automaton is $\log \frac{2}{\pi}$.

4.3 General Case

In a higher dimension it would be difficult to find the eigenfunction symbolically. Instead we can use standard procedures for eigenvalue approximation for positive operators.

Recall that the volume function satisfies the following equation

$$v_n = \Psi^n 1. \quad (16)$$

The following theorem is close to theorems 16.1-16.2 from [3].

Theorem 3. *If for some $\alpha, \beta \in \mathbb{R}, m \in \mathbb{N}$ the following inequality holds:*

$$\alpha v_m \leq v_{m+1} \leq \beta v_m, \quad (17)$$

and for the same m the volume $V_m = v_m(q_0, 0) > 0$, then

$$\log \alpha \leq \mathcal{H} \leq \log \beta.$$

Proof (sketch). Applying the positive operator Ψ^n to the inequalities (17), and using the formula (16) we obtain that for all n

$$\alpha v_{m+n} \leq v_{m+n+1} \leq \beta v_{m+n}.$$

From this by induction we prove that for all n

$$\alpha^n v_m \leq v_{m+n} \leq \beta^n v_m.$$

We apply this to the initial state $(q_0, 0)$ (remember that $V_n = v_n(q_0, 0)$):

$$\alpha^n V_m \leq V_{m+n} \leq \beta^n V_m.$$

Take a logarithm, divide by $m + n$ and take a $\limsup_{n \rightarrow \infty}$ (remember that $\mathcal{H} = \limsup_{n \rightarrow \infty} \log V_n/n$):

$$\log \alpha \leq \mathcal{H} \leq \log \beta$$

(we have used the fact that $V_m > 0$). □

This theorem yields the following procedure to estimate \mathcal{H} summarized in Table 2.

One possible optimization is to compute α and β separately on every strongly connected reachable component of the region graph, and take the maximal values.

- | |
|---|
| <ol style="list-style-type: none"> 1. Transform \mathcal{A} into the fleshy region-split form. 2. Choose an m and compute symbolically the piecewise polynomial functions v_m and v_{m+1} 3. Check that $v_m(q_0, 0) > 0$. 4. Compute $\alpha = \min(v_{m+1}/v_m)$ and $\beta = \max(v_{m+1}/v_m)$. 5. Conclude that $\mathcal{H} \in [\log \alpha; \log \beta]$. |
|---|

Table 2. Iterative algorithm: bounding \mathcal{H}

| 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 | $\frac{61}{720}(1-x)-\frac{1}{5040}(1-x)^7+\frac{1}{240}(1-x)^5-\frac{1}{144}(1-x)^3$ | 0.6364 | 0.6371 | -0.6518 | -0.6506 |
| 8 | $\frac{17}{315}(1-x)+\frac{1}{40320}(1-x)^8-\frac{1}{5040}(1-x)^7+\frac{1}{360}(1-x)^5-1/45(1-x)^3$ | | | | |

Table 3. Iterating the operator for \mathcal{A}_3 ($\mathcal{H} = \log(2/\pi) \approx \log 0.6366 \approx -0.6515$)

4.4 Example: Again \mathcal{A}_3

We apply the iterative procedure above to our running example \mathcal{A}_3 . As explained in Sect. 4.2, we can just consider the operator on $C(0;1)$

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

The iteration results are given in Table 3.

5 Discretization Approach

5.1 Discretizing the Volumes

Another approach to volume/entropy computation is by discretization. This approach sheds also a new light on the information-theoretic interpretation of entropy. The discretizations of timed languages we use are strongly inspired by [7, 8].

To use the discretization approach we do not put \mathcal{A} in the region-split form. Take an $\varepsilon = 1/N > 0$. A timed word w is ε -timed if all the delays in this word are multiples of ε . Any ε -timed word w over an alphabet Σ can be written as $w = h_\varepsilon(v)$ for an untimed $v \in \Sigma \cup \{\tau\}$, where the morphism h_ε is defined as follows:

$$h_\varepsilon(a) = a \text{ for } a \in \Sigma, \quad h_\varepsilon(\tau) = \varepsilon.$$

The discrete word v with ticks τ (standing for ε delays) represents in this way the ε -timed word w .

