Bandwidth of Timed Automata: 3 Classes

Université Paris Cité, CNRS, IRIF, Paris, France

Aldric Degorre

□

□

Université Paris Cité, CNRS, IRIF, Paris, France

Cătălin Dima ⊠®

LACL, Université Paris-Est Créteil, France

Bernardo Jacobo Inclán

□

Université Paris Cité, CNRS, IRIF, Paris, France

Abstract

Timed languages contain sequences of discrete events ("letters") separated by real-valued delays, they can be recognized by timed automata, and represent behaviors of various real-time systems. The notion of bandwidth of a timed language defined in [17] characterizes the amount of information per time unit, encoded in words of the language observed with some precision ε .

In this paper, we identify three classes of timed automata according to the asymptotics of the bandwidth of their languages with respect to this precision ε : automata are either meager, with an O(1) bandwidth, normal, with a $\Theta\left(\log\frac{1}{\varepsilon}\right)$ bandwidth, or obese, with $\Theta\left(\frac{1}{\varepsilon}\right)$ bandwidth. We define two structural criteria and prove that they partition timed automata into these 3 classes of bandwidth, implying that there are no intermediate asymptotic classes. The classification problem of a timed automaton is PSPACE-complete.

Both criteria are formulated using morphisms from paths of the timed automaton to some finite monoids extending Puri's orbit graphs; the proofs are based on Simon's factorization forest theorem.

2012 ACM Subject Classification Theory of computation \rightarrow Quantitative automata; Theory of computation \rightarrow Timed and hybrid models

Keywords and phrases timed automata, information theory, bandwidth, entropy, orbit graphs, factorization forests

Related Version To appear (without appendix) in FSTTCS'23: DOI 10.4230/LIPIcs.FSTTCS.2023.6

Funding This work was supported by the ANR project MAVeriQ ANR-CE25-0012 and by the ANR-JST project CyPhAI.

Acknowledgements We thank the anonymous reviewers for their careful reading and valuable comments which helped to improve the paper.

1 Introduction

The study of the growth rate of formal languages was initiated in the 1950s in the pioneering [10], where the growth of the number of words of length n with respect to n in regular languages was analyzed. Even earlier, visionary Shannon [25] related word counts to the capacity of a discrete noiseless channel. In the early '60s, Kolmogorov [18] conceptualized the counting ("combinatorial") approach to the quantity of information, as compared to the probabilistic and the algorithmic ones. According to this approach, the quantity of information conveyed by an element of a finite set S is just $\log_2 \# S$ bits, and in most cases, asymptotics of this amount w.r.t. some size parameter is studied. Nowadays, counting-based asymptotic analysis of formal languages constitutes a background of the theory of codes [7], and of practical protocols [16, Chapter 6] implemented in every hard disk drive and DVD [15]. In a nutshell, one can encode a source language with a growth rate A (that is with $\sim 2^{An}$

words of size n) into a channel with growth rate B only if $A \leq B$. The growth rate is also closely related to the entropy rate of subshifts in symbolic dynamics [20].

In the long run, we are porting this approach to timed languages and automata, with both theoretical and practical ambitions. The intrinsic difficulty here is the continuous infinity of such languages and the infinite-state nature of timed automata. For growth rate (or information quantity) per event, we have mostly solved the problem in [3] and [2].

However, in the above, time was seen as data carried by the language, which contrasts with many natural applications, where time describes the actual execution time of the system under study. For this reason, in [17] we formalized the notion of bandwidth, the quantity of information per time unit, given some observation precision, and proved its actual relevance to bounded delay channel coding. Unfortunately, how this bandwidth can be computed for any timed automaton is not yet known. Thus, as a preliminary step, in this paper, we propose a rough qualitative classification of automata into three classes according to the asymptotic of the bandwidth w.r.t. precision. We believe that this qualitative result

- is of interest by itself and provides a relevant classification of qualitative behaviors of timed automata;
- is in line with other classification results for discrete languages;
- involves various and interesting techniques and tools;
- paves the way to the more precise computation of bandwidth.

As a comparison, for multiple classes of languages, the growth rate provides a relevant classification of languages into big (exponential) and small (polynomial) ones, giving important behavioral insights; a gap between polynomial and exponential growth in regular languages is immediate from [10], for context-free languages it was established in [9]. An efficient algorithm for distinguishing between the polynomial and exponential growth of regular and context-free languages is presented in [12]. Finally, for indexed languages, there is no such gap and an intermediate growth is possible [13]. In the same vein but for a slightly different aspect, in [27], nondeterministic finite automata are classified into three classes: finitely, polynomially, and exponentially ambiguous; and the classes are characterized by the presence of specific patterns, as in our study.

For timed regular languages, the growth rate of their volume with respect to the number of events was explored in [3]. In particular, a dichotomy between thin and thick languages was established and related to the Zeno phenomena. In a nutshell, in a thick language (with exponential growth) most timed words are non-Zeno, and have nice properties (they satisfy a pumping lemma, admit discretization, etc.), while in a thin language, all timed words long enough are, in some sense, Zeno and involve very short delays.

In this paper, as already said, we look at the growth rate of timed languages **per time unit**. More precisely, instead of volumes, we study the size of the discretizations of the language, up to precision ε , using ε -nets and ε -separated sets [19]. The logarithms of these quantities are called, resp., ε -entropy and ε -capacity, and the limits of these quantities divided by the time bound of the language are called, resp., entropic and capacitive bandwidth. In [17] we justify the notion of bandwidth proving (up to some details) that a timed source S can be encoded to a timed channel C with bounded delay, iff the bandwidth of S does not exceed the bandwidth of C.

In this article, we partition timed regular languages (and timed automata) into three classes according to the asymptotic behavior of the bandwidth, as ε becomes small. We call them *meager*, with an O(1) (smallest) bandwidth, *normal*, with a $\Theta\left(\log\frac{1}{\varepsilon}\right)$ bandwidth, or *obese*, with $\Theta\left(\frac{1}{\varepsilon}\right)$ (highest) bandwidth. Intuitively, in an obese language, information can be encoded into time delays with a very high frequency; in a normal language, information can

be encoded every few time units; finally, a meager language is extremely constrained and information can be mostly encoded into discrete choices. This trichotomy can be deduced from two structural criteria on timed automata, which exactly characterize the asymptotic classes. Using these structural criteria we also prove that deciding to which class belongs a timed automaton is PSPACE-complete.

Our results build on several ingredients. The bandwidth of a timed language has been defined in [17], itself based on notions of ε -entropy and ε -capacity of functional spaces as studied in [19] and on a pseudo-distance on timed words from [4]. Structural information on timed automata is formulated in terms of orbit graphs from [22], which are discrete objects summarizing reachability relations between states. We had to enrich these graphs with some qualitative information in the form of elements from two finite semirings which represent (1) a rough estimate of the delay between symbols in a timed word and (2) how much choice one may have when taking an edge between two vertices. Structural criteria of meagerness and obesity are formulated in terms of those labels. More precisely, we identify non-meagerness and obesity patterns, and prove that a timed regular language is non-meager resp. obese, whenever it is recognized by a timed automaton with a cycle labeled with an orbit graph containing a corresponding pattern.

The lower bounds take thus the form "whenever a cycle with a pattern is present, the bandwidth is high", and we prove them by providing a direct construction of a big bunch of ε -separated runs along this cycle. The proofs of upper bounds "whenever a cycle with a pattern is absent, the bandwidth is low" are much more involved and rely on Simon's factorization forest theorem, a powerful result relating Ramsey style combinatorics and formal languages theory (see the original article [26] and a modern presentation in [8]). We pinpoint particular properties of Simon factorization whenever one of the non-meagerness or obesity patterns is absent, and build a small ε -net guided by this factorization.

The paper proceeds as follows: in Sect. 2, we recall some notions and tools, and state the problem of classification of timed automata according to their bandwidth. In Sections 3 and 4, we solve the problem for deterministic timed automata by establishing two structural criteria — for very low and very high bandwidth. In Sect. 5, we come up with a classification theorem (and establish the complexity of classification), extend it to the non-deterministic case, and compare our new classes with those from [3]. We conclude with some perspectives. For sake of readability, many technical details and proofs are relegated to the Appendix.

2 Background and Problem Statement

We start this section by recalling our approach [17] to bandwidth (information by time unit) in timed languages and its background: classical ε -entropy and capacity [19] and our pseudo-distance on timed words [4]. Next, we recall timed automata [1] and related constructions, including variants of the region-split form [3] and orbit graphs from [22]. Finally, we present the central notion of the paper: the three classes of timed automata.

2.1 Measuring information in timed languages

Given Σ , a finite alphabet of discrete events, a timed word over Σ is an element from $(\Sigma \times \mathbb{R}_+)^*$ of the form $w = (a_1, t_1) \dots (a_n, t_n)$, with $0 \le t_1 \le t_2 \dots \le t_n$. Number t_i should be interpreted as the date at which the event a_i happens. We always put $t_0 = 0$. For any timed word $w = (a_1, t_1) \dots (a_n, t_n)$, we denote its discrete length as $|w| \triangleq n$ and its duration as $\tau(w) \triangleq t_n$. The timing of w is the sequence $t_1 \dots t_n$, and its untiming is the sequence $a_1 \dots a_n$. A timed language over Σ is a set of timed words over Σ .

4 Bandwidth of Timed Automata: 3 Classes



Figure 1 The set of black dots is ε -separated and an ε -net in the segment

For determining the quantity of information contained in a timed language, we adopt the approach introduced in [19] (for compact functional spaces). Intuitively, when only words that are very close to one another can be observed, then one may not be able to distinguish many different words, thus little information may be conveyed. On the contrary, if there are many words that are set sufficiently far from each other, the fact of choosing among them conveys a large amount of information.

A second point of view is how many words are necessary to approximate the language up to some precision. Both points of view depend on the precision with which we observe, as formalized by the following definition, taken from [19] and adapted to pseudo-metric spaces¹:

- ▶ **Definition 1.** Let (U, d) be a pseudo-metric space and $A \subseteq U$, then:
- $M \subseteq A$ is an ε -separated subset of A if $\forall x \neq y \in M, d(x,y) > \varepsilon$;
- $N \subseteq U$ is an ε -net of A if $\forall y \in A, \exists s \in N$ s.t. $d(y,s) \leq \varepsilon$.

An ε -separated subset of A contains elements of A distant from each other by at least ε , hence distinguishable with precision ε . An ε -net of A allows approximating every element of A with precision ε , see Fig. 1. They give rise to two measures of the information content:

▶ **Definition 2.** For (U,d) a pseudo-metric space and $A \subseteq U$, the ε -capacity $C_{\varepsilon}(A)$ and ε -entropy $\mathcal{H}_{\varepsilon}(A)$ are defined as follows²:

$$C_{\varepsilon}(A) \triangleq \log \max\{\#M \mid M \text{ } \varepsilon\text{-separated set of } A\}; \text{ } \mathcal{H}_{\varepsilon}(A) \triangleq \log \min\{\#N \mid N \text{ } \varepsilon\text{-net of } A\}.$$

The following inequality [19] is known to hold for any A subset of U:

$$C_{2\varepsilon}(A) \le \mathcal{H}_{\varepsilon}(A) \le C_{\varepsilon}(A). \tag{1}$$

Two basic cases provide an intuition on the sense of $\mathcal C$ and $\mathcal H$ as information measures:

- For Σ a finite alphabet (with the discrete metrics), $C_{\varepsilon}(\Sigma) = \mathcal{H}_{\varepsilon}(\Sigma) = \log \# \Sigma$; in words, a letter in Σ brings $\log \# \Sigma$ bits of information.
- Consider now a segment [0,1], then $C_{\varepsilon}[0,1] \approx \mathcal{H}_{\varepsilon/2}[0,1] \approx \log 1/\varepsilon$. In words, a real number observed with precision ε contains $\log 1/\varepsilon$ bits of information, intuitively corresponding to $\log 1/\varepsilon$ digits after the binary point.

