# Volume and entropy of regular timed languages

Eugene Asarin, Aldric Degorre

# Volume and entropy of regular timed languages

Eugene Asarin[1] and Aldric Degorre[2]

[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 size measures: volume for languages with 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 quantitative comparison of languages, and the entropy can be viewed as information contents of a timed language. For languages accepted by 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 symbolical one for so-called "$1\frac{1}{2}$-clock" automata, and two numerical ones: one using techniques of functional analysis, another based on discretization. We give an information-theoretic interpretation of the entropy in terms of Kolmogorov complexity.

## 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. For example, $2a3.11b$ means that an event $a$ happened 2 time units after the system start, and $b$ happened 3.11 time units after $a$. A timed language, which is just a set of timed words, 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 $\varepsilon$, if the delays are measured with a finite precision $\epsilon$, 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. 7-8 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 the set of all behaviours violating a property or of a subset of those presented as a counter-example by a verification tool.

In this paper, we explore, and partly solve the following problems: given a prefix-closed timed language accepted by a 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 language.

**Related Work.** Our problems and techniques are inspired by works concerning the entropy of finite-state languages (cf. [2]). There the cardinality of the set $L_n$ of all elements of length $n$ of a prefix-closed regular language 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 $\mathcal{A}$.[3] The main technical tool used to compute the entropy of finite automata is the Perron-Frobenius theory for positive matrices, and, in this paper, in a first approach we will use its extensions to infinite-dimensional operators [3]. In a second approach, we also propose to reduce our problem by discretization to entropy computation for some discrete automata.

In [4, 5] 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 these results to uniform measures. As for unbounded languages, they use 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 (every clock to be reset at most every $D$ steps, but these resets need not be simultaneous), and we obtain only the entropy.

In [6] probabilities of LTL properties of one-clock timed automata (over infinite timed words) are computed using Markov chains techniques. It would be interesting to try to adapt our methods to this kind of problems.

Last, our studies of Kolmogorov complexity of rational elements of timed languages, relating this complexity to the entropy of the language, remind of

[3] This holds also for automata with multiplicities, see [2].

earlier works on complexity of rational approximations of continuous functions [7, 8], and those relating complexity of trajectories to the entropy of dynamical systems [9, 8].

**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 that 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. 7. 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 length $n$ over an alphabet $\Sigma$ is a sequence $w = t_1a_1t_2 \dots t_na_n$, where $a_i \in \Sigma, t_i \in \mathbb{R}$ and $0 \le t_i$ (notice that this definition rules out timed words ending by a time delay). Here $t_i$ represents the delay between the events $a_{i-1}$ and $a_i$. With such a timed word $w$ of length $n$ 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:

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

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 length $n$ the volumes of the 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 associate with every timed language a sequence of $n$-volumes $V_n$. We will show in Sect. 2.5 that, for languages of deterministic 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 $n \to \infty$. Typically, $V_n$ depends approximately exponentially on $n$. We define the entropy of a language as the rate of this dependence.

Formally, for a timed language $L$ we define its *entropy* as follows[4] (all logarithms in the paper are base 2):

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

*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 $0 \le d_1 \le \cdots \le d_n$, 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. Therefore, choosing date ($d_i$) or delay ($t_i$) representation has no influence on language volumes (and entropy). Due to the authors' preferences (justified in [10]), delays will be used in the sequel.

### 2.2 Three Examples

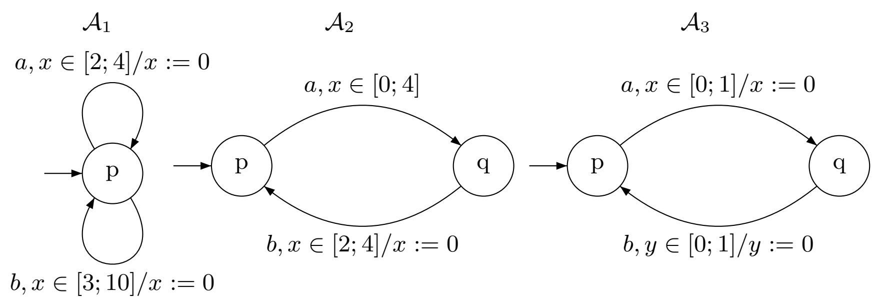


**Fig. 1.** Three simple timed automata $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$

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, it will be used to illustrate more advanced methods throughout the paper.

**Rectangles.** Consider the timed language defined by the expression

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

[4] In fact, due to Assumption A2 below, the languages we consider in the paper are prefix-closed, and lim sup is in fact a lim. This will be stated formally in Cor. 1.

recognized by $\mathcal{A}_1$ of Fig. 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 $2 \le t_i + s_i \le 4$. Since we want prefix-closed languages, the last $s_k b$ can be omitted.

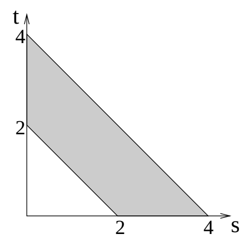


**Fig. 2.** Timings $(t_i, s_i)$ for $\mathcal{A}_2$.

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 $2 \le t_i + s_i \le 4$. 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 $0 \le t_{k+1} \le 4$, 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. An *ad hoc* 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 is

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

This gives the sequence of volumes:

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

In the sequel, we will also compute the entropy of this language.

### 2.3 Subclasses of Timed Automata

In the rest of the paper, we compute volumes and entropy for regular timed languages recognized by some subclasses of timed automata (TA). We assume that the reader is acquainted with timed automata; otherwise, we refer her or him to [11] for details. Here we only fix notations for components of timed automata and state several requirements they 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 do not need to specify accepting states due to A2 below, neither we need any invariants). A generic state of $\mathcal{A}$ is a pair $(q, \mathbf{x})$ of a control location and a vector of clock values. A generic element of $\Delta$ is written as $\delta = (q, a, \mathfrak{g}, \mathfrak{r}, q')$ meaning a transition from $q$ to $q'$ with label $a$, guard $\mathfrak{g}$ and reset $\mathfrak{r}$. We spare the reader the definitions of a run of $\mathcal{A}$ and of its accepted language.