Example Let $\varepsilon = 1/5$, then the timed word $0.6a0.4ba0.2a$ is ε -timed. Its representation is $\tau\tau\tau a\tau\tau b a\tau a$.

The notions of ε -timed words and their representation can be ported straightforwardly to languages.

For a timed word $w = t_1 a_1 t_2 a_2 \dots t_n a_n$ we introduce its North-East ε -neighbourhood like this:

$$\mathcal{B}_\varepsilon^{NE}(w) = \{s_1 a_1 s_2 a_2 \dots s_n a_n \mid \forall i (s_i \in [t_i; t_i + \varepsilon])\}.$$

Notice that for a timed word w of a size n the set $\mathcal{B}_\varepsilon^{NE}(w)$ is a hypercube of edge ε (in the delay space), and of volume ε^n . Notice also, that neighbourhoods of different ε -timed words are almost disjoint: the interior of their intersections are empty.

Suppose now that we have a timed language L recognized by a nice timed automaton \mathcal{A} and we want to compute its entropy (or just the volumes V_n). Take an $\varepsilon = 1/N > 0$. We consider two ε -timed languages L_+ and L_- such that

$$\mathcal{B}_\varepsilon^{NE}(L_-) \subset L \subset \mathcal{B}_\varepsilon^{NE}(L_+), \quad (18)$$

which gives, for the volumes

$$\#L_-(n) \cdot \varepsilon^n \leq \text{Vol}(L(n)) \leq \#L_+(n) \cdot \varepsilon^n.$$

In order to describe suitable L_+ and L_- and to compute the cardinalities we will describe (untimed) automata over the alphabet $\Sigma \cup \tau$ recognizing their representations.

The recipe is like this. Take the nice timed automaton \mathcal{A} accepting L . Discrete automata A_+^ε and A_-^ε can be constructed in two stages. First we build counter automata C_+^ε and C_-^ε . They have the same states as \mathcal{A} , but instead of every clock x they have a counter c_x (roughly representing x/ε). For every state add a self-loop labelled by τ and incrementing all the counters. Replace any reset of x by a reset of c_x . Whenever \mathcal{A} has a guard $x \in [l; u]$ (or $x \in (l; u)$, or some other interval), the counter automaton C_+^ε has a guard $c_x \in [l/\varepsilon - D; u/\varepsilon - 1]$ (always the closed interval) instead, where D is as in assumption A4. At the same time C_-^ε has a guard $c_x \in [l/\varepsilon; u/\varepsilon - D]$. Automata C_+^ε and C_-^ε with bounded counters can be easily transformed into finite-state ones A_+^ε and A_-^ε .

Lemma 3. *Languages $L_+ = h_\varepsilon(L(A_+^\varepsilon))$ and $L_- = h_\varepsilon(L(A_-^\varepsilon))$ have the required property (18).*

Proof (sketch).

Inclusion $\mathcal{B}_\varepsilon^{NE}(L_-) \subset L$. Let a discrete word $u \in L(A_-)$, let $v = h_\varepsilon(u)$ be its ε -timed version, and let $w \in \mathcal{B}_\varepsilon^{NE}(v)$. We have to prove that $w \in L$. Notice first that $L(A_-) = L(C_-)$ and hence u is accepted by C_- . Mimic the run of C_- on u , but replace every τ by an ε duration, thus one can obtain a run of \mathcal{A} on v . Moreover, in this run every guard $x \in [l, u]$ is respected with a security margin: in fact a stronger guard $x \in [l, u - D\varepsilon]$ is respected. Now one

can mimic the same run of \mathcal{A} on w . By definition of the neighbourhood, for any delay t_i in u the corresponding delay t'_i in w belongs to $[t_i, t_{i+1}]$. Clock values are always sums of several (up to D) consecutive delays. Whenever a narrow guard $x \in [l, u - D\varepsilon]$ is respected on v , its “normal” version $x' \in [l, u]$ is respected on w . Hence, the run of \mathcal{A} on w obtained in this way respects all the guards, and thus \mathcal{A} accepts w . We deduce that $w \in L$. \square

Inclusion $L \subset \mathcal{B}_\varepsilon^{NE}(L_+)$. First we define an approximation function on \mathbb{R}_+ as follows:

$$\underline{t} = \begin{cases} 0 & \text{if } t = 0 \\ t - \varepsilon & \text{if } t/\varepsilon \in \mathbb{N}_+ \\ \varepsilon \lfloor t/\varepsilon \rfloor & \text{otherwise.} \end{cases}$$

Clearly, \underline{t} is always a multiple of ε and belongs to $[t - \varepsilon, t)$ with the only exception that $\underline{0} = 0$.