In the developments thereafter, U will typically be U_{Σ} , the universal timed language on the working alphabet Σ , while different regular timed languages on Σ will play the role of A. The pseudo-metric will be the one we describe next.

To give a meaningful interpretation of capacity and entropy, an appropriate metric modeling the ability of the observer is required, that is, the ability to distinguish between two words. To this purpose, we imagine an observer that reads the discrete letters of the word exactly (they can determine whether or not a letter has occurred) but with some imprecision

¹ In pseudo-metric spaces one may have d(x,y) = 0 for $x \neq y$, but all other axioms of the distance hold. Moreover we also allow infinite distances.

 $^{^{2}\,}$ all logarithms in this article are base 2

Figure 2 Pseudo-distance between two timed words (dotted lines with arrows represent directed matches between letters). $\overrightarrow{d}(u,v) = 0.2$; $\overrightarrow{d}(v,u) = 0.3$, thus d(u,v) = 0.3.

w.r.t. time, so they cannot determine when two letters are very close to one another, which one came before the other, and not even how many times a letter was repeated within a short interval. Hence, we use the following pseudo-distance (similar to the Hausdorff distance between sets):

▶ **Definition 3** ([4]). The pseudo-distance d between two timed words $w = (a_1, t_1) \dots (a_n, t_n)$ and $v = (b_1, s_1) \dots (b_m, s_m)$ is defined by (with the convention $\min \emptyset = \infty$)

$$\overrightarrow{d}\left(w,v\right) \triangleq \max_{i \in \{1..n\}} \min_{j \in \{1..m\}} \{|t_i - s_j| : a_i = b_j\}; \qquad d(w,v) \triangleq \max(\overrightarrow{d}\left(w,v\right), \overrightarrow{d}\left(v,w\right)).$$

The definition is illustrated on Fig. 2. We remark that d(w,v) = 0 whenever w and v only differ by order and quantity of simultaneous letters, e.g. for w = (a,1)(b,1) and v = (b,1)(b,1)(a,1). For this reason, d is only a pseudo-distance.

There are two (related) reasons for using this pseudo-distance rather than more straightforward "uniform" distances such as defined in [14]: (1) it provides non-trivial values even when the two words w and v have a different number of events; (2) the set of timed words with duration bounded by some T is compact, which ensures that finite ε -nets always exist.

We now recall the notion of bandwidth, as introduced in [17], a characterization of the ability of a language to produce or convey information per time unit in the long run. Bandwidths come in two versions, depending on whether it is about producing or transmitting information. Bandwidths are asymptotics, with respect to time, of the growth rate of ε -capacity or ε -entropy, which finally yield quantities expressed in bits per time unit. We use here the notion of timed slice of a language L at some $T \in \mathbb{R}_+$, defined as $L_T \triangleq \{w : w \in L, \tau(w) \leq T\}$.

▶ **Definition 4** ([17]). The ε -entropic and ε -capacitive bandwidths of a language L are

$$\mathcal{BH}_{\varepsilon}(L) \triangleq \limsup_{T \to \infty} \frac{\mathcal{H}_{\varepsilon}(L_T)}{T}; \qquad \mathcal{BC}_{\varepsilon}(L) \triangleq \limsup_{T \to \infty} \frac{\mathcal{C}_{\varepsilon}(L_T)}{T}.$$

The following relationship between the two bandwidths is immediate from (1):

$$\mathcal{BC}_{2\varepsilon}(L) \le \mathcal{BH}_{\varepsilon}(L) \le \mathcal{BC}_{\varepsilon}(L).$$
 (2)

2.2 Timed Automata

Regular timed languages are timed languages recognized by timed automata introduced in [1]. In most of this paper, as in [3,22] we deal with deterministic timed automata (DTA). For a set of variables Ξ , G_{Ξ} denotes the set of constraints expressible as conjunctions of inequalities of the form $\xi \sim b$ with $\xi \in \Xi$, $\sim \in \{<, \leq, >, \geq\}$ and b an integer ≥ 0 .

- ▶ **Definition 5.** A timed automaton is a tuple $(Q, X, \Sigma, \Delta, S, I, F)$, where
- *Q* is the finite set of discrete locations;

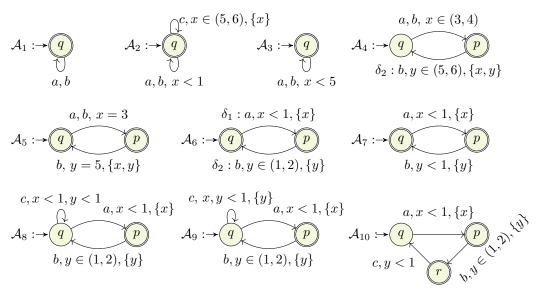


Figure 3 10 running examples of timed automata, $I(q) = \{0\}$, $F(\cdot) = \mathbf{true}$ for marked locations.

- X is the finite set of clocks;
- \blacksquare Σ is a finite alphabet;
- $S, I, F: Q \rightarrow G_X$ define respectively the starting, initial, and final clock values for each location;
- $\Delta \subseteq Q \times Q \times \Sigma \times G_X \times 2^X$ is the transition relation, whose elements are called edges. A timed automaton is deterministic if $\{(q,x)|x \models I(q)\}$ is a singleton and for any two edges $(q,q_1,a,\mathfrak{g}_1,\mathfrak{r}_1)$ and $(q,q_2,a,\mathfrak{g}_2,\mathfrak{r}_2)$ with $q_1 \neq q_2$, the constraint $\mathfrak{g}_1 \wedge \mathfrak{g}_2$ is non-satisfiable.

The role of S, I, F is the following: S(q) denotes the "starting constraint" which must be satisfied by clock values when entering location q. Accepting runs should start in configuration (q, x) with x satisfying I(q) and end by a transition to (q', x') with x' satisfying F(q').

Throughout the paper, we use as running examples the 10 automata described on Fig. 3.

2.2.1 Semantic details and notations

We define two operations that can be applied to any clock vector $x \in [0, \infty)^X$: first, given $t \geq 0$, the *timed successor* of x by t, denoted $x + t \in [0, \infty)^X$, is the clock vector such that for all clocks $c \in X$, $(x + t)_c = x_c + t$; second, given a set of clocks $\mathfrak{r} \subset X$, the \mathfrak{r} -reset of x is the clock vector $x[\mathfrak{r}] \in [0, \infty)^X$ defined as $x[\mathfrak{r}]_c = 0$ whenever $c \in \mathfrak{r}$ and $x[\mathfrak{r}]_c = x_c$ otherwise.

The following objects will be associated with a timed automaton, and used in the sequel (for all types of sequences, \circ denotes the concatenation and ϵ the empty sequence; the concatenation is lifted in the usual way to sets of sequences, seen as languages):

- For any $\delta \in \Delta$ we define the projections src_{δ} (source), dst_{δ} (destination), lbl_{δ} (label), \mathfrak{g}_{δ} (guard) and \mathfrak{r}_{δ} (set of reset clocks) such that $\delta = (src_{\delta}, dst_{\delta}, lbl_{\delta}, \mathfrak{g}_{\delta}, \mathfrak{r}_{\delta})$.
- Edges are elements of Δ , edge sequences are elements of Δ^* , paths are edge sequences $\delta_1 \dots \delta_n$ with $dst_{\delta_i} = src_{\delta_{i+1}}$ for all i.
- A state is a pair of a discrete location and of a clock vector, i.e. $(q, x) \in Q \times [0, \infty)^X$.
- A run is a sequence of the form: $\rho = (q_0, x_0) \xrightarrow{(\delta_1, t_1)} (q_1, x_1) \dots \xrightarrow{(\delta_n, t_n)} (q_n, x_n)$, such that for each i it holds that (by convention $t_0 = 0$)

$$x_i \models S(q_i); src_{\delta_i} = q_{i-1}; dst_{\delta_i} = q_i; x_{i-1} + t_i - t_{i-1} \models \mathfrak{g}_{\delta_i}; x_i = \mathfrak{r}_{\delta_i}(x_{i-1} + t_i - t_{i-1}).$$
 (3)

- The following projections from runs to locations, paths, and timed words are defined: $src(\rho) \triangleq (q_0, x_0), dst(\rho) \triangleq (q_n, x_n), path(\rho) \triangleq \delta_1 \dots \delta_n, word(\rho) \triangleq (lbl_{\delta_1}, t_1) \dots (lbl_{\delta_n}, t_n).$
- If ρ_1 and ρ_2 are runs, $\rho_1 \circ \rho_2$ is defined provided: either $\rho_1 = \epsilon$, $\rho_2 = \epsilon$ or $dst(\rho_1) = src(\rho_2)$.
- To any edge sequence π we associate its set of runs $Runs(\pi) \triangleq \{\rho : path(\rho) = \pi\}$. This implies $Runs(\pi) \neq \emptyset$ only when π is a path. Also $Runs(\pi_1 \circ \pi_2) = Runs(\pi_1) \circ Runs(\pi_2)$.
- Given a path π and two clock vectors x, x' we define $Runs(\pi, x, x')$ as the subset of $Runs(\pi)$ starting with clock values x and ending with clock value x'. We denote the timed language $\{word(\rho)|\rho \in Runs(\pi, x, x')\}$ by $L_{\pi}(x, x')$.
- We also define the language $L_{\bar{\pi}}(x,x')$, where all the guards are replaced by their closures.³
- A timed automaton \mathcal{A} defines a timed language $L(\mathcal{A})$: a timed word w belongs to $L(\mathcal{A})$ iff it is in $L_{\pi}(x, x')$ for some clock vectors x and x' and some path π going from some location q to some location q' with $x \models I(q)$ and $x' \models F(q')$ (so we require also the first state to be initial and the last one to be accepting).

For a small example consider a 3-edge path $\delta_1 \delta_2 \delta_1$ in \mathcal{A}_6 . One of its runs is

$$(q,0,0) \stackrel{(\delta_1,0.8)}{\to} (p,0,0.8) \stackrel{(\delta_2,1.5)}{\to} (q,0.7,0) \stackrel{(\delta_1,1.7)}{\to} (p,0,0.2),$$

and the corresponding timed word is (a, 0.8), (b, 1.5), (c, 1.7).

2.3 Region Abstraction and Region-Splitting

First, let us denote by M the maximal constant appearing in the constraints used in the automaton. Also, for $\xi \in \mathbb{R}$, we denote $\{\xi\}$ its fractional part.

