



Generating functions of timed languages

Eugene Asarin, Nicolas Basset, Aldric Degorre, Dominique Perrin

► To cite this version:

Eugene Asarin, Nicolas Basset, Aldric Degorre, Dominique Perrin. Generating functions of timed languages. 2012. hal-00678443v1

HAL Id: hal-00678443

<https://hal.science/hal-00678443v1>

Submitted on 12 Mar 2012 (v1), last revised 24 Apr 2012 (v2)

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.

Generating functions of timed languages^{*}

Eugene Asarin^{**1}, Nicolas Basset², Aldric Degorre¹, and Dominique Perrin²

¹ LIAFA, University Paris Diderot and CNRS, France

² LIGM, University Paris-Est Marne-la-Vallée and CNRS, France

Abstract. In order to study precisely the growth of timed languages, we associate to such a language a generating function. These functions (tightly related to volume and entropy of timed languages) satisfy compositionality properties and can be characterized by integral equations. We provide procedures for computation of generating functions for some classes of timed automata and regular expressions.

1 Introduction

Since the introduction of timed automata in [1], these automata and their languages are extensively studied both in theoretical perspective and in applications to verification of real-time systems. However, the natural question of measuring the size of timed languages was addressed only recently in [4, 3] and a couple of subsequent works. In those articles we have explored the asymptotic behaviour of the volume of a timed language when the number of events tends to ∞ . We have shown that for most timed automata this volume grows (or decreases) exponentially, defined entropy as its growth rate, characterized this entropy as a logarithm of the spectral radius of an integral operator Ψ and shown how to compute the entropy symbolically or numerically.

In this article, we make a much more precise size analysis of timed languages. We associate to such a language L a sequence of its volumes $\text{Vol}(L_n)$, and the generating function $f(z) = \sum_n \text{Vol}(L_n)z^n$. Thus the function $f(z)$ contains a complete information on the “size profile” of $\text{Vol}(L_n)$ as a function of n . To relate it to the previous work, we show that $f(z)$ can be expressed in terms of the resolvent of the operator Ψ , and that the entropy of a timed language depends only on the convergence radius of $f(z)$.

The example in Fig. 1, from [3], is extensively used throughout the paper to illustrate the notions of volume, generating function and entropy. The timed language recognized by this automaton is

$$L = \{t_1, a, t_2, b, t_3, a, \dots \mid \forall i (t_i + t_{i+1} \leq 1)\}.$$

^{*} The support of Agence Nationale de la Recherche under the project EQINOCS (ANR-11-BS02-004) is gratefully acknowledged.

^{**} A part of the work of this author was done during his stay at the Institute of Mathematical Sciences (National University of Singapore) within its Programme “Automata Theory and Applications”.

For any number of events n we have a polytope in \mathbb{R}^n :

$$L_n = \{t_1, t_2, t_3, \dots, t_n \mid \forall i (t_i + t_{i+1} \leq 1)\},$$

the sequence of volumes V_n of these polytopes is

$$1; 1; \frac{1}{2}; \frac{1}{3}; \frac{5}{24}; \frac{2}{15}; \frac{61}{720}; \frac{17}{315}; \frac{277}{8064}; \frac{62}{2835}; \frac{50521}{3628800} \dots,$$

and it was shown in [3] that this sequence behaves asymptotically like $(2/\pi)^n$. Here, we will show that the generating function of volumes is $\tan z + \sec z$.

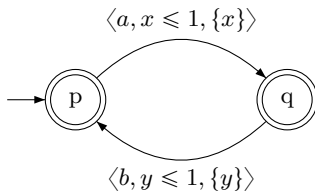


Fig. 1. A timed automaton (our running example)

The convergence radius of the series $\pi/2$ is the inverse of the growth rate of the sequence V_n . This series describes precisely the sequence of volumes, and a closed-form formula for V_n can be deduced (see [10]):

$$V_{2n-1} = \frac{B_{2n}(-4)^n(1-4^n)}{(2n)!}; \quad V_{2n} = \frac{(-1)^n E_{2n}}{(2n)!},$$

where B_s stand for Bernoulli numbers and E_s for Euler numbers.

Generating functions behave in a natural way with respect to simple operations on timed languages (disjoint union, unambiguous concatenation, unambiguous star). However in order to obtain an exact characterization and eventually closed-form expressions for generating function of timed regular languages a more involved analysis is needed. Such an analysis constitutes the main contribution of the article.

Related work. Generating functions of regular languages have been thoroughly studied and applied, see e.g. [6, 9]. We are not aware of any work on generating functions of timed languages. Techniques and ideas used in this article build on our previous works on volumes and entropy of timed languages, especially on [3] (however the current article is self-contained). As for automata and languages under study, we investigate timed regular languages of [1], clock languages and expressions as in [7], and subclasses of timed automata: regenerating automata from [11] and their subclass known as real-time automata [8], and $1\frac{3}{4}$ -clocks automata that extend both regenerating automata and $1\frac{1}{2}$ -clocks automata from [3].

Article structure. In Sect. 2 we introduce a formalism (inspired by [7]) for timed and clock languages, introduce volume functions of such languages, and investigate the properties of these functions. In Sect. 3 we introduce generating functions of timed languages and investigate their general properties. In Sect. 4 we explain how to compute generating functions for several subclasses of timed automata. We summarize the contributions and discuss the directions of future work in Sect. 5.

2 Preliminaries

2.1 Clock languages and timed languages

In this paper, we study timed language (mostly regular) using an approach based on clock languages due to Bouyer and Petit [7]. We present this approach in a slightly different form, and we use a multi-stage semantics like in [5]. The general idea is as follows: we are interested in timed languages. Timed languages are obtained as projections of clock languages. Clock languages are homomorphic images of discrete “triplet languages”. Triplet languages, in their turn, are generated by automata, regular expressions or grammars. Below we define formally all these notions and illustrate them on our running example.

An alphabet of *timed events* is the product $\mathbb{R}^+ \times \Sigma$ where Σ is a finite alphabet. The meaning of a timed event (t, a) is that t is the time delay before the event a . A *timed word* is a sequence of timed events and a *timed language* is just a set of timed words.

Inspired by [7] we enrich timed words and languages with d -dimensional clock vectors. A clock is a variable which takes values in \mathbb{R}^+ . In our setting, values of clocks will be bounded by a positive integer M . A clock word is a timed word together with an initial and a final clock vector, i.e. an element of $\mathbb{R}^d \times (\mathbb{R}^+ \times \Sigma)^* \times \mathbb{R}^d$. Two clock words $(\mathbf{x} \parallel \mathbf{w} \parallel \mathbf{y})$ and $(\mathbf{x}' \parallel \mathbf{w}' \parallel \mathbf{y}')$ are said to be compatible if $\mathbf{y} = \mathbf{x}'$, in this case we define their product by $(\mathbf{x} \parallel \mathbf{w} \parallel \mathbf{y})(\mathbf{y} \parallel \mathbf{w}' \parallel \mathbf{y}') = (\mathbf{x} \parallel \mathbf{w}\mathbf{w}' \parallel \mathbf{y}')$. A clock language is a set of clock words. The product of two clock languages \mathcal{L} and \mathcal{L}' is $\mathcal{L} \cdot \mathcal{L}' = \{c \cdot c' \mid c \in \mathcal{L}, c' \in \mathcal{L}', c \text{ and } c' \text{ compatible}\}$. The neutral element \mathcal{E} is $\{(\mathbf{x} \parallel \epsilon \parallel \mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^d\}$ and the Kleene star of a language \mathcal{L} is as usual $\mathcal{L}^* = \bigcup_k \mathcal{L}^k$ with $\mathcal{L}^0 = \mathcal{E}$.