Several combinations of the following Assumptions will be used in the sequel:

- A1. The automaton $\mathcal{A}$ is deterministic[5].
- A2. All its states are accepting.
- A3. Guards are rectangular (i.e. conjunctions of constraints $L_i \leq x_i \leq U_i$, strict inequalities are also allowed). Every guard upper bounds at least one clock.
- A4. There exists a $D \in \mathbb{N}$ such that on every run segment of $D$ transitions, every clock is reset at least once.
- A5. There is no punctual guards, that is in any guard $L_i < U_i$.

Below we motivate and justify these choices:

- A1: Most of known techniques to compute entropy of untimed 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. R. Lanotte pointed out to the authors that any TA satisfying A4 can be determinized.
- 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.

[5] That is any two transitions with the same source and the same label have their guards disjoint.

A4: We use this variant of non-Zenoness condition several times in our proofs and constructions. As the automaton of Fig. 3 shows, if we omit this assumption some anomalies can occur.

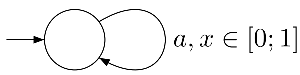


**Fig. 3.** An automaton without resets

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 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.

A5: While assumptions A1-A4 can be restrictive, we always can remove the transitions with punctual guards from any automaton, without changing the volumes $V_n$. Hence, A5 is not restrictive at all, as far as volumes are considered. In Sect. 6 we do not make this assumption.

### 2.4 Preprocessing Timed Automata

In order to compute volumes $V_n$ and entropy $\mathcal{H}$ of the language of a nice TA, we first transform this automaton into a normal form, which can be considered as a (timed) variant of the region graph, the quotient of the TA by the region equivalence relation defined in [11].

We say that a TA $\mathcal{A} = (Q, \Sigma, C, \delta, q_0)$ is in a *region-split form* if A1, A2, A4 and 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 the set of clock values with which $q$ is entered is exactly $\mathbf{r}_q$. For the initial location $q_0$, its entry region is the singleton $\{0\}$.
B3. The guard $\mathfrak{g}$ of every transition $\delta = (q, a, \mathfrak{g}, \mathfrak{r}, q') \in \Delta$ is just one clock region.

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

**Proposition 1.** *Given a nice TA $\mathcal{A}$, a region-split TA $\mathcal{A}'$ accepting the same language can be constructed*[6].

[6] 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.

*Proof (sketch).* Let $\mathcal{A} = (Q, \Sigma, C, \Delta, q_0)$ be a nice TA and let **Reg** be 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 $\delta = (q, a, \mathfrak{g}, \mathfrak{r}, q')$ 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 $\mathbf{r}' \subset \mathfrak{g}$, we define a new transition of $\mathcal{A}'$

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

3. Take as initial state $q_0' = (q_0, \{0\})$.
4. Remove all the states and transitions not reachable from the initial state. □

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 TA is *fleshy* if the following holds:

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

**Proposition 2.** *Given a region-split TA $\mathcal{A}$ accepting a language $L$, a fleshy region-split nice TA $\mathcal{A}'$ 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. 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 $V_n$. □

From now on, we suppose w.l.o.g. that the automaton $\mathcal{A}$ is in a fleshy region-split form (see Fig. 4).

### 2.5 Computing Volumes

Given a timed automaton $\mathcal{A}$ satisfying A1-A3, 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. For every state $(q, \mathbf{x})$, let $L(q, \mathbf{x})$ be the set of all the timed words corresponding to the runs of the automaton starting at this state, let $L_n(q, \mathbf{x})$ be its sublanguage

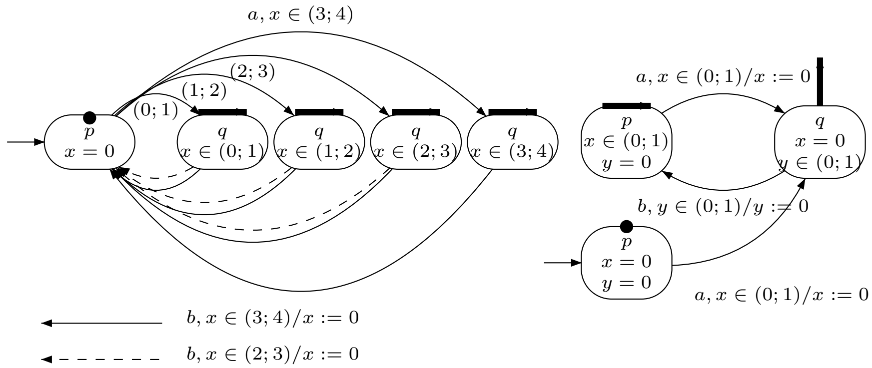


**Fig. 4.** Fleshy region-split forms of automata $\mathcal{A}_2$ and $\mathcal{A}_3$ from Fig. 1. An entry region is drawn at each location.

consisting of its words of length $n$, and $v_n(q, \mathbf{x})$ 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 language equations:

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

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:

$$v_0(q, \mathbf{x}) = 1; \tag{1}$$
$$v_{k+1}(q, \mathbf{x}) = \sum_{(q,a,\mathfrak{g},\mathfrak{r},q') \in \Delta} \int_{\tau : \mathbf{x}+\tau \in \mathfrak{g}} v_k(q', \mathfrak{r}(\mathbf{x} + \tau)) \, d\tau. \tag{2}$$

Remark that for a fixed location $q$, and within every clock region, as defined in [11], the integral over $\tau : \mathbf{x} + \tau \in \mathfrak{g}$ can be decomposed into several $\int_l^u$ 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.

**Lemma 1.** *The function $v_n(q, \mathbf{x})$ restricted to a location $q$ and a clock region 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 timed automaton $\mathcal{A}$ satisfying A1-A3, the volume $V_n$ is a rational number, computable from $\mathcal{A}$ and $n$ using the procedure 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 functional analysis, first introduced in [12].

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 $\Psi$ on this space applied to the unit function (Sect. 3.2). We explore some elementary properties of this operator in 3.3. This makes it possible to apply in 3.4 the theory of positive operators to $\Psi$ and to deduce the main theorem of this paper stating that the entropy equals the logarithm of the spectral radius of $\Psi$.