- ▶ **Definition 6** ([1]). For a clock vector x, we define $x_{\infty} = \{c \in X | x_c > M\}$ and $x_{\mathbb{N}} = \{c \in X | x_c \in \mathbb{N}\}$. Two clock vectors $x, y \in \mathbb{R}_+^X$ are region-equivalent whenever
- $x_{\infty} = y_{\infty} \text{ and } x_{\mathbb{N}} \setminus x_{\infty} = y_{\mathbb{N}} \setminus y_{\infty};$
- $\forall c \in X \setminus x_{\infty}, \lfloor x_c \rfloor = \lfloor y_c \rfloor;$
- $and \forall c_1, c_2 \in X \setminus x_{\infty}, \{x_{c_1}\} \leq \{x_{c_2}\} \text{ iff } \{y_{c_1}\} \leq \{y_{c_2}\}.$

A region is an equivalence class of this relation.

A region R is M-bounded whenever $R \subseteq [0, M]^X$. Any M-bounded region R is a simplex of some dimension $d \le \#X$. We denote by V(R) the set of all d+1 vertices of R.

A folklore result says that each timed automaton can be transformed into a normal form, which can be considered as a (timed) variant of the region graph defined in [1]. The following definition adapts this to our framework while ensuring that starting constraints are bounded:

- ▶ **Definition 7.** A region-split M-bounded TA (RsTA) is a DTA $(Q, X, \Sigma, \Delta, S, I, F)$, such that, for any location $q \in Q$:
- the starting constraint S(q) defines a non-empty M-bounded region;
- **a** all states in $\{q\} \times S(q)$ are reachable from the initial state and co-reachable to a final state:
- either I(q) = S(q) is a singleton⁴ or $I(q) = \emptyset$;
- for any edge $(q, q', a, \mathfrak{g}, \mathfrak{r}) \in \Delta$, it holds that $(\{S(q) + t \mid t \in \mathbb{R}_+\} \cap \mathfrak{g}) [\mathfrak{r}] = S(q')$.
- ▶ Proposition 8. For any DTA, there exists an RSTA accepting the same language, with the same number of clocks, and an exponentially larger number of locations.

 $^{^{3}}$ Note that closing guards in an automaton may create new paths but doing so for a single path is "safe".

⁴ by definition of DTA this is possible for a unique location q

8

2.4 Orbits Monoid

Now we define the monoid, based on Puri's orbit graphs [22], and summarizing reachability along paths of an RsTA. We fix some numbering of vertices of regions: for a location q, such that the simplex of the region S(q) has dimension d_q , we have $V(S(q)) = \{v_1^q, \ldots, v_{d_n+1}^q\}$.

▶ **Definition 9.** The finite monoid \mathbf{M} of p-orbits contains $\mathbf{0}, \mathbf{1}$ and, for any two locations q and q' all triples $\langle q, A, q' \rangle$ with A a Boolean $(d_q + 1) \times (d_{q'} + 1)$ -matrix. All p-orbits with zero matrices are identified with $\mathbf{0}$. Composition rules for $\mathbf{0}$ and $\mathbf{1}$ are standard; and

$$\langle p,A,p'\rangle\,\langle q,B,q'\rangle = \begin{cases} \langle p,AB,q'\rangle & \textit{if } p'=q;\\ \mathbf{0} & \textit{otherwise} \end{cases}$$

(matrices are multiplied using Boolean operations⁵).

It is convenient to visualize a p-orbit as a bipartite graph (or an oriented graph whenever q = q'), from vertices of S(q) to those of S(q'), with edges corresponding to 1s in the matrix. Following [22]⁶ we associate a p-orbit to each edge sequence.

- ▶ **Definition 10** (The p-orbit of a path summarizes reachability). Given an RSTA, we define a function γ from edge sequences to M:
- \mathbf{r} $\gamma(\epsilon) = \mathbf{1}$ and whenever sequence π is not a path, $\gamma(\pi) = \mathbf{0}$;
- for π a non-empty path from q to q' let $\gamma(\pi) = \langle q, A, q' \rangle$, with $A_{ij} = 1$ iff $L_{\bar{\pi}}(v_i^q, v_j^{q'}) \neq \emptyset$. In graph terms, there is an edge in the orbit graph of π from a vertex v to v' iff (q', v') is reachable from (q, v) along (the closure of) π .
- ▶ **Proposition 11** (probably [3]). γ is a monoid morphism.

Thanks to this property, we can easily compute $\gamma(\pi)$: for one edge using the definition, for a longer path by composition: $\gamma(\delta_1 \dots \delta_k) = \gamma(\delta_1) \dots \gamma(\delta_k)$.

It follows from the theory developed in [22], that p-orbits are an efficient abstraction for reachability: for each path π , its p-orbit $\gamma(\pi)$ fully describes the reachability relation between a clock vector x in the starting region of π and a clock vector x' in its ending region, through a run along π . Intuitively, the reason is that any run from (q, x) to (q', x') can be expressed as a convex combination of runs of surrounding vertices of S(q) to vertices of S(q'), summarized in the orbit $\gamma(\pi)$.

Consider the running example \mathcal{A}_6 , its RsTA has its main cycle⁷ between q and p with $S(q) = (0,1) \times \{0\}, S(p) = \{0\} \times (0,1)$. The p-orbits of its edges are the two graphs on the left of Fig. 4, and their matrix form is:

$$\gamma(q \to p) = \gamma_1 = \left\langle q, \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, p \right\rangle; \ \gamma(p \to q) = \gamma_2 = \left\langle p, \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, q \right\rangle.$$

The orbits of the two cycles, i.e. the two graphs on the right of Fig. 4, are the following:

$$\gamma(q \to p \to q) = \gamma_1 \cdot \gamma_2 = \left\langle q, \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, q \right\rangle; \ \gamma(p \to q \to p) = \gamma_2 \cdot \gamma_1 = \left\langle p, \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, p \right\rangle.$$

⁵ that is $(AB)_{ij} = \bigvee_{k} (A_{ik} \wedge B_{kj})$

⁶ in fact, [22] only considers cyclic paths in the region automaton

⁷ there are two more transient locations, not considered here

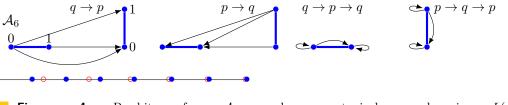


Figure 4 P-orbits for A_6 and a typical word in $L(A_6)$: (a, .9)(b, 1.2)(a, 1.8)(b, 2.4)(a, 2.7)(b, 3.5)(a, 3.6)(b, 4.52)(a, 4.59)(b, 5.53)(a, 5.58)(b, 6.535)(a, 6.575)

2.5 Central notion of the paper: the three classes

```
▶ Definition 12. A timed language L is meager whenever \mathcal{BH}_{\varepsilon}(L) = O(1); normal whenever \mathcal{BH}_{\varepsilon}(L) = \Theta(\log \frac{1}{\varepsilon}); obese whenever \mathcal{BH}_{\varepsilon}(L) = \Theta(\frac{1}{\varepsilon}), as \varepsilon \to 0.
```

It follows immediately from (2) that a similar and equivalent characterization can be done in terms of capacity by replacing \mathcal{BH} by \mathcal{BC} . We will now give some intuition concerning the three classes, using running examples.

Automata with obese language: information is produced with a high frequency. Thus, in \mathcal{A}_1 every ε seconds one can produce an a, a b, both or nothing, which corresponds to two bits of information, and yields a bandwidth of $2/\varepsilon$ bit/sec. In \mathcal{A}_2 such a high-frequency encoding is possible during 1 second, after that one should wait 4 seconds and take the c loop to reset x. In total, the bandwidth will be $\frac{2}{5\varepsilon}$.

Automata with normal language: during one unit of time they typically make O(1) discrete choices, and O(1) real-valued choices. The former convey O(1) bit/sec, the latter $O(\log 1/\varepsilon)$. Thus \mathcal{A}_4 makes a cycle in 5 to 6 seconds, with two real-valued choices (duration of stay in p and in q), and the bandwidth is $\frac{2}{5} \log 1/\varepsilon$.

Automata with meager language: In the long run such automata should not allow real-valued choices. A_5 has only discrete choices since its transitions happen at discrete dates only. A_3 can produce a huge amount of information $(2/\varepsilon)$ bits every second), but only during the first five seconds of its life. Its bandwidth is $\lim_{T\to\infty} 2/\varepsilon T = 0$. The automaton A_6 is much less evident. It requires a and b to interleave, with a being < 1 second apart and b on the contrary > 1 second apart, as shown on Fig. 4. Such interleaving is possible for any duration T, but real-valued intervals between symbols become more and more constrained. We prove later that this language is indeed meager.

In the rest of the paper, we propose a structural and decidable characterization of DTA establishing membership of their languages into one of these 3 classes, also proving that no other possibility exists.

3 Distinguishing between Meager and Other Automata

For the next two sections, our plan is to characterize the bandwidth of the timed language generated by a cycle looping over a region and to extend this analysis to the whole language. As bandwidth is about choice per unit of time, we enrich the monoid of p-orbits with one of the following two types of information: (1) the amount of choice one may have when taking an edge between two vertices, or (2) a rough estimate of the interval of durations of timed words generated by the cycle, starting in one vertex and ending in another vertex. The first

type of information is useful for discriminating meager from non-meager languages, while the second type will distinguish obese from non-obese languages. We start with the former, slightly simpler to formulate.

3.1 **Orbits with Abstracted Freedom**

The structural criterion for meagerness is based on an enrichment of the monoid of p-orbits in which matrix entries summarize the dimension of the reachability polytope between vertices. In more detail, matrix entries belong to $\Lambda_f \triangleq \{0, \mathbf{narrow}, \mathbf{wide}\}$, having a semiring structure⁸, defined in Table 1. The intuition is the following. Elements of Λ_f summarize reachability polytopes between region vertices through some path: no reachability for 0 (empty polytope), reachability without continuous choice for **narrow** (singleton polytope), presence of at least one continuous choice for wide (polytope of dimension ≥ 1). The addition of Λ_f represents the convex union of two compatible sets of runs, while the multiplication corresponds to the concatenation of sets of runs, in particular, two distinct singletons logically add up to a one-dimensional polytope, while the dimension of two concatenated sets of runs is the sum of the dimensions. Hence the monoid describes the dimension of the reachability polytope of a path: its degree of freedom.

+	0	narrow	wide
0	0	narrow	wide
narrow	narrow	wide	wide
\mathbf{wide}	wide	wide	wide

×	0	narrow	wide
0	0	0	0
narrow	0	narrow	\mathbf{wide}
wide	0	wide	\mathbf{wide}

- **Table 1** Addition and multiplication of the semiring Λ_f
- ▶ **Proposition 13.** $(\Lambda_f, +, \times, 0, \mathbf{narrow})$ is a commutative semiring.

Thus the definition of the monoid \mathbf{M}_f of f-orbits is identical with Def. 9, with the only difference, that matrix entries belong to the semi-ring Λ_f . Graphically, we represent all non-0 elements of Λ_f by labeled (or colored) edges, and 0 by an absence of edge, see Fig. 5.