A clock language \mathcal{L} is said to be deterministic whenever for each clock word the final clock vector is uniquely determined by the initial clock vector and the timed word, in other word there is a function $\sigma_{\mathcal{L}} : \mathbb{R}^d \times (\mathbb{R}^+ \times \Sigma)^* \rightarrow \mathbb{R}^d$ such that for any clock word $(\mathbf{x} \parallel \mathbf{w} \parallel \mathbf{y})$ of \mathcal{L} , we have that $\mathbf{y} = \sigma_{\mathcal{L}}(\mathbf{x}, \mathbf{w})$. In the following, we work with deterministic clock languages.

To a clock language we associate its timed projections. Given \mathcal{L} , we define $\mathcal{L}(\mathbf{x}, \mathbf{x}')$ as the timed language leading from \mathbf{x} to an element lower than \mathbf{x}' : $\mathcal{L}(\mathbf{x}, \mathbf{x}') = \{w \mid \exists \mathbf{y}(\mathbf{x} \parallel \mathbf{w} \parallel \mathbf{y}) \in \mathcal{L} \wedge \mathbf{y} \leq \mathbf{x}'\}$. We also define the timed language $\mathcal{L}(\mathbf{x}) = \{w \mid \exists \mathbf{y}(\mathbf{x} \parallel \mathbf{w} \parallel \mathbf{y}) \in \mathcal{L}\}$ as the language starting from \mathbf{x} . Remark that $\mathcal{L}(\mathbf{x}) = \mathcal{L}(\mathbf{x}, \mathbf{M})$ where $\mathbf{M} = (M, \dots, M)$ is the greatest clock vector possible.

2.2 Triplet, clock and timed languages

Triplets. Following [7] we define a *triplet* as a tuple $\langle a, \mathbf{g}, \mathbf{r} \rangle$ with: a a letter in Σ ; \mathbf{g} a set of constraints, called guard, $x_i \bowtie c$ ($i \in \{1, \dots, d\}$, $c \in \{0, \dots, M\}$, $\bowtie \in \{<, >, \leq, \geq\}$) and $\mathbf{r} \subseteq \{1 \dots d\}$ a set of indices of clocks to be reset. We suppose moreover that guards bound all the clocks by M . We denote by \mathcal{T} the finite alphabet of such triplets.

Clock semantics of triplets. The clock language of a triplet $\langle a, \mathbf{g}, \mathbf{r} \rangle$ is $\mathcal{L}(\langle a, \mathbf{g}, \mathbf{r} \rangle) = \{(\mathbf{x} \parallel (a, t) \parallel \mathbf{x}') \mid \mathbf{x} + t \models \mathbf{g} \wedge \mathbf{r}(\mathbf{x} + t) = \mathbf{x}'\}$. Here, for a clock vector $\mathbf{x} = (x_1, \dots, x_d)$, we denote by $\mathbf{x} + t$ the vector $(x_1 + t, \dots, x_d + t)$. Clock vectors are updated as follows: $\mathbf{r}(y_1, \dots, y_d) = (y'_1, \dots, y'_d)$ with $y'_i = 0$ if $i \in \mathbf{r}$ and $y'_i = y_i$ otherwise. The reader acquainted with timed automata will notice that a triplet corresponds to a transition of such an automaton.

This definition can be extended to all triplet words by: $\mathcal{L}(\epsilon) = \mathcal{E}$ and $\mathcal{L}(\pi_1 \dots \pi_n) = \mathcal{L}(\pi_1) \dots \mathcal{L}(\pi_n)$. Finally for a language $L \subseteq \mathcal{T}^*$, we define $\mathcal{L}(L) = \{\mathcal{L}(\pi) \mid \pi \in L\}$. In fact, \mathcal{L} is a morphism between the two Kleene algebras $(\mathcal{P}(\mathcal{T}^*), \cup, \cdot, \emptyset, \epsilon)$ and $(\mathcal{P}(\mathbb{R}^d \times (\mathbb{R}^+ \times \Sigma)^* \times \mathbb{R}^d), \cup, \cdot, \emptyset, \mathcal{E})$.

Timed automata and their languages. A timed automaton \mathcal{A} is a finite automaton with alphabet \mathcal{T} . Its discrete semantics L is the language of triplet words accepted by \mathcal{A} seen as a finite automaton; its clock semantics is $\mathcal{L}_{\mathcal{A}} = \mathcal{L}(L)$ and its timed semantics is $\mathcal{L}_{\mathcal{A}}(\mathbf{0})$. The timed automata considered in this article are supposed to be deterministic in the sense of [1].

A timed rational expression is defined as expression over the finite alphabet \mathcal{T} . Its discrete, clock and timed semantics are defined similarly to the automata.

These multi-stage timed semantics are equivalent to the usual semantics of timed automata, timed rational expressions³, etc.

Example 1. Consider our running example on Fig. 1. The discrete semantics is the set of triplet words of the form $\langle a, x \leq 1, \{x\} \rangle \langle b, y \leq 1, \{y\} \rangle \langle a, x \leq 1, \{x\} \rangle \langle b, y \leq 1, \{y\} \rangle \dots$, captured by a rational expression $(\langle a, x \leq 1, \{x\} \rangle \langle b, y \leq 1, \{y\} \rangle)^* \cdot (\epsilon + \langle a, x \leq 1, \{x\} \rangle)$. The clock language is $\mathcal{L}_{\text{even}} \cup \mathcal{L}_{\text{odd}}$, with

$$\begin{aligned} \mathcal{L}_{\text{even}} = \{ & ((x, y) \parallel (t_1, a)(t_2, b)(t_3, a) \dots (t_{2n}, b) \parallel (t_{2n}, 0)) \mid \\ & x + t_1 \leq 1 \wedge y + t_1 + t_2 \leq 1 \wedge \forall i \in 2..2n-1 (t_i + t_{i+1} \leq 1) \} \end{aligned}$$

and a similar expression for \mathcal{L}_{odd} . By projection, we obtain the timed language:

$$\{(t_1, a)(t_2, b)(t_3, a) \dots (t_n, a/b) \mid t_1 \leq 1 \wedge \forall i \in 1..n-1 (t_i + t_{i+1} \leq 1)\}.$$

A convenient way to see automata is the matrix form. A timed automaton \mathcal{A} over the set of control states Q and alphabet of transitions \mathcal{T} is uniquely described by three ingredients:

³ in the sense of [7], which is different from [2].

- a $Q \times Q$ -matrix \mathbb{T} whose element $\mathbb{T}_{qq'}$ is the set of triplets labelling transitions from q to q' ;
- a row vector \mathbf{I} describing initial states: for each control state p , its element $I_p = \{\epsilon\}$ iff p is initial, and \emptyset otherwise;
- a column vector \mathbf{F} describing final states: for each control state q , its element $F_q = \{\epsilon\}$ iff q is final, and \emptyset otherwise.