### 3.1 The Functional Space of a TA

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 $\mathcal{F}$, yielding 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 form $v_{k+1} = \Psi v_k$, where $\Psi$ is the positive linear operator on $\mathcal{F}$ defined by the equation:

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

We have also $v_0 = 1$. Hence $v_n = \Psi^n 1$, and the problem of computing volumes and entropy is now phrased as studying iterations of a positive bounded linear operator $\Psi$ on the functional space $\mathcal{F}$. The theory of positive operators guarantees, that under some hypotheses, $v_n$ is close in direction to a positive eigenvector $v^*$ of $\Psi$, corresponding to its leading eigenvalue $\rho$. Moreover, values of $v_n$ will grow/decay exponentially like $\rho^n$. In the sequel, we refer to the book [3] when a result concerning positive operators is needed.

### 3.3 Exploring the Operator $\Psi$

Let us first state some elementary properties of this operator, starting by rewriting (3) as an operator on $\mathcal{F}$ and separating all its summands.

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

For $\delta = (q, a, \mathfrak{g}, \mathfrak{r}, q')$ the operator $\psi_\delta$ 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:

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

Iterating (4), we obtain a formula for powers of operator $\Psi$

$$(\Psi^k f)_p(\mathbf{x}) = \sum_{\delta_1\ldots\delta_k \text{ from } p \text{ to } p'} (\psi_{\delta_1}\ldots\psi_{\delta_k} f_{p'})(\mathbf{x}). \tag{5}$$

Now we need some results on the iterations of $\psi_\delta$. For this, first we state some useful properties of $\psi_\delta$ and its partial derivatives:

**Proposition 3.** *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 $\delta$ resets $x_i$ then $\psi_\delta f$ is continuously differentiable by $x_i$ and $\|\frac{\partial}{\partial x_i}\psi_\delta f\| \leq 2\|f\|$*
4. *If $\delta$ does not reset $x_i$ and $f$ is continuously differentiable by $x_i$, then $\psi_\delta f$ is continuously differentiable by $x_i$ and $\|\frac{\partial}{\partial x_i}\psi_\delta f\| \leq 2\|f\| + \|\frac{\partial}{\partial x_i} f\|$.*

*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 $\mathbf{x}_2 \in \mathfrak{g}$ such that $\mathfrak{r}(\mathbf{x}_2) = \mathbf{x}_1$. As $\mathbf{x}_2 \in \mathfrak{g}$, there also exists $x_3 \in \mathbf{r}_q$ and $\tau \in \mathbb{R}_{\geq 0}$ verifying $\mathbf{x}_2 = \mathbf{x}_3 + \tau_0$.

Furthermore, because $\delta$ is fleshy, there exists $\tau_1$ and $\tau_2$, $\tau_1 < \tau_2$, such that for every $\tau \in [\tau_1, \tau_2]$, $x_3 + \tau \in \mathfrak{g}$.

Put together, the integration interval of $\psi_\delta f(\mathbf{x}_3) = \int_{\mathbf{x}_3+\tau\in\mathfrak{g}} f(\mathfrak{r}(\mathbf{x}_3+\tau))\, d\tau$ contains a value $\tau_0$, for which the integrated function is positive, and includes $[\tau_1, \tau_2]$, thus is neither empty nor a singleton. The integrated function being non-negative and continuous, its integral, $\psi_\delta f(\mathbf{x}_3)$, is positive.
(2) In $\psi_\delta f(\mathbf{x}) = \int_{\mathbf{x}+\tau\in\mathfrak{g}} f(\mathfrak{r}(\mathbf{x}+\tau))\, d\tau$, we estimate $|f(\cdot)|$ from above by the constant $\|f\|$ and the length of the integration interval by 1, as it is included in the region $\mathfrak{g}$. This gives us the requested bound.
(3) As $\mathfrak{r}$ resets $x_i$, $f(q', \mathfrak{r}(\mathbf{x}+\tau))$ does not depend on $x_i$, and thus $\psi_\delta(q, x) = \int_{\mathbf{x}+\tau\in\mathfrak{g}} f(q', \mathfrak{r}(\mathbf{x}+\tau))\, d\tau$ is differentiable by $x_i$. Its derivative is

$$\frac{\partial}{\partial x_i}\psi_\delta f(q, \mathbf{x}) = \frac{\partial}{\partial x_i}\int_{\mathbf{x}+\tau\in\mathfrak{g}} f(q', \mathfrak{r}(\mathbf{x}+\tau))\, d\tau \tag{6}$$

$$= \pm(f(q', \mathfrak{r}(\mathbf{x}+\tau_{max}) - f(q', \mathfrak{r}(\mathbf{x}+\tau_{min}))). \tag{7}$$

The choice of $+$ or $-$ sign in the line (7) and the bounds $\tau_{max}$ and $\tau_{min}$ depend on the form of the guard.

First, observe that the latter term is a sum of continuous functions and, as such, is continuous. Furthermore, this term is bounded in absolute value by $2\|f\|$. Thus, we prove $|\frac{\partial}{\partial x_i}\psi_\delta f(q, \mathbf{x})| \leq 2\|f\|$.

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

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

The resulting expression is still continuous in $x_i$. Indeed the newly added term in the last 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 $|(f(q', \mathfrak{r}(\mathbf{x}+\tau_{max}) - f(q', \mathfrak{r}(\mathbf{x}+\tau_{min})))|$ is smaller than $2\|f\|$. Also in $\int_{\mathbf{x}+\tau\in\mathfrak{g}} \frac{\partial}{\partial x_i} f(q', \mathfrak{r}(\mathbf{x}+\tau))\, d\tau$, we can estimate the integrated function from above by the norm $\|\frac{\partial}{\partial x_i} f\|$. As the integration interval is smaller than 1, the integral is smaller than this norm too. Hence, the required inequality holds: $|\frac{\partial}{\partial x_i}\psi_\delta f(q,\mathbf{x})| \le 2\|f\| + \|\frac{\partial}{\partial x_i} f\|$. □

Now, we can prove the following result on the powers of $\Psi$.

**Proposition 4.** *Consider operator $\Psi$.*

1. *If $f \ge 0$ is 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 A4 there exists a constant $E \in \mathbb{R}$ such that for any $f \in \mathcal{F}$ the following estimate hold:*

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

*Proof.*

(1) This is a straightforward induction using (5) and Prop. 3-1.

(2) For some $x_i$, and a location $p$, the following equality holds:

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

Let us consider one term of this sum corresponding to one path. By hypothesis, in this path, there is a first transition $\delta_k$, $1 \le k \le D$, such that $\delta_k$ resets $x_i$.