- ▶ **Definition 14** (The f-orbit of a path abstracts its freedom). Given an RSTA, we define a function γ_f from edge sequences to \mathbf{M}_f .
- $\gamma_f(\epsilon) = 1$. Whenever sequence π is not a path, $\gamma_f(\pi) = 0$.
- For π a non-empty path from q to q', let $\gamma_f(\pi) = \langle q, B, q' \rangle$, with the following matrix entries: for vertices $v = v_i^q \in V(S(q))$ and $v' = v_i^{q'} \in V(S(q'))$
 - $B_{ij} = \mathbf{0} \text{ iff } L_{\bar{\pi}}(v, v') = \emptyset;$
 - $B_{ij} = \mathbf{narrow} \ iff \ L_{\bar{\pi}}(v, v') \ is \ a \ singleton;$
 - $B_{ij} =$ wide otherwise.

We remark that in the last case $L_{\bar{\pi}}(v, v')$ has a dimension ≥ 1 .

▶ Proposition 15. γ_f is a monoid morphism.

Again, this property allows compositional computation of $\gamma_f(\pi)$.

also known as super-Boolean semiring \$B, [23]

Def. 14 may not seem effective but, for a single edge δ , $\gamma_f(\delta)$ can be computed in polynomial time in #X, by solving the constraints of δ , while $\gamma_f(\cdot)$ for longer paths can be computed using the multiplication.

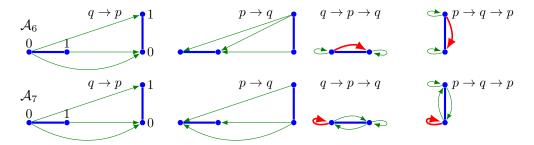


Figure 5 F-orbits for A_6 and A_7 ; green thin arrows are **narrow**, red thick ones **wide**.

▶ Definition 16. A cycle of an RSTA is structurally meager if its f-orbit has no wide self-loop (in matrix terms, a wide on the main diagonal). An RSTA is structurally meager when all its cycles are structurally meager.

Consider the running example \mathcal{A}_6 . The top row of Fig. 5 (which is a colorful version of Fig. 4) illustrates f-orbits $\gamma_f(q \to p) = \gamma_1'$ and $\gamma_f(p \to q) = \gamma_2'$, in matrix form:

$$\left\langle q, \begin{pmatrix} \mathbf{narrow} & \mathbf{narrow} \\ \mathbf{narrow} & 0 \end{pmatrix}, p \right\rangle; \text{ and } \left\langle p, \begin{pmatrix} 0 & \mathbf{narrow} \\ \mathbf{narrow} & \mathbf{narrow} \end{pmatrix}, q \right\rangle.$$

and the f-orbits of the cycles $q \to p \to q$ and $p \to q \to p$, i.e. the products $\gamma_1' \gamma_2'$ and $\gamma_2' \gamma_1'$:

$$\left\langle q, \begin{pmatrix} \mathbf{narrow} & \mathbf{wide} \\ 0 & \mathbf{narrow} \end{pmatrix}, q \right\rangle; \text{ and } \left\langle p, \begin{pmatrix} \mathbf{narrow} & 0 \\ \mathbf{wide} & \mathbf{narrow} \end{pmatrix}, p \right\rangle.$$

Similarly, the bottom row of Fig. 5 presents f-orbits of A_7 , and one can observe that A_6 is structurally meager (no **wide** self-loops), while A_7 is not.

The decision procedure for structural meagerness uses the following "pumping lemma":

- ▶ Lemma 17 (adapted from [24, Lem. 8.7.1]). Given a finite alphabet Γ , a finite monoid \mathbf{M} , and a morphism $\beta : \Gamma^* \longrightarrow \mathbf{M}$, for any $e \in \mathbf{M}$ having $\beta^{-1}(e) \neq \emptyset$, there exists a word $w \in \beta^{-1}(e)$ of a length $\leq \#\mathbf{M}$.
- ▶ **Theorem 1.** Structural meagerness of an RSTA of description size n with m clocks is decidable in $SPACE(log^2(n)poly(m))$.

Here poly stands for a polynomial. We will also use exp for an exponential function below.

Proof sketch. The RsTA has at most n locations, each orbit can be represented as a couple of locations and an (at most) $(m+1) \times (m+1)$ -matrix with elements in Λ_f (where $\#\Lambda_f = 3$). Thus $\#\mathbf{M}_f \leq n^2 \cdot 3^{(m+1)^2} = \mathsf{poly}(n)\mathsf{exp}(m)$. An automaton is structurally meager whenever it contains no cyclic path π with orbit $\gamma_f(\pi)$ having a **wide** entry on the matrix diagonal. By Lem. 17, only cycles up to length $\#\mathbf{M}_f$ need to be checked. We use now the reachability method [21, Sect. 7.3] to detect such a cycle in a space-efficient way, see Listing 1. An auxiliary function <code>isAPathOrbit(e, h)</code>, true iff there is a path $\pi \in \gamma_f^{-1}(e)$ of length $\leq 2^h$ in the RsTA, is computed by a divide-and-conquer algorithm. The algorithm requires a call stack of depth $\log \#\mathbf{M}_f$, the size of each stack frame is $O(\log n) + \mathsf{poly}(m)$, thus the computation takes space $(O(\log n) + \mathsf{poly}(m)) \cdot \log \#\mathbf{M}_f = \log^2(n)\mathsf{poly}(m)$.

Listing 1 Deciding structural meagerness

```
Boolean function is APathOrbit(e, h)
    if h = 0 return (e=1 || \exists \delta : \gamma_f(\delta) = e)
    for all orbits e1, e2
         if e1 \cdot e2 = e \&\& isAPathOrbit(e1, h-1) \&\& isAPathOrbit(e2, h-1)
              return true
    return false
 Boolean function isStrucMeager()
    for all cyclic orbits e
         if (containsWideOnDiag(e) && isAPathOrbit(e, \log \#\mathbf{M}_f))
              return false
    return true
```

Structural Meagerness ⇔ **Meagerness**

The upper bound can be formulated as follows.

▶ Proposition 18. The timed language of a structurally meager RSTA is meager.

The proof (see Appendix) is done in several steps, two of them using Simon's theorem. Let A be structurally meager. First, we explore the languages L((q, x), (q, x)), of words, looping from a state to itself, and observe that, for any structurally meager cycle π , $L_{\pi}(x,x)$ is a singleton. Intuitively, words of L((q,x),(q,x)) do only discrete choices, and thus convey only O(1) bit/second of information.

Next, we make a finite partition of all the regions into small (ε -sized) cells and define a set $Still_{\varepsilon}$ of words accepted along a cycle from such a cell to the same cell. Such words are close to singletons and thus convey O(1) bit/second of information when observed with precision ε . On the other hand, visits to different cells by the same cycle (and even by different cycles with the same orbit) always respect a certain partial order \succeq over cells.

Last, we use Puri's reachability characterization [22] and our technique of linear Lyapunov functions [3], together with Simon's theorem to factorize any word in $L_T(\mathcal{A})$ in factors that are either one letter or belong to $Still_{\varepsilon}^{s}$. The total number of factors does not depend on T, but only on ε . This factorization makes it possible to build a small ε -net and conclude.

On the other hand, we need a lower bound.

▶ Proposition 19. If an RsTA is not structurally meager then its bandwidth is $\Omega(\log(1/\varepsilon))$.

The proof (see Appendix) idea is to take a cycle that is not structurally meager, and show that it alone provides a bandwidth of $\log(1/\varepsilon)$. The main theorem of this section is now immediate from Props. 18 and 19:

▶ Theorem 2. The language of an RSTA is meager iff the RSTA is structurally meager.

4 Distinguishing between Obese and Other Automata

4.1 **Orbits with Abstracted Duration**

The structural criterion for obesity is based on another enrichment of p-orbits, in which the entries of the matrix A belong to a finite semiring $\Lambda_d \triangleq \{\text{instant}, \text{fast}, \text{slow}, 0\}$, with operations detailed in Table 2. Intuitively, the elements of Λ_d represent 4 abstracted classes of durations for sets of runs having the same path, same source, and same destination vertices;

+	instant	fast	slow	0
instant	instant	fast	fast	instant
fast	fast	\mathbf{fast}	fast	\mathbf{fast}
slow	fast	\mathbf{fast}	slow	slow
0	instant	fast	slow	0

×	instant	fast	slow	0
instant	instant	fast	slow	0
fast	fast	fast	slow	0
slow	slow	slow	slow	0
0	0	0	0	0

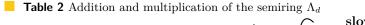




Figure 6 Obesity patterns. Left: Type I. Right: Type II.

similarly to Λ_f , the addition of Λ_d represents the convex union of two compatible sets of runs, while the multiplication corresponds to the concatenation of sets of runs.

▶ **Proposition 20.** $(\Lambda_d, +, \times, 0, \mathbf{instant})$ is a commutative semiring.

Similarly to \mathbf{M}_f in the meager case, we define \mathbf{M}_d , the monoid of d-orbits, where the only difference is that matrix entries belong to Λ_d instead of Λ_f . As in previous cases, we represent elements of \mathbf{M}_d as labeled graphs.

- ▶ **Definition 21** (The d-orbit of a path abstracts its speed). Given an RSTA, we define a function γ_d from edge sequences to \mathbf{M}_d .
- $\gamma_d(\epsilon) = 1$. Whenever sequence π is not a path, $\gamma_d(\pi) = 0$.
- For π a non-empty path from q to q' let $\gamma_d(\pi) = \langle q, B, q' \rangle$, with the following matrix entries: for vertices $v=v_i^q \in V(S(q))$ and $v'=v_j^{q'} \in V(S(q'))$
 - $B_{ij} = 0 \text{ iff } L_{\bar{\pi}}(v, v') = \emptyset;$
 - $B_{ij} = \mathbf{instant}$ iff $L_{\bar{\pi}}(v, v')$ is a singleton $\{w\}$, with $\tau(w) = 0$; $B_{ij} = \mathbf{slow}$ iff $L_{\bar{\pi}}(v, v') \neq \emptyset$ and $\tau(L_{\bar{\pi}}(v, v')) \subseteq [1, \infty)$;

 - $B_{ij} =$ **fast** otherwise.

It is not difficult to see that in the last case $\tau(L_{\bar{\pi}}(v,v')) \supseteq [0,1]$.

▶ Proposition 22. γ_d is a monoid morphism.

Again, this property allows a compositional computation of $\gamma_d(\pi)$.

The Obesity Pattern 4.2

▶ **Definition 23.** A cycle in an RSTA is structurally obese iff its d-orbit has the obesity pattern of one of the two types presented on Fig. 6, in the sense that either (a) $\gamma_d(\pi)$ has a **fast** diagonal element, or (b), for some indices u and v, it has two **instant** diagonal elements at (u,u) and (v,v), the element at position (u,v) is slow and some other realizable d-orbit on the same region has a non-0 element at position (v, u).

An RSTA is structurally obese if it has a structurally obese cycle.