The coefficient $(\mathbb{T}^n)_{p,q}$ of \mathbb{T}^n contains the language of all the triplet words of length n from p to q . The p^{th} coordinates of the column vector $\mathbb{T}^n \mathbf{F}$ contains the language recognized from state p and $\mathbf{I} \mathbb{T}^n \mathbf{F}$ contains the language of triplet words of length n recognized by \mathcal{A} . For our running example:

$$\mathbf{I} = (\{\epsilon\} \emptyset); \quad \mathbb{T} = \begin{pmatrix} \emptyset & \{\langle a, x \leq 1, \{x\} \rangle\} \\ \{\langle b, y \leq 1, \{y\} \rangle\} & \emptyset \end{pmatrix}; \quad \mathbf{F} = \begin{pmatrix} \{\epsilon\} \\ \{\epsilon\} \end{pmatrix}.$$

2.3 Volume(s) of timed and clock languages

For a timed language L and a word $w \in \Sigma^*$ of length $n \geq 0$, we define $L(w)$ as a subset of \mathbb{R}^n by $L(w) = \{\mathbf{t} \in \mathbb{R}^n \mid (\mathbf{t}, w) \in L\}$. For a clock language \mathcal{L} and a word $w \in \Sigma^*$ of length $n \geq 0$, we define the clock language $\mathcal{L}(w) = \{(\mathbf{x} \parallel (w, \mathbf{t}) \parallel \mathbf{x}') \mid (\mathbf{x} \parallel (w, \mathbf{t}) \parallel \mathbf{x}') \in \mathcal{L}\}$.

Measurable timed languages and clock languages. A timed language is *measurable* if for any word w , $L(w)$ is a measurable subset of \mathbb{R}^n . A clock language \mathcal{L} is measurable if it is deterministic and for every $w \in \Sigma^*$, $\sigma_{\mathcal{L}}(\cdot, (w, \cdot))$ is a measurable function of $\mathbb{R}^d \times \mathbb{R}^{|w|} \rightarrow \mathbb{R}^d$. We remark that timed languages and deterministic clock languages obtained from triplet languages (all the languages considered here) are measurable.

Volumes of a timed language [3]. The sequence of volumes $(V_n(L))_{n \in \mathbb{N}}$ associated to a measurable timed language is $V_n(L) = \sum_{w \in \Sigma^n} \text{Vol}(L(w))$, where Vol is the hyper-volume (i.e. Lebesgue measure) in \mathbb{R}^n . For dimension 0 we define $V_0(L) = 1$ if $\epsilon \in L$, and $V_0(L) = 0$ otherwise.

Volumes constrained by initial and final clock vectors. Timed regular languages considered below come from clock languages (which themselves come from triplet languages). The information about clock vectors is crucial to compute the volume of timed languages in a compositional manner.

Thus we define parametric volumes depending on initial and final clock vectors as follows $V_n^2(x, x') = V_n(\mathcal{L}(\mathbf{x}, \mathbf{x}'))$. We call this function the cumulative volume function (CVF)⁴ of \mathcal{L} . We also allow the following notations: for a clock language \mathcal{L} and a discrete events word w , $V_{\mathcal{L}(w)}^2(\mathbf{x}, \mathbf{x}') = V_{|w|}^2(\mathcal{L}(w)(\mathbf{x}, \mathbf{x}'))$; and for a triplets word π , $V_{\pi}^2(\mathbf{x}, \mathbf{x}') = V_{|\pi|}(\mathcal{L}(\pi)(\mathbf{x}, \mathbf{x}'))$. The notion of parametric volumes can be also applied to the clock language constrained only by initial clock vector $\mathcal{L}(\mathbf{x})$: $V_n^1(\mathbf{x}) = V_n(\mathcal{L}(\mathbf{x}))$. Clearly $V_n^1(\mathbf{x}) = V_n^2(\mathbf{x}, \infty) = V_n^2(\mathbf{x}, M)$.

⁴ similarly to cumulative distribution functions in probability theory.

CVFs for a triplet word. According to the following result, a CVF is easy to compute for a triplet word, and hence for a finite triplet language.

Proposition 1. *For a triplet word π the CVF V_π^2 is piecewise polynomial with rational coefficients of degree less or equal $|\pi|$. The pieces are polytopes, and an expression of this function is computable.*

Composing CVFs. In order to define a composition for CVF corresponding to the concatenation of triplet words and languages, we proceed as follows. We define composition of two functions of $\mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ as:

$$V_1^2 \star V_2^2(\mathbf{x}, \mathbf{x}') = \int_{\mathbf{y}} V_2^2(\mathbf{y}, \mathbf{x}') V_1^2(\mathbf{x}, d\mathbf{y}),$$

where the integral is the Lebesgue-Stieltjes integral⁵. We also define

$$V_1^2 \star v(\mathbf{x}) = \int_{\mathbf{y}} v(\mathbf{y}) V_1^2(\mathbf{x}, d\mathbf{y}),$$

when v is defined on \mathbb{R}^d . Then we can state the key lemma (to transpose concatenation of words to the CVFs world):

Proposition 2. *For any measurable clock languages \mathcal{L}_1 and \mathcal{L}_2 and discrete words w_1 and w_2 , $V_{\mathcal{L}_1(w_1)}^2 \star V_{\mathcal{L}_2(w_2)}^2$ is well defined and satisfies:*

$$V_{\mathcal{L}_1(w_1)}^2 \star V_{\mathcal{L}_2(w_2)}^2 = V_{\mathcal{L}_1(w_1) \cdot \mathcal{L}_2(w_2)}^2.$$

Volume functions in timed automata. As we did for languages, we introduce a Q -vector $\mathbf{V}_n(\mathbf{x})$ of volumes of clock languages and a $Q \times Q$ -matrix $\mathbb{V}(\mathbf{x}, \mathbf{x}')$ of cumulative volume functions of elements of the transition matrix \mathbb{T} :

$$\mathbf{V}_{n,q}(\mathbf{x}) = V_n^1(\mathcal{L}_q(\mathbf{x})); \quad \mathbb{V}_{qq'}(\mathbf{x}, \mathbf{x}') = V_1(\mathcal{L}(\mathbb{T}_{qq'})_{\leq}(\mathbf{x}, \mathbf{x}')).$$

It follows from the proposition above that the matrix element $(\mathbb{V}^{\star n})_{pq}$ (of the matrix power wrt \star) contains the CVF of $\mathcal{L}(\mathbb{T}^n)_{pq}$, that is of the language of all the clock words of length n leading from p to q . Finally, $\mathbf{V}_n = \mathbb{V}^{\star n} \star \mathbf{V}_F$ with \mathbf{V}_F a column vector with $\mathbf{V}_{F,p} = 1$ if p is initial, and $\mathbf{V}_{F,p} = 0$ otherwise.

Proposition 3. *In a timed automaton \mathcal{A} , for $n \geq 1$, the volume functions $\mathbf{V}_n(\mathbf{x})$ and $\mathbb{V}^{\star n}(\mathbf{x}, \mathbf{x}')$ are continuous wrt the initial clock vector \mathbf{x} .*

We can reformulate propositions 2-3 in the language of functional analysis. Consider the Banach space \mathcal{F} of Q -vectors of continuous functions on clock valuations. Thus an element of \mathcal{F} is a vector \mathbf{v} whose components are continuous $v_q : [0; M]^d \rightarrow \mathbb{R}$, and $\mathcal{F} = C([0; M]^d)^Q$. The matrix \mathbb{V} corresponds to an

⁵ By definition, Lebesgue-Stieltjes integral $\int f(\mathbf{x})g(d\mathbf{x})$ is the Lebesgue integral of f wrt the measure μ having cumulative distribution function g .

operator $\Psi : \mathcal{F} \rightarrow \mathcal{F}$ defined by $\Psi(\mathbf{v}) = \mathbb{V} \star \mathbf{v}$. It is easy to see that Ψ is a bounded linear operator on \mathcal{F} , represented by a matrix of integral operators⁶.