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

Now we differentiate this term. For every $j$, $1 \le j \le D$, iterating Prop. 3-2 $D - j$ times, we obtain $\left\|\psi_{\delta_j}\ldots\psi_{\delta_D} f_{p'}\right\| \le \|f\|$. Thus, by Prop. 3-3, we have

$\left\|\frac{\partial}{\partial x_i}\psi_{\delta_k}\dots\psi_{\delta_D}f_{p'}\right\| \leq 2\|f\|$. It follows by induction on the path, using Prop. 3-4, that $\left\|\frac{\partial}{\partial x_i}\psi_{\delta_1}\dots\psi_{\delta_k}\dots\psi_{\delta_D}f_{p'}\right\| \leq 2k\|f\|$.

Now, if we come back to the sum, we have, at least, the following bound: $\left\|\frac{\partial}{\partial x_i}(\Psi^D f)_p\right\| \leq 2d^D D\|f\|$ ($d$: maximal degree of the underlying graph of $\Delta$), which is true for every $p$, therefore $\|\frac{\partial}{\partial x_i}\Psi^D f\| \leq 2d^D D\|f\|$. □

Now we are ready to prove the following important property of $\Psi$:

**Theorem 2.** *The operator $\Psi^D$ is compact on $\mathcal{F}$.*

*Proof.* 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. 4-2, that the whole set $\Psi^D\mathcal{B}$ is Lipschitz continuous with constant $E\#C$, where $\#C$ is the dimension of the clock space. Hence it is equicontinuous, and, by Arzela-Ascoli theorem, compact. □

Next two lemmata will be used in the proof of the Main Theorem. Denote by $\rho$ the spectral radius of $\Psi$.

**Lemma 2.** *If $\rho > 0$ then it is an eigenvalue of $\Psi$ with an 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 Thm. 2. □

**Lemma 3.** *If $\rho > 0$ then the eigenvector $v^*$ satisfies $v^*(q_0, 0) > 0$.*

*Proof.* 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. 4-1, 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. □

### 3.4 Main Theorem

The main result of this paper can now be stated.

**Theorem 3.** *For any nice TA $\mathcal{A}$ the entropy $\mathcal{H}$ of its language coincides with logarithm of the spectral radius of the $\Psi$ operator defined on $\mathcal{F}$.*

*Proof.* Notice that

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

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: $\rho = \lim_{n\to\infty} \|\Psi^n\|^{1/n}$. Thus, we obtain the required upper bound for the entropy:

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

| |
|---|
| 1. Transform $\mathcal{A}$ into the fleshy region-split form and check that it has $1\frac{1}{2}$ clock. |
| 2. Write the integral eigenvalue equation (I) with one variable. |
| 3. Derivate (I) w.r.t. $x$ and get a differential equation (D). |
| 4. Instantiate (I) at 0, and obtain a boundary condition (B). |
| 5. Solve (D) with boundary condition (B). |
| 6. Take $\rho = \max\{\lambda \mid$ a non-0 solution exists$\}$. |
| 7. Return $\mathcal{H}(L(\mathcal{A})) = \log \rho$. |

**Table 1.** The idea of the symbolic algorithm: computing $\mathcal{H}$ for $1\frac{1}{2}$ clocks

In the case when $\rho > 0$ we also have to prove the lower bound. In this case Lemma 2 applies and an eigenvector $v^* \geq 0$ with norm 1 exists. This yields the inequality $v^* \leq 1$, to which, for any natural $n$, we can apply the positive operator $\Psi^n$. Using the fact that $v^*$ is an eigenvector and the formula for $v_n$ we obtain $\rho^n v^* \leq v_n$. Then, taking the values of the functions in the initial state we get $\rho^n v^*(q_0; 0) \leq V_n$. Hence, by Lemma 3, denoting the positive number $v^*(q_0; 0)$ by $\delta$: $\rho^n \delta \leq V_n$. Taking logarithm, dividing by $n$, and taking the limit we obtain:

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

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

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

## 4 Computing the Entropy

The characterization of $\mathcal{H}$ in Theorem 3 solves the main problem explored in this paper, but its concrete application requires computing the spectral radius of an integral operator $\Psi$, 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 4.2 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 (fleshy region-split) automata with entry regions of all the locations having dimension 0 or 1. In other words, in such automata for every discrete transition there is at most one clock non reset. We call this class $1\frac{1}{2}$ *clock automata*. The idea of the symbolic algorithm for computing the entropy of such automata is presented in Table 1.

Notice first that the set $S = \{(q, \mathbf{x}) \mid \mathbf{x} \in \mathbf{r}_q\}$ is now a disjoint union of unit length intervals and singleton points. After a change of variables, each of those unit intervals can be represented as $x \in (0; 1)$, and a singleton point as $x = 0$.

In both cases $x$ is a scalar variable, equal in the first case to the fractional part of $x_q$, where $x_q \in C$ is the only clock whose value is positive in $\mathbf{r}_q$. Thus, every $f \in F$ can be seen as a finite collection of functions $f_q$ of one scalar argument: $x$.

In this case the expression of the operator $\psi_\delta$, corresponding to one transition $\delta = (q, a, \mathfrak{g}, \mathfrak{r}, q')$, can be made more explicit. First we recall the definition of $\psi_\delta$:

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

A careful but straightforward analysis shows that from the entry region of every state $q$, non-degenerated regions of two types are alternatively visited: regions where $x_q$ is greater than the other clocks, and regions where it is not.

Assuming $t$ is the difference between the fractional parts of $x'_q$ and $x_q$, for guards $\mathfrak{g}$ that are regions of the first type (a), $x + \tau \in \mathfrak{g}$ is equivalent to $t \in (0, 1-x)$, and for the other type (b), it is equivalent to $t \in (1-x, 1)$.