Some comments are in order. The above definitions capture the intuition that obese behaviors are "fast" behaviors (cycles of unbounded frequency) that can be repeated during an unbounded total duration, and this can happen in two ways only:

Type I structural obesity: there exists a fast cycle that can be repeated with unbounded frequency for an unbounded accumulated duration.

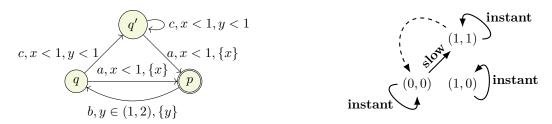


Figure 7 Type II obesity for A_8 . Left: its region-split form (transient locations omitted). Right: d-orbit for the self-loop on q' in the simplex S(q'); each tuple represents the value for clocks x and y respectively. The "resetting" cycle is $q' \to p \to q \to q'$.

Type II structural obesity: there exists a "Zeno" cycle π (with a d-orbit having a slow edge from a vertex u to another vertex v, and instant self-loops around both vertices) intersecting with another "resetting" cycle π' (with a p-orbit having an edge from v to u). The cycle π can be repeated with unbounded frequency for a total duration of ≤ 1 time unit, until π' is executed and resets this total, allowing, infinitely often, to spend another time unit in π .

Let us illustrate this definition on examples from Fig. 3. Obesity patterns for \mathcal{A}_1 and \mathcal{A}_2 can be easily seen. Consider now \mathcal{A}_8 . The d-orbit of the self-loop above its state q' is drawn in Fig. 7 (right). The type II obesity pattern occurs because, by taking the path $q' \to p \to q \to q'$, one may return from vertex (1,1) to (0,0) of the region S(q'). This orbit has an edge from (1,1) to (0,0), symbolized by the dashed line edge in Fig. 7 (right). Finally, the reader can check that \mathcal{A}_9 is not structurally obese.

Similarly to Thm. 1, structural obesity is in PSPACE.

▶ **Theorem 3.** Structural obesity of an RSTA of description size n with m clocks is decidable in SPACE($\log^2(n)$ poly(m)).

4.3 Structural Obesity ⇔ Obesity

Let us state first the upper bound for non structurally obese automata.

▶ Proposition 24. If an RsTA \mathcal{A} is not structurally obese then its bandwidth is $O(\log(1/\varepsilon))$.

The proof (see Appendix) is quite technical. At its first stage, we carefully explore Zeno cycles, which can convey a lot of information, but only for a short lapse of time. In the second stage we use Simon's theorem on factorization forests [26], and factorize words of duration T in the language, the number of factors is $\sim T$, some of them are Zeno (their total duration is only O(1)), others contain only one event or have duration 0. Finally, we make a profit from the factorization and build an ε -net for $L_T(A)$ of the required size.

The lower bound is formulated as follows.

▶ **Proposition 25.** The language of a structurally obese RsTA is obese.

The principle of the proof (see Appendix) is simple: we identify a cycle with an obesity pattern, and show that it may be used to transmit information with high frequency (every ε sec) which yields the required large bandwidth. When the clocks become too high, to continue looping through the cycle, a "resetting" path is taken.

The main result of this section is now immediate.

▶ Theorem 4. The language of an RSTA is obese iff the automaton is structurally obese.

5 Gathering Stones: Classification

5.1 Classifying DTA According to Bandwidth Asymptotics

▶ Theorem 5. A timed language accepted by a DTA is either meager, normal, or obese. Given a DTA, the three problems "is it meager?", "is it normal?", "is it obese?" are PSPACE-complete.

Proof. First, we put the DTA into RsTA form, with as many clocks but exponentially more states, as ensured by Prop. 8. If the RsTA is structurally meager (resp. structurally obese), then it is meager (obese) by Thms. 2 and 4. If it is neither structurally meager nor structurally obese, then by Props. 19 and 24 its bandwidth is both $O(\log(1/\varepsilon))$ and $\Omega(\log(1/\varepsilon))$, hence it is normal. PSPACE membership follows from Thms. 1 and 3. PSPACE-hardness can be proved (see Appendix) by reduction of reachability [1].

▶ Corollary 26. The same classification and complexity results also hold for nondeterministic timed automata without ϵ -transitions.

5.2 Comparing with Thin-Thick Classification

Previously in [3], we established a dichotomy between thin (whose language for n events is subexponentially vanishing as n goes to ∞) and thick automata (whose language has volumes growing exponentially) and proposed a structural characterization according to orbits: thick automata are those which contain a reachable *forgetful* cycle, i.e. a cycle with a p-orbit having a complete graph.

It is natural to compare the approach from [3] with the new one. First, their goals differ: the former assesses how much a time language grows at every discrete event, and the latter handles growth per unit of time. Second, in [3], only bounded guards were considered, because paths through such guards would have infinite volumes. Third, for a similar reason, paths with punctual transitions were ignored, because of their empty volumes, often entailing a thin verdict for trivial reasons. In the new approach, paths with punctual guards can contribute to meager, normal, and even obese, behaviors, while the presence of unbounded guards in a path is not a determining factor with respect to bandwidth class (efficient coding strategies tend to favor as short delays as possible). Nonetheless, in the bounded and non-punctual case, it is possible to examine the same automaton under both approaches and obtain the following result:

▶ **Theorem 6.** An RSTA with only bounded non-punctual guards cannot be both thick and meager. Also, it cannot be thin and obese of Type I. All other combinations are possible.

Proof sketch. Consider a thick automaton, with its forgetful cycle π . If its orbit has a single vertex, since π has a non-punctual transition it can be realized by several runs, hence the only edge of $\gamma_f(\pi)$ is **wide**. Otherwise, $\gamma(\pi)$ has a complete SCC of size at least 2, and then all its vertices have **wide** self-loops in $\gamma_f(\pi^2)$.

Now consider an automaton with a structurally obese cycle π of Type I. By definition, $\gamma_d(\pi)$ has a **fast** self-loop. It can be shown that such a cycle must have a complete p-orbit, hence the automaton cannot be thin.

For all the other combinations we just exhibit as examples the automata from Fig. 3:

- \blacksquare \mathcal{A}_6 is thin and meager;
- $= \mathcal{A}_{10}$ is thin and normal;

- $= A_8$ is thin and obese (type II);
- \blacksquare \mathcal{A}_4 is thick and normal;
- \mathcal{A}_1 is thick and obese (type I).
- $= A_2$ is thick and obese (type II).

6 Conclusion

We defined three classes of bandwidth and two structural criteria to classify automata into these three classes. We believe these classes will serve in the theory of timed automata since they correspond to very different kinds of behavior. We believe that techniques specific to the class of automaton would be useful for the random generation of its runs, and improve the performance and scalability of tools such as WORDGEN [5]. Our results have been formulated for timed words (time event sequences), and we think that porting them to timed signals would bring interesting insights. We believe that our results can be rephrased in terms of Kolmogorov complexity of ε -approximations of timed words.

In the future, we would like to be able to do more than classify automata: we would also like to compute the constant α that multiplies the main term in the bandwidth formulas. This step will be essential for applications to timed information coding, in the spirit of [2]. We hope that many tools introduced in this article will serve to reach this ambitious aim.

References

- 1 Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994. doi:10.1016/0304-3975(94)90010-8.
- 2 Eugene Asarin, Nicolas Basset, Marie-Pierre Béal, Aldric Degorre, and Dominique Perrin. Toward a timed theory of channel coding. In *Proc. FORMATS*, volume 7595 of *LNCS*, pages 27–42, 2012. doi:10.1007/978-3-642-33365-1_4.
- 3 Eugene Asarin, Nicolas Basset, and Aldric Degorre. Entropy of regular timed languages. *Information and Computation*, 241:142–176, 2015. doi:10.1016/j.ic.2015.03.003.
- 4 Eugene Asarin, Nicolas Basset, and Aldric Degorre. Distance on timed words and applications. In Proc. FORMATS, volume 11022 of LNCS, pages 199–214, 2018. doi: 10.1007/978-3-030-00151-3_12.
- 5 Benoît Barbot, Nicolas Basset, and Alexandre Donzé. Wordgen: a timed word generation tool. In Proc. HSCC, pages 16:1–16:7. ACM, 2023. doi:10.1145/3575870.3587116.
- **6** Johan Bengtsson and Wang Yi. *Timed Automata: Semantics, Algorithms and Tools*, pages 87–124. Springer Berlin Heidelberg, 2004. doi:10.1007/978-3-540-27755-2_3.
- 7 Jean Berstel, Dominique Perrin, and Christophe Reutenauer. Codes and Automata. Cambridge University Press, 2009.
- 8 Mikołaj Bojańczyk. Factorization forests. In $Proc.\ DLT$, volume 5583 of LNCS, pages 1–17, 2009. doi:10.1007/978-3-642-02737-6_1.
- 9 Martin R. Bridson and Robert H. Gilman. Context-free languages of sub-exponential growth. Journal of Computer and System Sciences, 64(2):308-310, 2002. doi:10.1006/jcss.2001. 1804
- Noam Chomsky and George A. Miller. Finite state languages. *Information and Control*, 1(2):91 112, 1958. doi:10.1016/S0019-9958(58)90082-2.
- David L. Dill. Timing assumptions and verification of finite-state concurrent systems. In *Proc. CAV*, volume 407 of *LNCS*, pages 197–212, 1989. doi:10.1007/3-540-52148-8_17.
- Paweł Gawrychowski, Dalia Krieger, Narad Rampersad, and Jeffrey Shallit. Finding the growth rate of a regular or context-free language in polynomial time. In *Proc. DLT*, volume 5257 of *LNCS*, pages 339–358, 2008. doi:10.1007/978-3-540-85780-8_27.

- Rostislav Grigorchuk and Antonio Machí. An example of an indexed language of intermediate growth. *Theoretical Computer Science*, 215(1):325–327, 1999. doi:10.1016/S0304-3975(98) 00161-3.
- Vineet Gupta, Thomas A. Henzinger, and Radha Jagadeesan. Robust timed automata. In *Hybrid and Real-Time Systems*, volume 1201 of *LNCS*, pages 331–345. Springer, 1997. doi:10.1007/BFb0014736.
- 15 K.A.S. Immink. EFMPlus: The coding format of the multimedia compact disc. *IEEE Transactions on Consumer Electronics*, 41(3):491–497, 1995. doi:10.1109/30.468040.
- 16 K.A.S. Immink. Codes for Mass Data Storage Systems. Shannon Foundation Publ., 2004.
- 17 Bernardo Jacobo Inclán, Aldric Degorre, and Eugene Asarin. Bounded delay timed channel coding. In *Proc. FORMATS*, volume 13465 of *LNCS*, pages 65–79, 2022. doi:10.1007/978-3-031-15839-1_4.
- 18 Andrei Kolmogorov. Three approaches to the quantitative definition of information. International Journal of Computer Mathematics, 2(1-4):157–168, 1968. doi:10.1080/00207166808803030.
- 19 Andrei Kolmogorov and Vladimir Tikhomirov. ε-entropy and ε-capacity of sets in function spaces. Uspekhi Matematicheskikh Nauk, 14(2):3-86, 1959. URL: http://mi.mathnet.ru/eng/umn7289, doi:10.1007/978-94-017-2973-4_7.
- 20 Douglas Lind and Brian Marcus. An Introduction to Symbolic Dynamics and Coding. Cambridge University Press, 1995.
- 21 Christos H. Papadimitriou. Computational Complexity. Addison-Wesley, 1994.
- Anuj Puri. Dynamical properties of timed automata. Discrete Event Dynamic Systems, 10(1-2):87-113, 2000. doi:10.1023/A:1008387132377.
- 23 John Rhodes and Pedro V. Silva. Boolean Representations of Simplicial Complexes and Matroids. Springer International Publishing, 2015. doi:10.1007/978-3-319-15114-4.
- Ocan Sankur. Robustness in timed automata: analysis, synthesis, implementation. PhD thesis, École normale supérieure de Cachan, Paris, France, 2013. URL: https://tel.archives-ouvertes.fr/tel-00910333.
- Claude E. Shannon. A mathematical theory of communication. The Bell System Technical Journal, 27(3):379–423, 1948. doi:10.1002/j.1538-7305.1948.tb01338.x.
- 26 Imre Simon. Factorization forests of finite height. Theoretical Computer Science, 72(1):65–94, 1990. doi:10.1016/0304-3975(90)90047-L.
- Andreas Weber and Helmut Seidl. On the degree of ambiguity of finite automata. *Theoretical Computer Science*, 88(2):325–349, 1991. doi:10.1016/0304-3975(91)90381-B.