Theorem 1 ([3]). *For a timed automaton \mathcal{A} the volume vector can be obtained by iteration of operator Ψ : $\mathbf{V}_n = \Psi^n(\mathbf{V}_F)$.*

3 Generating functions

3.1 Definitions

To study volume sequences associated to timed and clock languages we define their generating functions. As usual for generating functions, they allow recovering the sequence, its growth rate, momenta etc; and they have nice compositional properties. Given a timed language L its generating function is defined as follows:

$$f_L(z) = \sum_k z^k V_k(L).$$

Given a clock language \mathcal{L} , we define a (parametric) generating function with a given initial clock vector

$$f_{\mathcal{L}}^1(z, \mathbf{x}) = \sum_k z^k V_k(\mathcal{L}(\mathbf{x})) = f_L(z), \text{ with } L = \mathcal{L}(\mathbf{x}).$$

For a clock language \mathcal{L} we also define another cumulative generating function with a given initial clock vector and a bound on the final clock vector:

$$f^2(z, \mathbf{x}, \mathbf{x}') = \sum_k z^k V_k^2(\mathbf{x}, \mathbf{x}') = f_L(z), \text{ with } L = \mathcal{L}(\mathbf{x}, \mathbf{x}').$$

To summarize, we are interested to compute $f(z)$, but this computation will be based on $f^1(z, \mathbf{x})$, and sometimes on $f^2(z, \mathbf{x}, \mathbf{x}')$.

Given a timed automaton, timed and clock languages, and thus generating function are naturally associated to its states, for example

$$f_q^1(z, \mathbf{x}) = \sum_k z^k V_k(\mathcal{L}_q(\mathbf{x})) = f_L(z), \text{ with } L = \mathcal{L}_q(\mathbf{x}).$$

Taken for all states, functions f_q and f_q^1 form $|Q|$ -dimensional vector functions $\mathbf{f}(z, \mathbf{x})$, $\mathbf{f}^1(z, \mathbf{x})$, while functions $f_{q,q'}^2$ form a $Q \times Q$ -matrix function $\mathbb{f}^2(z, \mathbf{x}, \mathbf{x}')$.

3.2 Analytic characterization

Elementary properties. First, let us state the relations between the three kinds of generating functions:

⁶ in [3] a slightly different definition of the operator Ψ was given. Here we add a matricial point of view but results of [3] extend straightforwardly to the present setting.

Proposition 4. *Functions f, f^1, f^2 are related as follows: $f(z) = f^1(z, \mathbf{0})$; $f^1(z, \mathbf{x}) = f^2(z, \mathbf{x}, \mathbf{M})$.*

By definition, f^2, f^1 and f are analytic functions of z . Since we consider timed automata with guards bounded by some constant M , all the volumes V_k (with any initial or final conditions) can be upper bounded by $(M|\Sigma|)^k$. This implies that convergence radius of series for f^2, f^1 and f is at least $(M|\Sigma|)^{-1} > 0$. More precisely, the radius of convergence of f is $1/\limsup_{k \rightarrow \infty} (V_k(L))^{1/k} = 2^{-\mathcal{H}(L)}$, where $\mathcal{H}(L)$ is called the volumetric entropy of L (see [3]).

For generating functions associated to timed automata, the following result is a straightforward corollary of Prop. 3:

Proposition 5. *Within its convergence radius, the generating function $\mathbf{f}^1(z, \mathbf{x})$ associated to a timed automaton \mathcal{A} is continuous wrt the initial clock vector \mathbf{x} .*

Integral equation and resolvent of Ψ . Consider a timed automaton. Its generating function can be computed as follows:

$$\mathbf{f}^1(z, \mathbf{x}) = \sum_k z^k V_k(\mathcal{L}(\mathbf{x})) \mathbf{V}_k = \sum_k z^k \mathbb{V}^{\star k} \star \mathbf{V}_F,$$

which implies the following characterization of generating functions of timed regular languages by a system of integral equations.

Theorem 2 (Integral equation). *In the interior of its convergence circle, the generating function f^1 is the unique solution of the integral equation*

$$\mathbf{f}^1 - z \mathbb{V} \star \mathbf{f}^1 = \mathbf{V}_F. \quad (1)$$

This theorem can be rephrased in terms of the operator Ψ associated to the automaton. Recall that, by definition, the resolvent of an operator A is $R(\lambda, A) = (A - \lambda I)^{-1}$; it is well defined when λ does not belong to the spectrum of A , in particular for $|\lambda| > \rho(A)$, where ρ denotes the spectral radius.

Corollary 1 (Generating function and resolvent). *The generating function f^1 satisfies the formula: $\mathbf{f}^1 = -z^{-1} R(z^{-1}, \Psi) \mathbf{V}_F$, which holds in the interior of the circle $|z| < \rho(\Psi)^{-1}$.*

Example (1, continued). For our running example

$$\mathbb{V} = \begin{pmatrix} 0 & \min(1 \div x, y' \div y) \mathbb{1}_{x' \geq 0} \\ \min(1 \div y, x' \div x) \mathbb{1}_{y' \geq 0} & 0 \end{pmatrix}; \quad \mathbf{V}_F = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Equation (1) gives:

$$f_p^1(z, x, y) - z \int_{y'=y}^{y+1-x} f_q^1(z, 0, y') dy' = 1; \quad f_q^1(z, x, y) - z \int_{x'=x}^{x+1-y} f_p^1(z, x', 0) dx' = 1. \quad (2)$$

In Section 4.1 below we develop a technique for solving such equations (for a subclass of automata including this one), and compute the generating function for this language.

3.3 Inductive characterisation of generating function

The form of generating functions of finite triplet languages is immediate from Prop. 1:

Proposition 6. *For a finite triplet language L with maximal word length ℓ , the generating functions f^2, f^1 are piecewise polynomial in $z, \mathbf{x}, \mathbf{x}'$ (pieces are polytopes in \mathbf{x}, \mathbf{x}') of degree $\leq \ell$ wrt z and wrt \mathbf{x} and \mathbf{x}' .*

More complex language can be obtained from fined ones using Kleene algebra operations. We consider first the simple case of timed languages.

Proposition 7. *Generating functions behave well for unambiguous operations on measurable timed languages: $f_{L_1 \cup L_2} = f_{L_1} + f_{L_2}$ whenever $L_1 \cap L_2 = \emptyset$; $f_{L_1 \cdot L_2} = f_{L_1} f_{L_2}$ whenever the product is non ambiguous; $f_{L^*} = 1 + f_L f_L^*$ whenever $\epsilon \notin L$ and the star is not ambiguous.*

However, in order to obtain general timed regular languages we need operations on clock languages, which are more involved.