Furthermore, the reset function $\mathfrak{r}$ can behave in three different ways: either it resets every clock but one that is not $x_q$ (1), or it resets every clock but $x_q$ (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 $\Delta_{qb3}$, such that $\Psi$ can now be written the following way:

$$\begin{aligned}\Psi f(q,x) = & \sum_{\delta \in \Delta_{qa1}} \int_0^{1-x} f(q', x+t)dt && + \sum_{\delta \in \Delta_{qb1}} \int_{-x}^{0} f(q', x+t)dt \\ & + \sum_{\delta \in \Delta_{qa2}} \int_0^{1-x} f(q', t)dt && + \sum_{\delta \in \Delta_{qb2}} \int_{1-x}^{1} f(q', t)dt \\ & + \sum_{\delta \in \Delta_{qa3}} (1-x) f(q', 0) && + \sum_{\delta \in \Delta_{qb3}} x f(q', 0).\end{aligned}$$

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

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

This is the explicit formula for $\Psi$ we have been looking for. Now, computing the entropy of the language of the automaton using Thm. 3 involves finding the leading eigenvalue of $\Psi$, that is the greatest $\lambda \in \mathbb{R}$ such that for some non-zero function $f \in \mathcal{F}$:

$$\Psi f = \lambda f. \tag{8}$$

We will solve this by transforming this equality into a differential equation. A smooth function $h : [0,1] \to \mathbb{R}$ equals 0 iff $h(0) = 0$ and $h'(x) = 0$ for all

$x \in (0,1)$. Applying this to $(\Psi f - \lambda f)$[7] we obtain that (8) is equivalent to the differential equation

$$\lambda f'(x) = \quad (D_{b1} - D_{a1}) f(x) + (D_{b2} - D_{a2}) f(1-x) + (D_{b3} - D_{a3}) f(0). \quad (9)$$

with boundary condition

$$\lambda f(0) = (D_{a1} + D_{a2}) \int_0^1 f(t)dt + D_{a3} f(0). \quad (10)$$

Now we solve the differential equation (9) 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 $x \mapsto f(1-x)$ and enables us to rewrite the previous equation as a differential system:

$$\begin{cases} \lambda u'(x) = & Aw(x) \\ \lambda w'(x) = & Bu(x) + C(u(0) + w(0)) \\ w(\frac{1}{2}) = & 0 \end{cases}, \quad (11)$$

where $A \triangleq D_{b1} - D_{a1} - D_{b2} + D_{a2}$, $B \triangleq D_{b1} - D_{a1} + D_{b2} - D_{a2}$ and $C \triangleq D_{b3} - D_{a3}$.

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 (9) can be recovered by adding those two equations term by term.

System (11) implies

$$\begin{cases} \lambda^2 w''(x) = BAw(x) \\ \lambda u'(x) = Aw(x) \\ w(\frac{1}{2}) = 0. \end{cases} \quad (12)$$

The first equation of (12) 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 (12) yields a solution space of dimension $2n$.

Now having a solution $(u, w)$ to (12) implies that $\lambda^2 w''(x) = BAw(x)$, which implies $\lambda w'(x) = \frac{1}{\lambda} \int_0^x BAw(t)dt + \lambda w'(0)$ and thus $\lambda w'(x) = \int_0^x Bu'(t)dt + \lambda w'(0) = Bu(x) - Bu(0) + \lambda w'(0)$.

Therefore $(u, w)$ is also a solution to (11) if and only if $C(u(0) + w(0)) + Bu(0) = \lambda w'(0)$. Coming back to (9), $f \triangleq \frac{u+w}{2}$ is a solution to this system if and only if

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

[7] It is easy to see that for eigenfunctions $f$ this expression should be smooth and well-defined in 0 and 1.

or again

$$(\lambda - B - 2C)f(0) = (\lambda + B)f(1). \tag{13}$$

To sum up, the dimension of the space of the solutions of (12) is $2n$, thus so is the space $S$ of the functions $f = \frac{u+w}{2}$ such that $(u, w)$ is solution to (12). This allows us to write every such $f$ as $FM$ where $F$ is an $n \times 2n$ matrix whose columns are a basis of $S$, and $M$ is a vertical vector of $\mathbb{R}^{2n}$.

Every such $f = FM$ is a solution of (8) if and only if both (10) and (13) hold, which we rewrite here, replacing $f$ by $FM$:

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

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

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

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

This is a transcendental equation on $\lambda$ (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 $\Psi$ (Lem. 2). The logarithm of this value is the entropy of the language (Thm. 3).

Summing up all those computations yields the complete algorithm for automata with $1\frac{1}{2}$ clocks depicted in Table 2.

1. Transform $\mathcal{A}$ into the fleshy region-split form and check that it has $1\frac{1}{2}$ clock.
2. Compute the matrices $D_{ij}$ and next $A, B, C$.
3. Deduce the general solution $FM$ to (9).
4. Find the greatest root $\rho$ (w.r.t. the unknown $\lambda$) of

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

5. Then we have $\mathcal{H}(L(\mathcal{A})) = \log \rho$.

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

**Application to the Running Example** We apply the method just described to compute the entropy of the language of the automaton $\mathcal{A}_3$ of Fig. 1 which is a "$1\frac{1}{2}$ clocks" one. Its fleshy region-split form is presented on Fig. 4.

By symmetry, the volume of a path of length $n \in \mathbb{N}$ is the same function $v_n$ in both non-initial states. Thus $v_n$ is characterized by:

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

According to Thm. 3, the entropy can be found as $\log \rho(\Psi)$, and by Lemma 2 $\rho(\Psi)$ is the maximal eigenvalue of $\Psi$. Let us write the eigenvalue equation:

$$\lambda v(x) = \int_0^{1-x} v(t)dt. \tag{14}$$

Differentiating it twice w.r.t $x$ we get:

$$\lambda v'(x) = -v(1-x) \tag{15}$$

$$\lambda^2 v''(x) = -v(x) \tag{16}$$

The solutions have the form $v(x) = \alpha \sin(\frac{x}{\lambda}) + \beta \cos(\frac{x}{\lambda})$. Using (14) with $x = 1$ we find $v(1) = 0$. We inject this in (15) for $x = 0$ and deduce $\alpha = 0$. Thus $v(x) = \beta \cos(\frac{x}{\lambda})$ and $\cos(\frac{1}{\lambda}) = 0$. This implies that the solutions correspond to $\lambda = \frac{2}{(2k+1)\pi}$ with $k \in \mathbb{Z}$. The highest of those is $\lambda = 2/\pi$, and we can verify that $v(x) = \cos(\frac{x\pi}{2})$ satisfies $\frac{2}{\pi} v = \Psi v$. Therefore $\rho(\Psi) = 2/\pi$, and the entropy of this automaton is $\log(2/\pi)$.