Now we can proceed with the proof. Let $w = t_1 a_1 \dots t_n a_n \in L$. We define its ε -timed approximation v by approximating all the delays: $v = \underline{t}_1 a_1 \dots \underline{t}_n a_n$. By construction $w \in \mathcal{B}_\varepsilon^{NE}(v)$. The run of \mathcal{A} on w respects all the guards $x \in [l; u]$. Notice that the clock value of x on this run is a sum of several (up to D) consecutive t_i . If we try to run \mathcal{A} over the approximating word v , the value x' of the same clock at the same transition would be a multiple of ε and it would belong to $[x - D\varepsilon; x)$. Hence $x' \in [l - D\varepsilon, u - \varepsilon]$. By definition of C_+ this means that the word $u = h_\varepsilon^{-1}(v)$ is accepted by this counter automaton. Hence $v \in L_+$.

Let us summarize: for any $w \in L$ we have constructed $v \in L_+$ such that $w \in \mathcal{B}_\varepsilon^{NE}(v)$. This concludes the proof. \square

5.2 Counting Discrete Words

Once the automata A_+^ε and A_-^ε constructed, we have to count the number of words with n events and its asymptotic behaviour. We proceed in three stages. First we remove useless states (unreachable from the initial state). Next, we eliminate the tick transitions τ . As we are counting paths, we obtain an automaton without silent transitions, but with multiplicities representing the number of realizations of every transition.

Let M_- and M_+ be the adjacency matrices with multiplicities of the automaton obtained. It is well-known that the $\#L_-(n)$ (or $\#L_+(n)$) can be found as sums of the first line of the matrix M_-^n (or M_+^n). This gives lower and upper bounds on the volume V_n . Moreover, using Perron-Frobenius theorem we obtain that $\#L_-(n) \sim \lambda_-^n$ where λ_- is the spectral radius of M_- and the same is true for the upper bounds.

Theorem 4. *For a nice timed automaton \mathcal{A} , the entropy of its language satisfies the estimates:*

$$\log_2(\varepsilon\lambda_-) \leq \mathcal{H}(L(\mathcal{A})) \leq \log_2(\varepsilon\lambda_+).$$

| |
|---|
| <ol style="list-style-type: none"> 1. Choose an $\varepsilon = 1/N$. 2. Build the counter automata C_-^ε and C_+^ε. 3. Transform them into finite automata A_-^ε and A_+^ε. 4. Eliminate τ transitions introducing multiplicities. 5. Obtain adjacency matrices M_- and M_+. 6. Compute their spectral radii ρ_- and ρ_+. 7. Conclude that $\mathcal{H} \in [\log \rho_-; \log \rho_+]$. |
|---|

Table 4. Discretization algorithm: bounding \mathcal{H}

We summarize the algorithm in Table 4.

This theorem can be used to estimate the entropy. But it can also be read in a converse direction: the cardinality of L restricted to n events and discretized with quantum ε is close to $2^{\mathcal{H}n}/\varepsilon^n$. Hence we can encode $\mathcal{H} - \log \varepsilon$ bits of information per event. These information-theoretic considerations are made more explicit in Sect. 5.3 below.

A Case Study Consider again our running example $L_3 = \{t_1at_2bt_3at_4b \dots \mid t_i + t_{i+1} \in [0; 1]\}$. We need two clocks to recognize this language, and they are never reset together. We choose $\varepsilon = 0.05$ and build the automata on Fig. 4 according to the recipe (the discrete ones A_+ and A_- are too big to fit on the figure).