Proposition 8. *Generating functions f^2 behave well for unambiguous operations on deterministic measurable clock languages: $f_{\mathcal{L}_1 \cup \mathcal{L}_2}^2 = f_{\mathcal{L}_1}^2 + f_{\mathcal{L}_2}^2$ whenever $\mathcal{L}_1 \cap \mathcal{L}_2 = \emptyset$; $f_{\mathcal{L}_1 \cdot \mathcal{L}_2}^2 = f_{\mathcal{L}_1}^2 \star f_{\mathcal{L}_2}^2$ whenever the product is non ambiguous; $f_{\mathcal{L}^*}^2 = \mathbb{1}_{\mathbf{x} \leq \mathbf{x}'} + f_{\mathcal{L}}^2 \star f_{\mathcal{L}}^2$ whenever $\epsilon \notin \mathcal{L}$ and the Kleene star is not ambiguous.*

Corollary 2. *Generating function f^1 for compositions of clock languages can be computed as follows. $f_{\mathcal{L}_1 + \mathcal{L}_2}^1 = f_{\mathcal{L}_1}^1 + f_{\mathcal{L}_2}^1$ whenever $\mathcal{L}_1 \cap \mathcal{L}_2 = \emptyset$; $f_{\mathcal{L}_1 \cdot \mathcal{L}_2}^1 = f_{\mathcal{L}_1}^1 \star f_{\mathcal{L}_2}^1$ whenever the product is non ambiguous; $f_{\mathcal{L}^*}^1 = 1 + f_{\mathcal{L}}^1 \star f_{\mathcal{L}}^1$ whenever $\epsilon \notin \mathcal{L}$ and the Kleene star is not ambiguous.*

4 Computing generating functions

The generating function of a timed language represented by an automaton is characterized by a system of integral equations (1). The generating function of a timed language represented by a rational expression can be found recursively from piecewise polynomial functions using operations $+$, \star and solving fixpoint integral equations of Prop. 8 and Cor. 2. Unfortunately, both procedures involve computation of integrals, and solution of integral equations, for this reason, the result cannot be always presented by an explicit formula. Below we consider several subclasses of timed automata, for which generating functions can be obtained in closed form, or at least admit a simpler characterization.

4.1 Generating functions for particular classes of automata

System of equations. Our closed-form solutions for subclasses of timed automata will be obtained using a variant of language equations.

Let $Q = G \cup B$ be a disjoint partition of the states of a timed automaton \mathcal{A} into *good* and *bad*. We want to describe the vector $\bar{\mathbf{L}}$ of triplet languages L_q recognized from good states $q \in G$ only. This vector satisfies the equation:

$$\bar{\mathbf{L}} = \bar{\mathbb{T}} \cdot \bar{\mathbf{L}} + \bar{\mathbf{F}}, \quad (3)$$

where $\bar{\mathbb{T}}$ is a $G \times G$ -matrix and $\bar{\mathbf{F}}$ is a G -vector of triplet languages. Their elements are defined as follows: $\bar{\mathbb{T}}_{pq}$ consists of all words leading from p to q via bad states only; \bar{F}_p consists of all words leading from p to a final state via bad states only⁷.

Automata with regeneration. Following [11], we call an automaton regenerating if there exists a partition $Q = G \cup B$ having two properties:

- every cycle in the automaton contains a state in G (good);
- all the transitions coming into a good state reset all clocks.

W.l.o.g. we suppose that the initial state is good. The first condition implies that no cycle is possible within bad states, and thus all the elements of $\bar{\mathbb{T}}$ and $\bar{\mathbf{F}}$ are finite triplet languages (with maximal word length $\leq |B| + 1$). The second condition means that (3) can be rewritten in timed languages (instead of clock languages), since when entering in a good state all clocks are reset. This gives

$$\bar{\mathbf{L}}_{\text{timed}} = \bar{\mathbb{T}}_{\text{timed}} \cdot \bar{\mathbf{L}}_{\text{timed}} + \bar{\mathbf{F}}_{\text{timed}}. \quad (4)$$

Applying simple compositionality conditions for generating functions for timed languages (Prop. 7) we obtain that

$$\bar{\mathbf{f}} = \bar{\mathbb{f}}\bar{\mathbf{f}} + \bar{\mathbf{f}}_F. \quad (5)$$

Due to Prop. 6 all the coefficients (elements of matrix $\bar{\mathbb{f}}$ and vector $\bar{\mathbf{f}}_F$) are polynomials of z . Solving the linear $|G|$ -dimensional system (5) we express $\bar{\mathbf{f}}$ as a vector of rational functions of z :

$$\bar{\mathbf{f}} = (I - \bar{\mathbb{f}})^{-1} \bar{\mathbf{f}}_F. \quad (6)$$

The generating function f of the timed language accepted by the automaton is just one element of this vector $\bar{\mathbf{f}}$. We conclude.

Theorem 3. *For a regenerating automaton the generating function $f(z)$ is a rational function.*

Example 2. Consider a regenerating automaton on Fig. 2 (left). We choose good and bad states as follows: $G = \{p, q\}$; $B = \{r\}$. The system of equations on timed languages of good states takes the form

$$\begin{pmatrix} L_p \\ L_q \end{pmatrix} = \begin{pmatrix} \emptyset & T_{pq} \\ T_{qp} & \emptyset \end{pmatrix} \cdot \begin{pmatrix} L_p \\ L_q \end{pmatrix} + \begin{pmatrix} F_p \\ \emptyset \end{pmatrix} \quad \text{with} \quad (7)$$

$$\begin{aligned} T_{pq} &= \{(t_1, a)(t_2, b) | 2 < t_1 < 3 \wedge t_1 + t_2 < 5\} \cup \{(t, b) | t < 8\}; \\ T_{qp} &= \{(t, b) | t < 7\}; \\ F_p &= \{(t, a) | 2 < t < 3\}. \end{aligned}$$

⁷ if this final state is good the word should be ϵ .

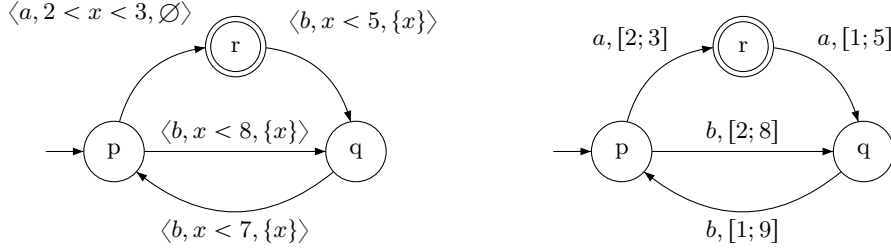


Fig. 2. A regenerating automaton (left) and a real-time automaton (right)

For generating functions this yields:

$$\begin{pmatrix} f_p \\ f_q \end{pmatrix} = \begin{pmatrix} 0 & 2.5z^2 + 8z \\ 7z & 0 \end{pmatrix} \cdot \begin{pmatrix} f_p \\ f_q \end{pmatrix} + \begin{pmatrix} z \\ 0 \end{pmatrix}. \quad (8)$$

Solving this linear system we find the required

$$f_p(z) = \frac{2z}{2 - 35z^3 - 112z^2}.$$

It converges for $|z| < 0.1309$, its Taylor coefficients (i.e. volumes V_n for $n = 0..11$) are

$$0; 1; 0; 56; 17\frac{1}{2}; 3136; 1960; 175922\frac{1}{4}; 164640; 9885946; 12298479\frac{3}{8}; 556494176.$$

Real-time automata. We consider here transition-labeled real-time automata (t-RTA) from [8]. They are automata, in which to any transition $p \xrightarrow{a} q$ is associated a time interval $[l, u]$. This transition can be taken after spending between l and u time units in p . Equivalently, a real-time automaton can be seen as a timed automaton with only one clock, which is reset on any transition.