A Preliminary details

A.1 Difference bound matrices

Difference bound matrices — DBM [11] are a compact way to represent and manipulate systems of difference constraints, they are extensively used in the study of timed automata. We will need only one specific property of a restricted class of DBMs.

▶ **Definition 27.** An n-DBM is a matrix $A = (a_{ij})_{i,j=0..n}$ with all $a_{ij} \in \mathbb{R} \cup \{\infty\}$. We say that a vector $x \in \mathbb{R}^n$ satisfies a DBM A whenever for all i, j = 0..n it holds that $x_j - x_i \leq a_{ij}$, where by convention $x_0 = 0$. We denote the set of all vectors satisfying a DBM A by [A] and call it a closed zone.

Timings of runs in a timed automaton can be described by DBMs, as stated here:

▶ Proposition 28. Given a timed automaton with closed guards, a path π of length n, and clock vectors x, y, the set of timings of words $L_{\pi}(x, y)$ can be described by an n-DBM $A(x, y) = (a_{ij})$ of the following parametric form:

- its entries "in the middle", i.e. a_{ij} for i, j = 1..n 1, take values in $\mathbb{Z} \cup \{\infty\}$;
- entries "on the border", that is a_{0k} , a_{k0} , a_{nk} , a_{kn} with $k \in 1..n 1$ can take values of the form $\min(x_k + b, d)$, $\min(-x_k + b, d)$, $\min(y_k + b, d)$, $\min(-y_k + b, d)$ respectively, with $b, d \in \mathbb{Z} \cup \{\infty\}$;
- finally $a_{0n} = -a_{n0} = y_k x_k$ for some k or $a_{0n} = a_{n0} = \infty$.

Proof. We translate all the conditions on timings in the run as in (3) into a DBM. We remark that the initial clock vector x is used to compute clock values, and we check that final clock values coincide with the coordinates of y. The translation can be succinctly described as follows (variables i ranges over 1..n-1 and j over 1..n):

- We initialize all the entries of the DBM at ∞ ;
- we treat all the constraints in the guards along the path as follows:
 - whenever a clock c_k , reset at t_i , is tested at t_j with the guard $b \le c_k \le d$, this can be rewritten as $b \le t_j t_i \le d$, and translated to the DBM $a_{ij} := \min(a_{ij}, d)$ and $a_{ji} := \min(a_{ji}, -b)$;
 - whenever a non-reset clock c_k (initially equal to x_k) is tested at t_j with the guard $b \le c_k \le d$, this can be rewritten as $b \le x_k + t_j \le d$, and translated to the DBM $a_{0j} := \min(a_{0j}, d x_k)$ and $a_{j0} := \min(a_{j0}, x_k b)$;
- \blacksquare and we make sure that the final values of clocks equal y as follows:
 - if a clock c_k is reset the last time at t_i , then the final value of the clock should be attained at t_n , i.e. $t_n t_i = y_k$, which translates into $a_{in} := \min(a_{in}, y_k)$ and $a_{ni} := \min(a_{ni}, -y_k)$.
 - finally, if a clock c_k is never reset along π , then $y_k = x_k + t_n$, which yields $a_{0n} = -a_{n0} = y_k x_k$.

A path in a DBM is just a sequence of indices $\sigma = i_1, i_2, \dots, i_k$ all belonging to 0..n, it is a cycle whenever $i_k = i_1$. The sum of a path is defined as follows: $sum(\sigma) = \sum_{j=1}^{k-1} a_{i_j, i_{j+1}}$.

We will use the well-known result:

▶ Proposition 29. $[A] = \emptyset$, iff there exists a cycle σ with $sum(\sigma) < 0$. For nonempty [A], its projection on x_i can be characterized as follows: $proj_i[A] = [L_i, U_i]$ where $U_i = \min\{sum(\sigma) \mid \sigma \text{ a path from 0 to } i\}$ and $L_i = \max\{-sum(\sigma) \mid \sigma \text{ a path from } i \text{ to } 0\}$. The \min and \max can also be taken w.r.t. simple paths.

We remark for knowledgeable readers that $U_i = a'_{0i}$ and $L_i = -a'_{i0}$ where $A' = (a'_{ij})$ is the canonical form of DBM A.

Later on, we will apply the following corollary on Lipshitz continuity.

- ▶ Corollary 30. Suppose that two n-DBMs A and A' with nonempty [A] and [A'] are: equal in the middle: $a_{ij} = a'_{ij}$ for all i, j = 1..n 1; ε -close on the border: $|a_{0i} a'_{0i}| < \varepsilon$ for all i = 0..n 1; similarly for a_{i0}, a_{in}, a_{ni} . and their projections on x_i are $proj_i[A] = [L_i, U_i]$ and $proj_i[A'] = [L'_i, U'_i]$. Then these projections are close to each other: $|L'_i L_i|, |U'_i U_i| < 3\varepsilon$.
- **Proof.** For any simple path σ from 0 to i, at most three weights are different in A and A', and at most by ε : one of the form $a_{0\bullet}$, and possibly one $a_{\bullet n}$ and one $a_{n\bullet}$. Thus $sum_A(\sigma)$ can differ from $sum_{A'}(\sigma)$ at most by 3ε . Since U_i and U_i' are the minima of finitely many such sums, we conclude that $|U_i' U_i| < 3\varepsilon$. The reasoning for L_i is symmetric.

A.2 Region-split form

We establish a preliminary lemma:

▶ Lemma 31. Given a TA A, it is possible to construct another TA B recognizing the same timed language, such that its starting constraint for every location is bounded for every clock.

Proof sketch. We extend the state space of the automaton \mathcal{A} to obtain the automaton $\mathcal{B} = (Q \times 2^X, X, \Sigma, \Delta', S', I', F')$ defined as follows:

- For all $q \in Q$ and $Y \subseteq X$, S'((q,Y)), I'((q,Y)) and F'((q,Y)) are obtained from, respectively, S(q), I(q) and F(q) by adding, for each clock $c \in X \setminus Y$, the constraint $x_c \leq M$ and replacing, for each clock $c \in Y$, all the constraints involving x_c by $x_c = 0$.
- For any edge $\delta \in (q, a, \mathfrak{g}, \mathfrak{r}, q') \in \Delta$, and $Y, Y' \subseteq X$, Δ' contains all edges of the following form: $(q, Y), a, \mathfrak{g} \wedge (\bigwedge_{c \in Y'} x_c > M) \wedge (\bigwedge_{c \in X \setminus (Y' \cup \mathfrak{r})} x_c \leq M), \mathfrak{r} \cup Y', (q', Y'))$, except those having a guard with a constraint of the form $x_c < a$ or $x_c \leq a$ for $a \leq M$ and $c \in Y$ (a "large" clock cannot satisfy upper bounded constraints).

Intuitively, we split every state with respect to the set Y of "large" clocks, forcing them to reset to 0 every time a location where this clock is large is entered. Y is updated as follows: when an edge from \mathcal{A} resets a clock, it becomes non-large (is removed from Y) in the destination state of all matching edges of \mathcal{B} ; when a guard checks that a clock is > M, it becomes large (is added to Y) in the destination (and thus is reset).

We obtain an automaton with the same language, but such that the starting constraint of every location is bounded. \blacktriangleleft

▶ Proposition 8. For any DTA, there exists an RSTA accepting the same language, with the same number of clocks, and an exponentially larger number of locations.

Proof. We start by constructing an equivalent automaton \mathcal{B} with bounded starting constraints, using Lem. 31. Then we use the construction from [3] to obtain an RsTA \mathcal{C} .

The construction goes as follows (simplified for the current setting where starting constraints are already given):

- 1. Split every location $q \in Q$ into sublocations q_R for each region R included in S(Q) (and we set $S(q_R) = R$).
- **2.** For every q_R , set $I(q_R) = I(q) \wedge R$ and $F(q_R) = F(q) \wedge R$.
- 3. For every edge from q to q', copy it for every version of q and q' with its guard intersected with the time successor of S(q) and the reset predecessor of S(q') (under the condition that the resulting guard is satisfiable).
- **4.** Remove all the locations and transitions not reachable from the initial states or not co-reachable from the final states.

Note that the guards, which are zones, may still contain an unbounded region, projecting to the region S(q') after the reset by \mathfrak{r} . Nonetheless, by construction, if we chose $q, x, t, q', x', \mathfrak{r}$, such that $x \in S(q)$, $x' \in S(q')$ and $\mathfrak{r}(x+t) = x'$, if R is the region of x, R' the region of x', then

- if $\delta = (q, a, \mathfrak{g}, \mathfrak{r}, q')$ is an edge of \mathcal{B} , then there is an edge of \mathcal{C} , $\delta' = (q_R, a, \mathfrak{g}', \mathfrak{r}, q'_{R'})$ (\mathfrak{g}' as defined above) such that $x + t \models \mathfrak{g}$ iff $x + t \models \mathfrak{g}'$ (i.e. one can be fired iff the other can be fired with the same timing, going from the same origin clock vector to the same destination clock vector);
- if there is an edge of \mathcal{C} of the form $\delta' = (q_R, a, \mathfrak{g}', \mathfrak{r}, q'_{R'})$, then there is an edge δ of \mathcal{B} , $\delta = (q, a, \mathfrak{g}, \mathfrak{r}, q')$ is an edge of \mathcal{B} , such that $x + t \models \mathfrak{g}$ iff $x + t \models \mathfrak{g}'$.

Using this, by induction on the length of runs, we deduce that for any run ρ starting from an initial state of \mathcal{B} there is a run ρ' in \mathcal{C} , starting from an initial state, such that both runs have the same word and one ends in an accepting state if and only if the other one does. So the languages of \mathcal{B} and \mathcal{C} are the same.