### 4.2 General Case

If several clocks are not reset in some transitions, then the entry regions are multi-dimensional, and the volume functions therefore depend on several real variables. Hence, we cannot reduce the integral equation to an ordinary differential equation, which makes it difficult to find the eigenfunction symbolically. Instead, we can use standard iterative procedures for eigenvalue approximation for positive operators. Recall that the volume function satisfies $v_n = \Psi^n 1$. The following theorem is close to Thms. 16.1-16.2 from [3].

**Theorem 4.** *If for some $\alpha, \beta \in \mathbb{R}, m \in \mathbb{N}$ the following inequality holds: $\alpha v_m \leq v_{m+1} \leq \beta v_m$, and the volume $V_m = v_m(q_0, 0) > 0$, then $\log \alpha \leq \mathcal{H} \leq \log \beta$.*

*Proof.* Applying the positive operator $\Psi^n$ to the inequalities $\alpha v_m \leq v_{m+1} \leq \beta v_m$, and using the formula $v_n = \Psi^n 1$ 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 $n$

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

| |
|---|
| 1. Transform $\mathcal{A}$ into the fleshy region-split form.<br>2. Choose an $m$ and compute symbolically the piecewise polynomial functions $v_m$ and $v_{m+1}$.<br>3. Check that $v_m(q_0, 0) > 0$.<br>4. Compute $\alpha = \min(v_{m+1}/v_m)$ and $\beta = \max(v_{m+1}/v_m)$.<br>5. Conclude that $\mathcal{H} \in [\log \alpha; \log \beta]$. |

**Table 3.** Iterative algorithm: bounding $\mathcal{H}$

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

**Table 4.** Iterating the operator for $\mathcal{A}_3$ ($\mathcal{H} = \log(2/\pi) \approx \log 0.6366 \approx -0.6515$)

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 $\limsup_{n\to\infty}$ (remember that $\mathcal{H} = \limsup_{n\to\infty} \log V_n/n$):

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

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

This theorem yields a procedure[8] to estimate $\mathcal{H}$ summarized in Table 3.

**Example: Again $\mathcal{A}_3$** We apply the iterative procedure above to our running example $\mathcal{A}_3$. As explained in Sect. 4.1, 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 4.

[8] One possible optimization is to compute $\alpha$ and $\beta$ separately on every strongly connected reachable component of the automaton, and take the maximal values.

## 5 Discretization Approach

### 5.1 Discretizing the Volumes

Another approach, we first published in [13], is 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 [14, 15].

### 5.2 $\varepsilon$-words and $\varepsilon$-balls

We start with a couple of preliminary definitions. Take an $\varepsilon = 1/N > 0$. A timed word $w$ is $\varepsilon$-timed if all the delays in this word are multiples of $\varepsilon$. Any $\varepsilon$-timed word $w$ over an alphabet $\Sigma$ can be written as $w = h_\varepsilon(v)$ for an untimed $v \in \Sigma \cup \{\tau\}$, where the morphism $h_\varepsilon$ is defined as follows:

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

The discrete word $v$ with ticks $\tau$ (standing for $\varepsilon$ delays) represents in this way the $\varepsilon$-timed word $w$.

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

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

For a timed word $w = t_1a_1t_2a_2\dots t_na_n$ we introduce its North-East $\varepsilon$-neighbourhood like this:

$$\mathcal{B}_\varepsilon^{NE}(w) = \{s_1a_1s_2a_2\dots s_na_n \mid \forall i\,(s_i \in [t_i; t_i + \varepsilon])\}\,.$$

For a language $L$, we define its NE-neighbourhood elementwise:

$$\mathcal{B}_\varepsilon^{NE}(L) = \bigcup_{w \in L} \mathcal{B}_\varepsilon^{NE}(w). \tag{17}$$

The next simple lemma will play a key role in our algorithm (here $\#L$ stands for the cardinality of $L$).

**Lemma 4.** *Let $L$ be some finite set of timed words of length $n$. Then*

$$\mathrm{Vol}(\mathcal{B}_\varepsilon^{NE}(L)) \le \varepsilon^n \#L.$$

*If, moreover, $L$ is $\varepsilon$-timed, then*

$$\mathrm{Vol}(\mathcal{B}_\varepsilon^{NE}(L)) = \varepsilon^n \#L.$$

*Proof.* Notice that for a timed word $w$ of a length $n$ the set $\mathcal{B}_\varepsilon^{NE}(w)$ is a hypercube of edge $\varepsilon$ (in the delay space), and of volume $\varepsilon^n$. Notice also that neighbourhoods of different $\varepsilon$-timed words are almost disjoint: the interior of their intersections are empty. With these two remarks, the two statements are immediate from (17). □

### 5.3 Discretizing Timed Languages and Automata

Suppose now that we have a timed language $L$ recognized by a timed automaton $\mathcal{A}$ satisfying A2-A5 and we want to compute its entropy (or just the volumes $V_n$). Take an $\varepsilon = 1/N > 0$. We will build two $\varepsilon$-timed languages $L_-$ and $L_+$ that under- and over-approximate $L$ in the following sense:

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

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

**Lemma 5.** *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 $u \in L(A_-^\varepsilon)$, let $v = h_\varepsilon(u)$ be its $\varepsilon$-timed version, and let $w \in \mathcal{B}_\varepsilon^{NE}(v)$. We have to prove that $w \in L$. Notice first that $L(A_-^\varepsilon) = L(C_-^\varepsilon)$ and hence $u$ is accepted by $C_-^\varepsilon$. Mimic the run of $C_-^\varepsilon$ on $u$, but replace every $\tau$ by an $\varepsilon$ duration, thus, a run of $\mathcal{A}$ on $v$ can be obtained. 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 $[t_i, t_i+\varepsilon]$. 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 $w \in L$. □

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

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

Clearly, $\underline{t}$ is always a multiple of $\varepsilon$ 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 $\varepsilon$-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 $\varepsilon$ and it would belong to $[x - D\varepsilon; x)$. Hence $x' \in [l \dot{-} D\varepsilon, u - \varepsilon]$. 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. □

### 5.4 Counting Discrete Words

Once the automata $A_+^\varepsilon$ and $A_-^\varepsilon$ constructed, we can count the number of words with $n$ events and its asymptotic behaviour using the following simple result.