It is easy to see that real-time automata are regenerating (all their states are good). Thus equation (6) applies. Its coefficients can be found in a more explicit form:

$$\mathbf{f} = (I - z\mathbb{A})^{-1}\mathbf{V}_F, \quad (9)$$

where matrix A_{pq} is the sum of lengths of all time intervals associated to transitions from p to q , and, as before, $V_{Fq} = 1$ iff q is final, and 0 otherwise. This can be seen as a simplified version of the resolvent equation (1) for real-time automata: instead of a matrix of integral operators, a matrix of polynomials is inverted.

Example 3. For the real-time automaton on the right of Fig. 2 equation (9) takes the form:

$$\begin{pmatrix} f_p \\ f_q \\ f_r \end{pmatrix} = \left(I - z \begin{pmatrix} 0 & 6 & 1 \\ 8 & 0 & 0 \\ 0 & 4 & 0 \end{pmatrix} \right)^{-1} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

which gives the required generating function:

$$f_p = \frac{z}{1 - 48z^2 - 32z^3}$$

with convergence circle $|z| < 0.13812$ and first 11 Taylor coefficients (volumes V_n):

$$0; 1; 0; 48; 32; 2304; 3072; 111616; 221184; 5455872; 14188544; 268959744.$$

$1\frac{3}{4}$ -clocks automata. We call an automaton $1\frac{3}{4}$ -clocks if there exists a partition of $Q = G \cup B$ into good and bad states having two properties:

- every cycle in the automaton contains a good state;
- all the transitions coming into a good state reset all clocks except one.

W.l.o.g. we make two other assumptions:

- the initial state is a good one (this can be achieved by adding a new initial state);
- for each good state p there is at most one clock $x_{i(p)}$ non reset by incoming transition (this can be achieved by splitting good states).

Similarly to regenerating automata, we apply equations (3), and observe that all the coefficients are finite triplet languages. Unfortunately, since some clocks are not reset, we cannot write an equation on timed languages similar to (4). Instead, we pass to clock languages and their generating functions, as in the general case. This gives:

$$\overline{\mathbf{f}^1} = \overline{\mathbf{f}^2} \star \overline{\mathbf{f}^1} + \overline{\mathbf{f}^1}_F, \quad (10)$$

an integral equation with piecewise polynomial coefficients. We notice that functions in the last equation depend on the clock vector $\mathbf{x} \in \mathbb{R}^d$ (or on two clock vectors \mathbf{x}, \mathbf{x}'), but in fact for any good state $p \in G$ only one clock $x_{i(p)}$ matters. This allows extracting simpler integral equations from (10), involving only functions of scalar argument.

We proceed as follows: given a G -vector \mathbf{v} whose elements v_p are functions on \mathbb{R}^d , we define reduced functions on \mathbb{R} : $\tilde{v}_p(x) = v_p(0, \dots, 0, x, 0, \dots, 0)$, with the argument x at position $i(p)$. Reduced G -vector $\tilde{\mathbf{v}}$ consists of reduced elements \tilde{v}_p . Reduced version of matrices are defined similarly.

The following identity is based on the requirement of clock resets:

Lemma 1. *For a $1\frac{3}{4}$ -clocks automaton the following holds: $\widetilde{\overline{\mathbf{f}^1}} = \widetilde{\overline{\mathbf{f}^2}} \star \widetilde{\overline{\mathbf{f}^1}} = \widetilde{\overline{\mathbf{f}^2}} \star \widetilde{\overline{\mathbf{f}^1}}$.*

Equation (10), reduced to $\widetilde{\overline{\mathbf{f}^1}} = \widetilde{\overline{\mathbf{f}^2}} \star \widetilde{\overline{\mathbf{f}^1}} + \widetilde{\overline{\mathbf{f}^1}_F}$, implies that the reduced vector of generating functions is a solution of equations of the form:

$$\mathbf{f}(z, x) = (\mathbf{A} \star \mathbf{f})(z, x) + \mathbf{b}(z, x), \quad (11)$$

where all the coefficients are piecewise polynomial functions of z and a scalar argument x .

Lemma 2. *An integral equation of the form (11) can be transformed into a system of linear ordinary differential equation with piecewise polynomial coefficients (depending on x and z).*

Theorem 4. *For a $1\frac{3}{4}$ -clocks automaton the generating function f can be obtained by solving a system of linear ordinary differential equations with piecewise polynomial coefficients.*

We notice that the theorem gives a rather explicit characterization of f , but not always a closed-form expression.

Example (1, completed). For our running example, we choose $G = \{p, q\}$, $B = \emptyset$. The matrix \mathbb{A} and the vector \mathbf{b} are

$$\mathbb{A} = \begin{pmatrix} 0 & z \min(1 \div x, x') \mathbb{1}_{x' \geq 0} \\ z \min(1 \div x, x') \mathbb{1}_{x' \geq 0} & 0 \end{pmatrix}; \quad \mathbf{b} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Equations (11) on generating functions take the form⁸:

$$\widetilde{f}_p^1(z, x) = z \int_{x'=0}^{1-x} \widetilde{f}_q^1(z, x') dx' + 1; \quad \widetilde{f}_q^1(z, x) = z \int_{x'=0}^{1-x} \widetilde{f}_p^1(z, x') dx' + 1.$$

By symmetry the two generating functions \widetilde{f}_p^1 and \widetilde{f}_q^1 are equal to a unique function f^1 which satisfies

$$f^1(z, x) = z \int_{x'=0}^{1-x} f^1(z, x') dx' + 1.$$

Differentiating it twice w.r.t x we obtain:

$$\frac{\partial f^1}{\partial x}(z, x) = -z f^1(z, 1-x); \quad \frac{\partial^2 f^1}{\partial x^2}(z, x) = -z^2 f^1(z, x).$$

The solution has the form $f^1(z, x) = A(z) \cos zx + B(z) \sin zx$. Using $\frac{\partial f^1}{\partial x}(z, 0) = -z f^1(z, 1) = -z$ we obtain $zB(z) = -z$ and thus $B(z) = -1$. Then $f^1(z, 1) = 1$ implies $A(z) \cos z - \sin z = 1$ and thus $A(z) = \frac{1+\sin z}{\cos z} = \tan z + \sec z$. We are done since $f(z) = f^1(z, 0) = A(z) = \tan z + \sec z$.

Example (Airy).

Consider the automaton on Fig. 3. We choose $G = \{p, q\}$, $B = \{r\}$. As in the previous example we have $\mathbb{A}_{pq}(x, x') = \min(1 \div x, x') \mathbb{1}_{x' \geq 0}$. For \mathbb{A}_{qp} , we must compute the volume of the language $\mathcal{L}_{cb}(x_2, x'_1) = \mathcal{L}_{cb}(x, x') = \{t_1, t_2 \geq 0 \mid x + t_1 + t_2 \leq 1 \wedge t_1 + t_2 \leq x'\}$. This is the area of the right-angled triangle of equations $t_1, t_2 \geq 0, t_1 + t_2 \leq u$ where $u = \min(1 \div x, x')$ i.e. $\frac{\min(1 \div x, x')^2}{2}$. We can now give the matrix \mathbb{A} and the vector \mathbf{b} :

$$\mathbb{A} = \begin{pmatrix} 0 & z \min(1 \div x, x') \mathbb{1}_{x' \geq 0} \\ z^2 \frac{\min(1 \div x, x')^2}{2} \mathbb{1}_{x' \geq 0} & 0 \end{pmatrix}; \quad \mathbf{b} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

⁸ It is in fact the reduced form of equation (2).