A.3 Closed reachability of region-split automata

First, we fix some definitions concerning the "closure" of various objects depending on clocks. For any constraint on \mathbb{R}_+ , its closure consists in replacing < and > by \le and \ge . The closure of a set of constraints/polytope/zone/region consists in replacing all its constraints with their closure (this definition coincides with the topological closure of the set of points satisfying the constraints). Finally we also define the closure of an edge $\delta = (q, a, \mathfrak{g}, \mathfrak{r}, q')$ as $\bar{\delta} \triangleq (q, a, \bar{\mathfrak{g}}, \mathfrak{r}, q')$ and extend this notion to any path $\pi = \delta_1 \dots \delta_n$: we denote $\bar{\pi} \triangleq \bar{\delta}_1 \dots \bar{\delta}_n$.

Next, let us remind, and adapt to RsTA, a couple of folk results about the shape of successors and predecessors, which will be useful in further developments. In the first one, we deal with regions, and in the second with their closures.

In both proofs, we use standard properties of regions and zones, extensively used in the verification of timed automata, see [6].

- ▶ Lemma 32. For an RsTA edge $\delta \in \Delta$,
- the successor of $S(src_{\delta})$ by δ is exactly $S(dst_{\delta})$;
- the predecessor of $S(dst_{\delta})$ by δ is exactly $S(src_{\delta})$.

Proof. By construction of the RsTA, the successor, and the predecessor are non-empty. A region is a zone, its successors and predecessors w.r.t. an edge δ are also zones. Also, the successor of $S(src_{\delta})$ and the predecessor of $S(dst_{\delta})$ are subsets of, resp., $S(dst(\delta))$, and $S(src(\delta))$, which are bounded regions. A non-empty zone subset of a bounded region coincides with this region (since bounded regions are minimal non-empty zones), and we can conclude.

- ▶ **Lemma 33.** For an RsTA edge $\delta \in \Delta$ and a region R,
- if $R \subseteq \overline{S(src_{\delta})}$, then its successor $\{y \mid \exists x \in \overline{R} \text{ s.t. } (x,y) \in Reach_{\overline{\delta}}\}$ is the closure of some region $R' \subseteq \overline{S(dst_{\delta})}$;
- if $R \subseteq \overline{S(dst_{\delta})}$, then its predecessor $\{x \mid \exists y \in \overline{R} \text{ s.t. } (x,y) \in Reach_{\overline{\delta}}\}$ is the closure of some region $R' \subseteq \overline{S(src_{\delta})}$.

Corollary: the same results hold by replacing the single edge δ by a path π .

Proof. Let us prove first that the successor is non-empty. We start by noticing that for $x,y\in [0,M]^X$ if $(x,y)\in Reach_\delta$, then $x\stackrel{(\delta,t)}{\to} y$ with some $t\in [0,M+1]$. Let $x\in \bar{R}$, and $x_1,x_2,\dots\in S(src_\delta)$ a sequence converging to x. By Lem. 32, for each of those x_i there exists an $y_i\in S(dst_\delta)$ and $t_i\in [0,M+1]$ such that $x_i\stackrel{(\delta,t_i)}{\to} y_i$ with all $t_i\leq M+1$. By compactness, one can choose a subsequence of indices i_k with y_{i_k} converging to some $y'\in \overline{S(dst_\delta)}$ and t_{i_k} converging to some $t'\in [0,M+1]$. Then $x\stackrel{(\bar{\delta},t')}{\to} y'$, and the successor contains at least y' and is thus non-empty. The proof for the predecessor is similar.

The successor, resp. predecessor, of \bar{R} is a non-empty closed zone with integer vertices, included in $\overline{S(dst_{\delta})}$, resp. $\overline{S(src_{\delta})}$, which are region closures. Therefore it must be a facet of this simplex, hence a region closure.

▶ Corollary 34. In the (p,f,d)-orbit $\langle p,M,q\rangle$ of any path, the matrix M has a non-0 entry in every line and every column (i.e., in the graph representation, every vertex of the destination location has an incoming edge and every vertex of the source location has an outgoing edge).

The above was already stated in [22] and remains true in the current context, thanks to Lem. 33 (considering singleton regions $\{v\}$, where v is a vertex).

Finally, the following lemma justifies that we can study the information content of the language of a closed path instead of that of its open counterpart:

- ▶ **Lemma 35.** In an RsTA, for any path π and clock vectors x, y:
- $L_{\pi^*,T}(x,y) \subseteq L_{\bar{\pi}^*,T}(x,y);$
- $\mathcal{H}_{\varepsilon}(L_{\pi^*,T}(x,y)) = \mathcal{H}_{\varepsilon}(L_{\bar{\pi}^*,T}(x,y));$
- $\mathcal{C}_{\varepsilon}(L_{\pi^*,T}(x,y)) \leq \mathcal{C}_{\varepsilon}(L_{\bar{\pi}^*,T}(x,y));$
- for any $0 < \varepsilon' < \varepsilon$, $C_{\varepsilon'}(L_{\pi^*,T}(x,y)) > C_{\varepsilon}(L_{\bar{\pi}^*,T}(x,y))$.

Proof. The first item is immediate, considering that the system of constraints defining $L_{\bar{\pi}^*,T}$ is less constraining than that of $L_{\pi^*,T}$. The third item immediately follows.

The second item is a consequence of the inclusion of the languages (for one direction) and the fact that $L_{\bar{\pi}^*,T}$ is the topological closure of $L_{\pi^*,T}$ (for the other direction) because closed ε -balls covering L_{π^*} also cover its closure.

For the last item, we fix some $t \in L_{\pi^*,T}(x,y)$, as such a run exists, and define the homothety $h_{\lambda}: \mathbb{R}^{|\pi|} \to \mathbb{R}^{|\pi|}, u \mapsto (1-\lambda)t + \lambda u$. We extend the notation h to words of $L_{\bar{\pi}^*}$: $h_{\lambda}(w) \triangleq w'$ where $letters(w') = letters(\pi)$ and $timing(w') = h_{\lambda}(timing(w))$.

Now consider S, an ε -separated set of $L_{\bar{\pi}^*,T}(x,y)$ and let us take 2 elements s_1 and s_2 of S. Without loss of generality, there is some index i such that $d(s_1,s_2)=\min_j |s_1[i]-s_2[j]| \geq \varepsilon$. Then the following inequalities hold:

$$\begin{split} d(h_{\lambda}(s_1),h_{\lambda}(s_2)) &\geq \min_{j} |h_{\lambda}(s_1)[i] - h_{\lambda}(s_2[j])| \geq \min_{j} |\lambda(s_1[i] - s_2[j]) + (1-\lambda)(t[i] - t[j])| \\ &\geq \min_{j} \left(\lambda|s_1[i] - s_2[j]| - (1-\lambda)|t[i] - t[j]|\right) \geq \lambda\varepsilon - (1-\lambda)T = \lambda(\varepsilon + T) - T. \end{split}$$

Hence, if we fix $\varepsilon' \in (0, \varepsilon)$, then the set $S' \triangleq \{h_{\frac{T+\varepsilon'}{T+\varepsilon}}(w), w \in S\}$ is ε' -separated. Furthermore $S_{\lambda} \subseteq L_{\pi^*, T}$: indeed, $L_{\pi^*, T}$ is convex and $L_{\bar{\pi}^*, T}$ is its topological closure.

This finishes proving all the intended results.

Note this result can only be used to assess the information content of a single path. Indeed, the automaton obtained by closing all the guards of an RsTA can exhibit many qualitative behaviors (paths) that were not originally present and thus a bandwidth larger than that of the RsTA.

A.4 Morphisms to finite monoids and Simon's theorem

We will need the following fundamental result

▶ **Theorem 7** (Simon [8,26]). Let Γ be a finite alphabet, \mathbf{M} a finite monoid, and $\beta : \Gamma^* \to \mathbf{M}$ a monoid morphism. Then the sequence of sets

$$X_0 = \Gamma \cup \{\epsilon\}; \qquad X_{h+1} = X_h \cdot X_h \cup \bigcup_{ee=e} (X_h \cap \beta^{-1}(e))^*$$
 (4)

stabilizes at some finite level K (in fact $K \leq 3 \cdot \#\mathbf{M}$) with $X_K = \Gamma^*$.

And now we will prove the pumping lemma.

▶ **Lemma 17** (adapted from [24, Lem. 8.7.1]). Given a finite alphabet Γ , a finite monoid \mathbf{M} , and a morphism $\beta : \Gamma^* \longrightarrow \mathbf{M}$, for any $e \in \mathbf{M}$ having $\beta^{-1}(e) \neq \emptyset$, there exists a word $w \in \beta^{-1}(e)$ of a length $\leq \#\mathbf{M}$.

Proof. In the hypotheses of the Lemma, let $w = a_1 \dots a_N$ be the shortest word in $\beta^{-1}(e)$, and consider, for any $i \in 0..N$, the prefix image $f_i = \beta(a_1 ... a_i) \in \mathbf{M}$. If $f_i = f_j$ for some i < j, then $\beta(a_0 \dots a_i a_{i+1} \dots a_N) = e$, which would contradict the minimality of w. We conclude that all f_i are different, and thus $N+1 \leq \#\mathbf{M}$.

A.5 Reachability and Lyapunov functions

We recall here Puri's characterization of reachability in terms of the orbit graph. The presentation is close to that of [3].

Given a region R and a clock vector $x \in R$, we denote $\lambda(x) = (\lambda_v(x))_{v \in V(R)}$ the barycentric coordinates of x w.r.t. the vertices of R. Hence, $x = \sum_{v \in V(R)} \lambda_v(x)v$.

▶ Lemma 36 ([22]). The reachability in an RSTA along an edge sequence π with $\gamma(\pi)$ $\langle q, A, q' \rangle$ can be characterized as follows. For any two clock vectors $x \in S(q)$ and $x' \in S(q')$, we have that $(x,x') \in Reach_{\pi}$ iff there exists a stochastic matrix $P \leq A$ (the inequality is element-wise) such that $\lambda(x)P = \lambda(x')$. Consequently, two paths have the same p-orbit iff they have the same reachability relation.

To be more precise about this citation:

- The lemma was originally stated for cyclic paths only, but the fact π is a cycle was not used in its proof.
- Also only one direction of the implication is stated (reachability \implies existence of a matrix), but the final argument can be reversed: the existence of a matrix allows writing y as a convex combination $\sum_i \lambda_i(x) s_i$ with $s_i = \sum_j P_{ij} w_{ij}$, where the w_{ij} are vertices of $S(dst(\pi))$ reachable from vertex v_i of $S(dst(\pi))$. So $y \in \sum_i \lambda_i(x)R(v_i) =$ $R(\sum_{i} \lambda_{i}(x)v_{i}) = R(x)$, where $\forall x, R(x) \triangleq \{y | L_{\bar{\pi}}(x,y) \neq \emptyset\}$. Note this reasoning (for both directions) uses the key identity $R(\lambda x + (1 - \lambda)y) =$ $\lambda R(x) + (1 - \lambda)R(y)$, which is proved in another lemma.
- Last, the result from [22] was established for a setting where all regions are bounded. Fortunately, the original proof can be interpreted in the case where the set of vectors reachable in every intermediary step is an arbitrary zone (in particular, an unbounded region).