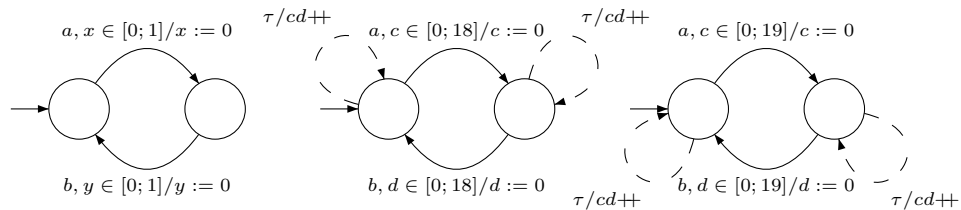


Fig. 4. A two-clock timed automaton \mathcal{A}_3 and its approximations $C_-^{0.05}$ and $C_+^{0.05}$

We transform $C_-^{0.05}$ and $C_+^{0.05}$, into A_+ and A_- , eliminate silent transitions and unreachable states, and compute spectral radii of adjacency matrices (their sizes are 38x38 and 40x40): $\#L_-^{0.05}(n) \sim 12.41^n$, $\#L_+^{0.05}(n) \sim 13.05^n$. Hence $12.41^n \cdot 0.05^n \leq V_n \leq 13.05^n \cdot 0.05^n$, and the entropy

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

Taking a smaller $\varepsilon = 0.01$ provides a better estimate for the entropy:

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

We recall that the true value of the entropy is $\mathcal{H} = \log(2/\pi) \approx \log 0.6366 \approx -0.6515$.

5.3 Kolmogorov complexity of timed words

To interpret the results above in terms of information content of timed words we state some estimates of Kolmogorov complexity of timed words. Remark first that a timed word within a “simple” timed language can involve rational delays of a very high complexity, or even uncomputable real delays. However they always have approximations of a bounded complexity. This observation motivates the following two results.

Theorem 5. *Let \mathcal{A} be a nice timed automaton, L its language, \mathcal{H} its entropy. For any rational $\varepsilon > 0$, any $\alpha > 0$ and any $n \in \mathbb{N}$ large enough there exists a timed word $w \in L$ of size n such that for any rational timed word v such that $w \in \mathcal{B}_\varepsilon^{NE}(v)$, the Kolmogorov complexity of v is lower bounded as follows*

$$K(v|n, \varepsilon) \geq n(\mathcal{H} + \log 1/\varepsilon - \alpha).$$

Proof. By definition of the entropy, for n large enough

$$V_n > 2^{n(\mathcal{H}-\alpha)}.$$

Consider the set S of all timed words v with a smaller complexity:

$$S = \{v \mid K(v|n, \varepsilon) \leq n(\mathcal{H} + \log(1/\varepsilon) - \alpha)\}.$$

The cardinality of S can be bounded as follows:

$$\#S \leq 2^{n(\mathcal{H} + \log(1/\varepsilon) - \alpha)} = 2^{n(\mathcal{H}-\alpha)} / \varepsilon^n.$$

The ε -NE-neighborhood of S is a union of $\#S$ cubes of side ε . Hence

$$\text{Vol}(\mathcal{B}_\varepsilon^{NE}(S)) \leq \varepsilon^n \#S \leq 2^{n(\mathcal{H}-\alpha)} < V_n.$$

We deduce that the set L_n of timed words from L of size n cannot be included into $\mathcal{B}_\varepsilon^{NE}(S)$. Thus, there exists a word $w \in L_n \setminus \mathcal{B}_\varepsilon^{NE}(S)$. By construction, it cannot be approximated by any low-complexity word with precision ε . \square

Theorem 6. *Let \mathcal{A} be a nice timed automaton, L its language, $\alpha > 0$ a rational number. Consider a “bloated” automaton \mathcal{A}' which is like \mathcal{A} , but in all the guards each constraint $x \in [L, U]$ is replaced by $x \in [L - \alpha, U + \alpha]$. Let \mathcal{H}' the entropy of its language. Then the following holds for any $\varepsilon = 1/N \in (0; \alpha/D)$, and any n large enough.*