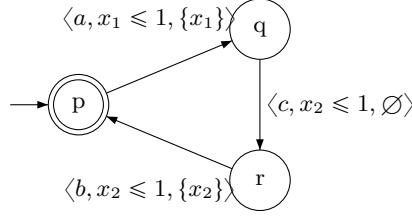


Fig. 3. The Airy automaton

Equations (11) on generating functions take the form:

$$\widetilde{f}_p^1(z, x) = z \int_{x'=0}^{1-x} \widetilde{f}_q^1(z, x') dx'; \quad \widetilde{f}_q^1(z, x) = z^2 \int_{x'=0}^{1-x} x' \widetilde{f}_p^1(z, x') dx' + 1.$$

Differentiating w.r.t x we obtain:

$$\frac{\partial \widetilde{f}_p^1}{\partial x}(z, x) = -z \widetilde{f}_q^1(z, 1-x); \quad \frac{\partial \widetilde{f}_q^1}{\partial x}(z, x) = -z^2(1-x) \widetilde{f}_p^1(z, 1-x).$$

Differentiate once again the former equation gives: $\frac{\partial^2 \widetilde{f}_p^1}{\partial x^2}(z, x) = z \frac{\partial \widetilde{f}_q^1}{\partial x}(z, 1-x)$, combining with the latter one this gives:

$$\frac{\partial^2 f^1}{\partial x^2}(z, x) = -z^3 x f^1(z, x) \text{ with } f^1 = \widetilde{f}_p^1.$$

The solution has the form $f^1(z, x) = \alpha(z)\text{Ai}(-zx) + \beta(z)\text{Bi}(-zx)$ where Ai and Bi are the Airy's functions and $\alpha(z)$, $\beta(z)$ two functions to be determined with the border conditions $f^1(z, 1) = 1$, $\frac{\partial f^1}{\partial x}(z, 0) = 0$. We obtain the following equations:

$$\alpha(z)\text{Ai}(-z) + \beta(z)\text{Bi}(-z) = 1; \quad \alpha(z)\text{Ai}'(0) + \beta(z)\text{Bi}'(0) = 0.$$

Solving this system and simplifying using classical formulae for $\text{Ai}(0)$, $\text{Ai}'(0)$, $\text{Bi}(0)$, $\text{Bi}'(0)$ and Euler's reflection formula, we obtain the final result:

$$f(z) = f^1(z, 0) = \frac{4}{\pi} \cdot \frac{1}{\text{Bi}'(0)\text{Ai}(-z) - \text{Ai}'(0)\text{Bi}(-z)}.$$

5 Conclusions

In this article, we have introduced generating functions of timed languages, explored their properties and characterized them by integral equations. For subclasses of timed regular languages we have presented closed-form expressions or simpler characterization of generating functions. Generating functions describe with a high precision the quantitative behaviour of timed languages.

At the current stage of research, the computation of generating functions is a tedious semi-manual task, and important restrictions are imposed to the

automata. We are planning to explore theoretical and practical algorithmics of timed generating functions, and to implement the algorithm. On the other hand, we want to see whether some analysis is possible beyond the class of $1\frac{3}{4}$ languages.

We hope that this approach will lead to various combinatorial results for timed regular languages, better quantitative characterization of such languages with applications to information theory and algorithm analysis. On the other hand, the approach can be extended to non-regular timed languages or to richer models such as hybrid automata.

References

1. Alur, R., Dill, D.L.: A theory of timed automata. *Theoretical Computer Science* 126, 183–235 (1994)
2. Asarin, E., Caspi, P., Maler, O.: Timed regular expressions. *Journal of the ACM* 49, 172–206 (2002)
3. Asarin, E., Degorre, A.: Volume and entropy of regular timed languages: Analytic approach. In: *FORMATS’09*. pp. 13–27. LNCS 5813 (2009)
4. Asarin, E., Degorre, A.: Volume and entropy of regular timed languages: Discretization approach. In: *Concur’09*. pp. 69–83. LNCS 5710 (2009)
5. Asarin, E., Dima, C.: Balanced timed regular expressions. *Electr. Notes Theor. Comput. Sci.* 68(5), 16–33 (2002)
6. Berstel, J., Reutenauer, C.: Noncommutative rational series with applications, *Encyclopedia of Mathematics and its Applications*, vol. 137. Cambridge University Press, Cambridge (2011)
7. Bouyer, P., Petit, A.: A Kleene/Büchi-like theorem for clock languages. *Journal of Automata, Languages and Combinatorics* 7(2), 167–186 (2002)
8. Dima, C.: Real-time automata. *Journal of Automata, Languages and Combinatorics* 6(1), 3–24 (2001)
9. Flajolet, P., Sedgewick, R.: *Analytic combinatorics*. Cambridge University Press, Cambridge (2009)
10. Korn, G., Korn, T.: *Mathematical Handbook for Scientists and Engineers: Definitions, Theorems, and Formulas for Reference and Review*. McGraw-Hill (1968)
11. Sassoli, L., Vicario, E.: Close form derivation of state-density functions over DBM domains in the analysis of non-Markovian models. In: *QEST’07*. pp. 59–68. IEEE Computer Society (2007)

A Proof of Proposition 1

Proof. The clock language $\mathcal{L}(\pi)$ is a polytope in $\mathbb{R}^d \times \mathbb{R}^n \times \mathbb{R}^d$. Its timed section $\mathcal{L}(\pi)(\mathbf{x}, \mathbf{x}')$ is a polytope in \mathbb{R}^n whose coefficients are linear function of \mathbf{x} and \mathbf{x}' . Volume of such polytopes are computable and have the required form.

B Proof of Proposition 2

Proof. First recall that

$$\begin{aligned} V_{\mathcal{L}_1(w_1) \cdot \mathcal{L}_2(w_2)}^2(\mathbf{x}, \mathbf{x}') &= \text{Vol}\{\mathbf{t} \mid \sigma_{\mathcal{L}_1 \cdot \mathcal{L}_2}(\mathbf{x}, (w_1 w_2, \mathbf{t})) \leq \mathbf{x}'\} \\ &= \int_{\mathbf{t}} d\mathbf{t} \cdot \mathbb{1}_{\sigma_{\mathcal{L}_1 \cdot \mathcal{L}_2}(\mathbf{x}, (w_1 w_2, \mathbf{t})) \leq \mathbf{x}'}, \end{aligned}$$

which gives

$$V_{\mathcal{L}_1(w_1) \cdot \mathcal{L}_2(w_2)}^2(\mathbf{x}, \mathbf{x}') = \int_{\mathbf{t}_1, \mathbf{t}_2} d\mathbf{t}_1 d\mathbf{t}_2 \cdot \mathbb{1}_{\sigma_{\mathcal{L}_2}(\sigma_{\mathcal{L}_1}(\mathbf{x}, (w_1, \mathbf{t}_1)), (w_2, \mathbf{t}_2)) \leq \mathbf{x}' }.$$