**Lemma 6.** *Given an automaton $\mathcal{B}$ over an alphabet $\{\tau\} \cup \Sigma$, let*

$$L_n = L(\mathcal{B}) \cap (\tau^* \Sigma)^n .$$

*Then (1) $\#L_n$ is computable; and (2) $\lim_{n\to\infty}(\log \#L_n/n) = \log \rho_{\mathcal{B}}$ with $\rho_{\mathcal{B}}$ a computable algebraic real number.*

*Proof.* We proceed in three stages. First, we determinize $\mathcal{B}$ and remove all the useless states (unreachable from the initial state). These transformations yield an automaton $\mathcal{D}$ accepting the same language, and hence having the same cardinalities $\#L_n$. Since the automaton is deterministic, to every word in $L_n$ corresponds a unique accepting path with $n$ events from $\Sigma$ and terminating with such an event.

Next, we eliminate the tick transitions $\tau$. As we are counting paths, we obtain an automaton without silent ($\tau$) transitions, but with multiplicities representing the number of realizations of every transition. More precisely, the procedure is as follows. Let $\mathcal{D} = (Q, \{\tau\} \cup \Sigma, \delta, q_0)$. We build an automaton with multiplicities $\mathcal{E} = (Q, \{e\}, \Delta, q_0)$ over one-letter alphabet. For every $p, q \in Q$ the multiplicity of the transition $p \to q$ in $\mathcal{E}$ equals the number of paths from $p$ to $q$ in $\mathcal{D}$ over words from $\tau^*\Sigma$. A straightforward induction over $n$ shows that the number of paths in $\mathcal{D}$ with $n$ non-tick events equals the number of $n$-step paths in $\mathcal{E}$ (with multiplicities).

Let $M$ be the adjacency matrix with multiplicities of $\mathcal{E}$. It is well known (and easy to see) that the $\#L(n)$ (that is the number of $n$-paths) can be found as the sum of the first line of the matrix $M^{\underline{n}}$. This allows computing $\#L(n)$. Moreover, using Perron-Frobenius theorem we obtain that $\#L(n) \sim \rho^n$ where $\rho$ is the spectral radius of $M$, the greatest (in absolute value) real root $\lambda$ of the integer characteristic polynomial $\det(M - \lambda I)$. □

### 5.5 From Discretizations to Volumes

As soon as we know how to compute the cardinalities of under- and over- approximating languages $\#L_-(n)$ and $\#L_+(n)$ and their growth rates $\rho_-$ and $\rho_+$, we can deduce the following estimates solving our problems.

**Theorem 5.** *For a timed automaton $\mathcal{A}$ satisfying A2-A5, the $n$-volumes of its language satisfy the estimates:*

$$\#L_-(n) \cdot \varepsilon^n \le V_n \le \#L_+(n) \cdot \varepsilon^n.$$

*Proof.* In inclusions (18) take the volumes of the three terms, and use Lemma 4. □

**Theorem 6.** *For a timed automaton $\mathcal{A}$ satisfying A2-A5, the entropy of its language satisfies the estimates:*

$$\log(\varepsilon\rho_-) \le \mathcal{H}(L(\mathcal{A})) \le \log(\varepsilon\rho_+).$$

*Proof.* Just use the previous result, take the logarithm, divide by $n$ and pass to the limit. □

We summarize the algorithm in Table 5.

| |
|---|
| 1. Choose an $\varepsilon = 1/N$. |
| 2. Build the counter automata $C_-^\varepsilon$ and $C_+^\varepsilon$. |
| 3. Transform them into finite automata $A_-^\varepsilon$ and $A_+^\varepsilon$. |
| 4. Eliminate $\tau$ transitions introducing multiplicities. |
| 5. Obtain adjacency matrices $M_-$ and $M_+$. |
| 6. Compute their spectral radii $\rho_-$ and $\rho_+$. |
| 7. Conclude that $\mathcal{H} \in [\log \varepsilon\rho_-; \log \varepsilon\rho_+]$. |

**Table 5.** Discretization algorithm: bounding $\mathcal{H}$

This theorem can be used to estimate the entropy. However, it can also be read in a converse direction: the cardinality of $L$ restricted to $n$ events and discretized with quantum $\varepsilon$ 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. 6 below.

**A Case Study.** Consider the example $L_3 = \{t_1 a t_2 b t_3 a t_4 b \cdots \mid t_i + t_{i+1} \in [0; 1]\}$ from Sect. 2.2. 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. 5 according to the recipe (the discrete ones $A_+$ and $A_-$ are too big to fit on the figure).

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 $12.41^n \cdot 0.05^n \le V_n \le 13.05^n \cdot 0.05^n$, and the entropy

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

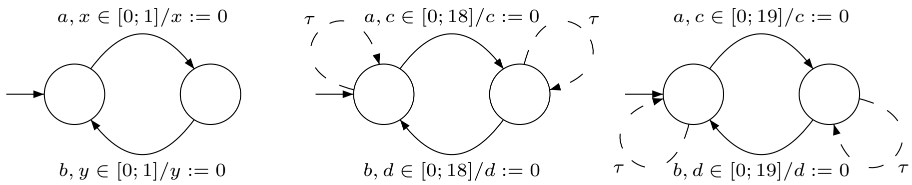


**Fig. 5.** A two-clock timed automaton $\mathcal{A}_3$ and its approximations $C_-^{0.05}$ and $C_+^{0.05}$. All $\tau$-transitions increment counters $c$ and $d$.

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 proved in 4.1 that the true value of the entropy is $\mathcal{H} = \log(2/\pi) \approx \log 0.6366 \approx -0.6515$.

## 6 Kolmogorov Complexity of Timed Words

To interpret the results above in terms of information content of timed words we state, using similar techniques, some estimates of Kolmogorov complexity of timed words. Recall first the basic definition from [16] (see also [17]). Given a partial computable function (decoding method) $f : \{0; 1\}^* \times B \to A$, a description of an element $x \in A$ knowing $y \in B$ is a word $w$ such that $f(w, y) = x$. The Kolmogorov complexity of $x$ knowing $y$, denoted $K_f(x|y)$ is the length of the shortest description. According to Kolmogorov-Solomonoff theorem, there exists the best (universal) decoding method providing shorter descriptions (up to an additive constant) than any other method. The complexity $K(x|y)$ with respect to this universal method represents the quantity of information in $x$ knowing $y$.