For any timed word $w \in L$ of size n there exists a rational and ε -timed word v such that $w \in \mathcal{B}_\varepsilon^{NE}(v)$ and its Kolmogorov complexity is upper bounded as follows:

$$K(v|n, \varepsilon) \leq n(\mathcal{H}' + \log 1/\varepsilon + \alpha).$$

Proof. Denote the language of \mathcal{A}' by L' , the set of words of length n in this language by L'_n and its n -volume by V'_n . We remark that for n large enough

$$V'_n < 2^{n(\mathcal{H}' + \alpha/2)}.$$

Let now $w = t_1 a_1 \dots t_n a_n$ in L_n . We construct its rational ε -approximation as in Lemma 3: $v = \underline{t}_1 a_1 \dots \underline{t}_n a_n$. To find an upper bound for the complexity of v we notice that $v \in \bar{V}$, where U is the set of all ε -timed words u of n letters such that $\mathcal{B}_\varepsilon^{NE}(u) \subset L'_n$. Since these balls are all of volume ε^n and that they have disjoint interiors, the cardinality of U admits the upper bound:

$$\#U \leq V'_n / \varepsilon^n < 2^{n(\mathcal{H}' + \alpha/2)} / \varepsilon^n.$$

Hence, in order to encode v (knowing n and ε) it suffice to give its number in a lexicographical order of U , and

$$K(v|n, \varepsilon) \leq \log \#U + c \leq n(\mathcal{H}' + \log 1/\varepsilon + \alpha/2) + c \leq n(\mathcal{H}' + \log 1/\varepsilon + \alpha)$$

for n large enough. □

We believe that these results can be refined further, but even in its current form they show clearly, that the maximal information content of timed words in a language L , observed (or generated) with a finite precision ε , is close to $\mathcal{H} + \log(1/\varepsilon)$ bits/event.

6 Conclusions and Further Work

In this paper, we have defined size characteristics of timed languages: volume and entropy. Exact formulas for volumes have been given. As for the entropy, it has been characterized as logarithm of the leading eigenvalue of a positive operator on the space of continuous functions on a part of the state space. Several procedures have been suggested to compute it.

Research in this direction is very recent, and many questions need to be studied. We are planning to explore practical feasibility of the procedures described here. We believe that, as usual for timed automata, they should be transposed from regions to zones. We will explore potential applications mentioned in the introduction.

Many theoretical questions still require exploration. It would be interesting to remove some of Assumptions A1-A4, especially to compute entropy of non-deterministic timed automata. Kolmogorov complexity estimates can be improved. Extending results to probabilistic timed automata is another option. Our entropy represents the amount of information per timed event. It would be interesting to find the amount of information per time unit.

Acknowledgment

The authors are thankful to Oded Maler for motivating discussions and valuable comments on the manuscript.

References

1. Ben Salah, R., Bozga, M., Maler, O.: On timed components and their abstraction. In: SAVCBS'07, ACM (2007) 63–71
2. Lind, D., Marcus, B.: An introduction to symbolic dynamics and coding. Cambridge University Press (1995)
3. Krasnosel'skij, M., Lifshits, E., Sobolev, A.: Positive Linear Systems: The method of positive operators. Number 5 in Sigma Series in Applied Mathematics. Heldermann Verlag, Berlin (1989)
4. Bucci, G., Piovosi, R., Sassoli, L., Vicario, E.: Introducing probability within state class analysis of dense-time-dependent systems. In: QEST'05, IEEE Computer Society (2005) 13–22
5. Asarin, E., Caspi, P., Maler, O.: Timed regular expressions. *Journal of the ACM* **49** (2002) 172–206
6. Alur, R., Dill, D.L.: A theory of timed automata. *Theoretical Computer Science* **126** (1994) 183–235
7. Asarin, E., Maler, O., Pnueli, A.: On discretization of delays in timed automata and digital circuits. In: CONCUR'98. LNCS 1466, Springer-Verlag (1998) 470–484
8. Henzinger, T.A., Manna, Z., Pnueli, A.: What good are digital clocks? In: ICALP'92. LNCS 623, Springer-Verlag (1992) 545–558