By Fubini's theorem this can be rewritten as

$$V_{\mathcal{L}_1(w_1) \cdot \mathcal{L}_2(w_2)}^2(\mathbf{x}, \mathbf{x}') = \int_{\mathbf{t}_1} d\mathbf{t}_1 \left(\int_{\mathbf{t}_2} d\mathbf{t}_2 \mathbb{1}_{\sigma_{\mathcal{L}_2}(\sigma_{\mathcal{L}_1}(\mathbf{x}, (w_1, \mathbf{t}_1)), (w_2, \mathbf{t}_2)) \leq \mathbf{x}' } \right).$$

Applying again the formula for $V_{\mathcal{L}_2(w_2)}^2$, we get that

$$\begin{aligned} V_{\mathcal{L}_1(w_1) \cdot \mathcal{L}_2(w_2)}^2(\mathbf{x}, \mathbf{x}') &= \int_{\mathbf{t}_1} d\mathbf{t}_1 \cdot V_{\mathcal{L}_2(w_2)}^2(\sigma_{\mathcal{L}_1}(\mathbf{x}, (w_1, \mathbf{t}_1)), \mathbf{x}') \\ &= \int_{\mathbf{x}_1} V_{\mathcal{L}_2(w_2)}^2(\mathbf{x}_1, \mathbf{x}') V_{\mathcal{L}_1(w_1)}^2(\mathbf{x}, d\mathbf{x}_1), \end{aligned}$$

as required. The last change of variables can be justified as follows. It has the form:

$$\int_{\mathbf{t}_1} d\mathbf{t}_1 \cdot u(\sigma_{\mathcal{L}_1}(\mathbf{x}, (w_1, \mathbf{t}_1))) = \int_{\mathbf{x}_1} u(\mathbf{x}_1) V_{\mathcal{L}_1(w_1)}^2(\mathbf{x}, d\mathbf{x}_1), \quad (12)$$

and we have to prove that it holds for any measurable u . Indeed, whenever $u(\mathbf{y})$ is an indicator $\mathbb{1}_{\mathbf{y} \leq \mathbf{a}}$, both left-hand and right-hand sides equal $V_{\mathcal{L}_1(w_1)}^2(\mathbf{x}, \mathbf{a})$, thus (12) holds. Any other function u can be obtained as a limit of linear combinations of such indicators.

C Proof of Proposition 3

Proof. We prove the result for V_π^2 for a triplet word π . It is then straightforward to lift the result to the cumulative and non-cumulative volume functions of the n -language of an automaton (finite sum of triplet words).

For a triplet word π , $L(\pi)(\mathbf{x}, \mathbf{x}')$ is actually a polytope, intersection of half-spaces H of equations of one of the following forms:

- either $x_i + s_j \bowtie c$,
- $x_i + s_{|\pi|} \bowtie x'_i$,
- $s_j - s_l \bowtie c$
- or $s_{|\pi|} - s_l \bowtie x'_i$

Choices of i, j, \bowtie and c depend on H . Here s_j is the time since the beginning of the word after j transitions, i.e. $s_j = \sum_{i=1}^j t_i$, in particular s_0 is the constant 0. Note that the Jacobian of the change of variables $\mathbf{t} = (t_1, \dots, t_{|\pi|})$ to $\mathbf{s} = (s_1, \dots, s_{|\pi|})$ is 1. So we can write:

$$V_\pi^2(\mathbf{x}, \mathbf{x}') = \int \mathbb{1}_{L(\pi)(\mathbf{x}, \mathbf{x}')}(t) dt = \int \prod_{H: \text{half-space}} \mathbb{1}_{H(\mathbf{s}, \mathbf{x}, \mathbf{x}')} d\mathbf{s},$$

thus, for any $i \in 1..d$:

$$\begin{aligned} \frac{\partial V_\pi^2(\mathbf{x}, \mathbf{x}')}{\partial x_i} &= \int \sum_{H_l: x_i \text{ appears}} \pm \delta(x_i + s_{j_l} - c_l \text{ or } x_i + s_{|\pi|} - x'_i) \prod_{H \neq H_l} \mathbb{1}_H d\mathbf{s} \\ &= \int \sum_{H_l: x_i \text{ appears}} \pm \prod_{H \neq H_l} \mathbb{1}_{H[x_i \leftarrow x'_i - s_{|\pi|} \text{ or } x_i \leftarrow c_l - s_{j_l}]} d\tilde{\mathbf{s}}. \end{aligned}$$

where $\tilde{\mathbf{s}}$ is the vector of coordinates of \mathbf{s} different from j_l for all l such that $H_l : x_i + s_{j_l} \bowtie_{k_l} c_{k_l}$. We note that the Dirac's δ s were all eliminated after the integration by s_{j_l} and that the expression that remains under the integral is a proper function of \mathbf{x}, \mathbf{x}' and $\tilde{\mathbf{s}}$. This implies that $V_\pi^2(\mathbf{x}, \mathbf{x}')$ was actually continuous with respect to x_i for all i and thus continuous.

D Proof of Proposition 8

Proof. Recall that:

$$f_{\mathcal{L}_1 \cdot \mathcal{L}_2}^2(\mathbf{x}, \mathbf{x}', z) = \sum_{w \in \Sigma^*} z^{|w|} V_{\mathcal{L}_1 \cdot \mathcal{L}_2(w)}^2(\mathbf{x}, \mathbf{x}').$$

Since the product is non ambiguous,

$$(\mathcal{L}_1 \cdot \mathcal{L}_2)(w) = \bigsqcup_{\substack{w_1, w_2 \in \Sigma^* \\ w = w_1 w_2}} \mathcal{L}_1(w_1) \cdot \mathcal{L}_2(w_2),$$

and thus

$$V_{\mathcal{L}_1 \cdot \mathcal{L}_2(w)}^2 = \sum_{\substack{w_1, w_2 \in \Sigma^* \\ w = w_1 w_2}} V_{\mathcal{L}_1(w_1) \cdot \mathcal{L}_2(w_2)}^2 = \sum_{\substack{w_1, w_2 \in \Sigma^* \\ w = w_1 w_2}} V_{\mathcal{L}_1(w_1)}^2 \star V_{\mathcal{L}_2(w_2)}^2,$$

with the last equality given by proposition 2. We deduce:

$$\begin{aligned}
\sum_{w \in \Sigma^*} z^{|w|} V_{\mathcal{L}_1 \cdot \mathcal{L}_2}^2(w) &= \sum_{w \in \Sigma^*} \sum_{\substack{w_1, w_2 \in \Sigma^* \\ w = w_1 w_2}} z^{|w_1|} V_{\mathcal{L}_1(w_1)}^2 \star z^{|w_2|} V_{\mathcal{L}_2(w_2)}^2 \\
&= \sum_{w_1 \in \Sigma^*} \sum_{w_2 \in \Sigma^*} z^{|w_1|} V_{\mathcal{L}_1(w_1)}^2 \star z^{|w_2|} V_{\mathcal{L}_2(w_2)}^2 \\
&= \sum_{w_1 \in \Sigma^*} z^{|w_1|} V_{\mathcal{L}_1(w_1)}^2 \star \sum_{w_2 \in \Sigma^*} z^{|w_2|} V_{\mathcal{L}_2(w_2)}^2,
\end{aligned}$$

and finally:

$$f_{\mathcal{L}_1 \cdot \mathcal{L}_2}^2 = f_{\mathcal{L}_1}^2 \star f_{\mathcal{L}_2}^2.$$