Coming back to timed words and languages, remark that a timed word within a "simple" timed language can involve rational delays of a very high complexity, or even uncomputable real delays. For this reason, we consider timed words with finite precision $\varepsilon$. For a timed word $w$ and $\varepsilon = 1/N$ we say that a timed word $v$ is a rational $\varepsilon$-approximation of $w$ if all delays in $v$ are rational and $w \in \mathcal{B}_\varepsilon^{NE}(v)$[9].

**Theorem 7.** *Let $\mathcal{A}$ be a timed automaton satisfying A2-A4, $L$ its language, $\mathcal{H}$ its entropy. For any rational $\alpha, \varepsilon > 0$, and any $n \in \mathbb{N}$ large enough there exists a timed word $w \in L$ of length $n$ such that the Kolmogorov complexity of all the rational $\varepsilon$-approximations $v$ of the word $w$ is lower bounded as follows*

$$K(v|n, \varepsilon) \geq n(\mathcal{H} + \log 1/\varepsilon - \alpha). \tag{19}$$

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

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

[9] In this section, we use such South-West approximations $v$ for technical simplicity only.

Consider the set $S$ of all timed words $v$ violating the lower bound (19)

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

The cardinality of $S$ can be bounded as follows:

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

Applying Lemma 4 we obtain

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

We deduce that the set $L_n$ of timed words from $L$ of length $n$ 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 $\varepsilon$. □

**Theorem 8.** *Let $\mathcal{A}$ be a timed automaton satisfying A2-A4, $L$ 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 $x \in [l \dot{-} \alpha, u+\alpha]$. Let $\mathcal{H}'$ be 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 length $n$, there exists its $\varepsilon$-approximation $v$ with Kolmogorov complexity 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 $\varepsilon$-approximation as in Lemma 5: $v = \underline{t_1} a_1 \dots \underline{t_n} a_n$. To find an upper bound for the complexity of $v$ we notice that $v \in U$, where $U$ is the set of all $\varepsilon$-timed words $u$ of $n$ letters such that $\mathcal{B}_\varepsilon^{NE}(u) \subset L'_n$. Applying Lemma 4 to the set $U$ we obtain the bound

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

Hence, in order to encode $v$ (knowing $n$ and $\varepsilon$) it suffices 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. □

Two theorems above provide close upper and lower bounds for complexity of $\varepsilon$-approximations of elements of a timed language.

However, the following example shows that because we removed Assumption A5, in some cases these bounds do not match and $\mathcal{H}'$ can possibly not converge towards $\mathcal{H}$ when $\alpha$ becomes small.

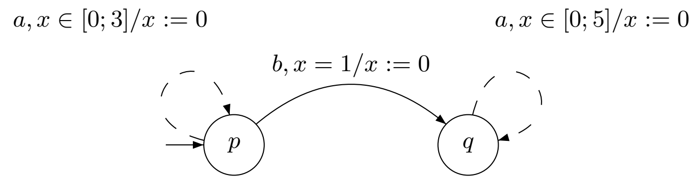


**Fig. 6.** A pathological automaton

*Example 1.* Consider the automaton of Fig. 6. For this example, the state $q$ does not contribute to the volume, and $\mathcal{H} = \log 3$. Nevertheless, when we bloat the guards, both states become "usable" and, for the bloated automaton $\mathcal{H}' \approx \log 5$. As for Kolmogorov complexity, for $\varepsilon$-approximations of words from the sublanguage $1b([0;5]a)^*$ it behaves as $n(\log 5 + \log(1/\varepsilon))$. Thus, for this bothering example, the complexity matches $\mathcal{H}'$ rather than $\mathcal{H}$.

## 7 Conclusions and Further Work

In this paper, we have defined size characteristics of timed languages: volume and entropy. The entropy 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. Three 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 and compare them to each other. 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 estimate the gap between our upper and lower bounds for the entropy (we believe that this gap tends to 0 for strongly connected automata) and establish entropy computability. We would be happy to remove some of Assumptions A1-A5, in particular non-Zenoness. Kolmogorov complexity estimates can be improved, in particular, as shows Example 1, it could be more suitable to use another variant of entropy, perhaps $\mathcal{H}^+ = \max \mathcal{H}_q$, where the entropy is maximized with respect to initial states $q$. 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. Another research direction is to associate a dynamical system (a subshift) to a timed language and to explore entropy of this dynamical system.

## 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. Sassoli, L., Vicario, E.: Close form derivation of state-density functions over dbm domains in the analysis of non-Markovian models. In: QEST'07, IEEE Computer Society (2007) 59–68
6. Bertrand, N., Bouyer, P., Brihaye, T., Markey, N.: Quantitative model-checking of one-clock timed automata under probabilistic semantics. In: QEST'08, IEEE Computer Society (2008) 55–64
7. Asarin, E., Pokrovskii, A.: Use of the Kolmogorov complexity in analyzing control system dynamics. Automation and Remote Control (1) (1986) 25–33
8. Rojas, C.: Computability and information in models of randomness and chaos. Mathematical Structures in Computer Science **18**(2) (2008) 291–307
9. Brudno, A.: Entropy and the complexity of the trajectories of a dynamical system. Trans. Moscow Math. Soc. **44** (1983) 127–151
10. Asarin, E., Caspi, P., Maler, O.: Timed regular expressions. Journal of the ACM **49** (2002) 172–206
11. Alur, R., Dill, D.L.: A theory of timed automata. Theoretical Computer Science **126** (1994) 183–235
12. Asarin, E., Degorre, A.: Volume and entropy of regular timed languages: Analytic approach. To appear in proceedings of FORMATS'09 (2009)
13. Asarin, E., Degorre, A.: Volume and entropy of regular timed languages: Discretization approach. To appear in proceedings of Concur'09 (2009)
14. 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
15. Henzinger, T.A., Manna, Z., Pnueli, A.: What good are digital clocks? In: ICALP'92. LNCS 623, Springer-Verlag (1992) 545–558
16. Kolmogorov, A.: Three approaches to the quantitative definition of information. Problems of Information Transmission **1**(1) (1965) 1–7
17. Li, M., Vitányi, P.: An introduction to Kolmogorov complexity and its applications. 3 edn. Springer (2008)