Indeed, the identity $R(\lambda x + (1 - \lambda)y) = \lambda R(x) + (1 - \lambda)R(y)$ just comes from the fact that the time successor and reset operations are affine operations from and to convex sets. This is still the case in our context.

Nonetheless, it is still important that the starting and ending regions of the path are both bounded, for the final result to be stated in terms of vertex-to-vertex reachability and barycentric coordinates.

▶ **Definition 37** ([3]). Given a cycle π on a region R, that is with $R = S(src(\pi))$, we call $\ell: R \to [0;1]$ a Lyapunov function for π whenever the following holds: if $y \in R$ can be reached from $x \in \bar{R}$ through $\bar{\pi}$, then $\ell(x) \geq \ell(y)$.

We will slightly generalize the construction of a Lyapunov function for a cycle in timed automata from [3].

Construction 38. Let e be an orbit graph and V = V(R) its supporting set of vertices. Let $I \subseteq V$ be an initial set, i.e. such that all incoming edges of e to I have origins in I. Given $x \in \overline{R}$, let λ_i be the barycentric coordinates of x w.r.t. V. We define the function $\ell_I(x) = \sum_{i \in I} \lambda_i$.

▶ **Lemma 39.** The construction above always yields a Lyapunov function for any path π such that $\gamma(\pi) = e$.

The proof is identical to that of [3, Lemma 11].

To have many independent Lyapunov functions, we proceed as follows:

▶ Construction 40. Given an orbit graph e and its supporting set V, we partition V into SCCs, sort these SCCs in a topological order $(S_0, S_1, S_2, ..., S_s)$ and define initial sets $I_i = \bigcup_{j=0}^{i-1} S_j$, their family $\mathcal{I} = (I_i)_{i \in 1...s}$, and the corresponding family of Lyapunov functions $\Lambda = \{\ell_I \mid I \in \mathcal{I}\}$. We also write $\Lambda(x)$ for the vector $(\ell_I(x))_{I \in \mathcal{I}}$. In further developments, for such a Lyapunov function ℓ_I , we call the initial set I the support of ℓ_I .

We also remark that $V \setminus I_i$ are final sets of vertices.

▶ **Lemma 41.** Given any cycle π such that $e = \gamma(\pi)$, the construction above yields a family Λ of s = #SCC(e) - 1 independent Lyapunov functions for π .

The independence means that for any subset of Λ , values of functions inside the subset do not determine the value of a function outside it. For a set of affine functions, this is equivalent to linear independence of their slopes (and to linear independence of the set $\Lambda \cup \{x \in \bar{R} \mapsto 1\}$).

Proof. Choose any $i \in 1...s$ and consider the affine function $v:[0;1] \to R$ defined as

$$v(\lambda) = \frac{1-\lambda}{\#S_i} \cdot \sum_{j \in S_i} v_j + \frac{\lambda}{\#S_{i+1}} \cdot \sum_{j \in S_{i+1}} v_j.$$

The segment v([0;1]) is exactly the one that links the barycenter of S_i and that of S_{i+1} . For any j > i, this segment is entirely included in $ConvHull(I_j)$, hence $\ell_{I_j}(v(\cdot))$ is the constant 1; for j < i, the segment is included in $ConvHull(V(r) \setminus I_j)$, so $\ell_{I_j}(v(\cdot))$ is the constant 0. But the image of $\ell_{I_i}(v(\cdot))$ is the full interval (0;1) (indeed $v(0) \in I_i$, so $\ell(v(0)) = 1$ and v(1) has null barycentric coordinates on I_i , so $\ell(v(1)) = 0$). Hence ℓ_i cannot be obtained by any linear combination of the other Lyapunov functions (i.e. $(\ell_j)_{i \neq i}$).

▶ Corollary 42. Any set of clock vectors mutually reachable through the cycle $\bar{\pi}$ on a region R can be contained in an affine set of dimension $\dim R - \#SCC(\gamma(\pi)) + 1$.

Proof. If x and y are mutually reachable, then they should satisfy $\ell(x) = \ell(y)$ for any Lyapunov function ℓ . Thus any set of mutually reachable clock vectors is contained in an affine set of the form $\{x \mid \forall \ell \in \Lambda \ \ell(x) = c_{\ell}\}$, where Λ is a set of $\#SCC(\gamma_{\pi}) - 1$ independent affine functions, hence the dimension.

▶ Lemma 43. In the case when $\#SCC(\gamma(\pi)) = \dim R + 1$ (i.e. all SCCs are singletons), $||x-y||_{\infty} \leq K||\Lambda(x)-\Lambda(y)||_{\infty}$ (i.e. Λ is a dilatation), for some positive real number K depending only on $\gamma(\pi)$. In this case, Λ is a bijection between \bar{R} and the simplex $0 \leq \ell_1 \leq \cdots \leq \ell_{\dim R} \leq 1$.

Proof. We use the notations of Constr. 40.

For any clock vector x, $\Lambda(x)$ is an affine function of the form $x \mapsto LPB(x-v_s)$ where:

- v_s is the unique vertex of S_s .
- B the square matrix such that $B(x v_s)$ is the vector of barycentric coordinates of x, excluding the coordinate in v_s (clocks and vertices both in canonical order: i.e. clocks sorted by fractional part value and vertices sorted lexicographically).

- P is the permutation matrix reordering rows such that output vector coordinates correspond to the topological order from Constr. 40 instead of the lexicographical order.
- L is the matrix associating a vector of barycentric coordinates to its Lyapunov values, one per line. Hence L is such that its ith row has a 1 for each position of index v such that $v \in I_i$ and 0 everywhere else. Because vertices are in topological order and each SCC has exactly one more vertex than the previous one, L actually is the lower triangular matrix having only 1's in the diagonal and below.

Since all three matrices L, P and B are invertible, we can write $(x-v_s)=(LPB)^{-1}\Lambda(x)$, hence $(x-y)=(LPB)^{-1}(\Lambda(x)-\Lambda(y))$ and thus $||x-y||_{\infty}\leq ||B^{-1}||_{\infty}\cdot||P^{-1}||_{\infty}\cdot||L^{-1}||_{\infty}\cdot||\Lambda(x)-\Lambda(y)||_{\infty}$.

B Proofs on obesity

For technical reasons (in particular, since Lem. 61 depends on Lem. 47), we proceed first with proofs of results from Sect. 4, and next from Sect. 3.

In this section and the next one, for $\bullet \in \{d, f\}$, we use the notation $\gamma_{\bullet}(\pi)[u, v]$ to denote the label (in Λ_{\bullet}) associated to the edge from vertex u to vertex v in the \bullet -orbit of a path π (if there is no such edge, then $\gamma_{\bullet}(\pi)[u, v] = 0$).

B.1 Monoid properties

▶ Proposition 22. γ_d is a monoid morphism.

Proof. We only need to prove, for any two edge sequences π_1 , π_2 , that $\gamma_d(\pi_1 \circ \pi_2) = \gamma_d(\pi_1)\gamma_d(\pi_2)$. Notice that this is trivially true if any of the two sequences is ϵ , or if $\pi_1 \circ \pi_2$ is not a path. So assume both π_1 and π_2 are paths of length ≥ 1 such that $\pi_1 \circ \pi_2$ also is a path and let us call R the destination region of π_1 .

Let us choose any source vertex u and any destination vertex v for $\pi_1 \circ \pi_2$. Then $(\gamma_d(\pi_1)\gamma_d(\pi_2))[u,v]$ has to take one of the four values of Λ_d :

- If it is **instant**, it means there is at least one vertex w of R such that $\gamma_d(\pi_1)[u,w] =$ **instant** and $\gamma_d(\pi_2)[w,v] =$ **instant**. By definition, for a run from u to v through w, the prefix to w has duration 0 and also the suffix from w, hence the whole run lasts 0. Moreover all vertices w of R are either like this or such that either $\gamma_d(\pi_1)[u,w] = 0$ or $\gamma_d(\pi_2)[w,v] = 0$, in which case there is no run from u to v through w. Hence all runs from u to v last 0 second and thus $\gamma_d(\pi_1 \circ \pi_2)[u,v] =$ **instant** $= (\gamma_d(\pi_1)\gamma_d(\pi_2))[u,v]$.
- If it is **slow**, it means there is at least one vertex w of R such that $\{\mathbf{slow}\}\subseteq \{\gamma_d(\pi_1)[u,w], \gamma_d(\pi_2)[w,v]\}\subseteq \{\mathbf{instant}, \mathbf{fast}, \mathbf{slow}\}$. In this case, there is a run through w such that either its prefix or suffix lasts more than 1, hence the full run lasts more than 1. Moreover all vertices w of R are either like this or such that either $\gamma_d(\pi)[u,w]=0$ or $\gamma_d(e)[w,v]=0$, in which case there is no run from u to v through w. Hence all runs from u to v last more than 1 second and thus $\gamma_d(\pi_1\circ\pi_2)[u,v]=\mathbf{slow}=(\gamma_d(\pi_1)\gamma_d(\pi_2))[u,v]$.
- If it is **fast**, it means either there is at least one vertex w of R such that $\{\mathbf{fast}\}\subseteq \{\gamma_d(\pi_1)[u,w],\gamma_d(\pi_2)[w,v]\}\subseteq \{\mathbf{instant},\mathbf{fast}\}$ or there exists w_1 and w_2 such that $\gamma_d(\pi_1)[u,w_1]=\mathbf{instant}$ and $\gamma_d(\pi_2)[w_1,v]=\mathbf{instant}$ and $\{\mathbf{slow}\}\subseteq \{\gamma_d(\pi_1)[u,w_2],\gamma_d(\pi_2)[w_2,v]\}\subseteq \{\mathbf{instant},\mathbf{fast},\mathbf{slow}\}$. In the first case, there are runs through w with durations 0 and 1. In the second case, there is a run through w_1 with duration 0 and a run through w_2 with duration ≥ 1 . Hence in all cases there are runs of length 0 and 1 (and therefore all durations in between), thus $\gamma_d(\pi_1\circ\pi_2)[u,v]=\mathbf{fast}=(\gamma_d(\pi_1)\gamma_d(\pi_2))[u,v]$.

- If it is 0, it means that $\{w|\gamma_d(\pi_1)[u,w]\neq 0\}\cap \{w|\gamma_d(\pi_2)[w,v]\neq 0\}=\emptyset$. By Lem. 36, it implies there is no clock vector in R such that it can be reached from u through π_1 and that v can be reached from it through π_2 , in other words there is no run from u to v through $\pi_1 \circ \pi_2$. Hence, $(\gamma_d(\pi_1)\gamma_d(\pi_2))[u,w]=0=(\gamma_d(\pi_1)\gamma_d(\pi_2))[u,v]$
- ▶ Lemma 44 (Orbits of zero-time transitions). Let $\delta \in \Delta$ be an RSTA edge that can be traversed in 0 time units from the vertex v_0 having the lowest coordinates in its source region. Then for any source vertex u there is a destination vertex v such that $\gamma_d(\delta)[u,v] \in \{\text{instant}, \text{fast}\}.$

Proof. Let us just observe that $v_0 \models \mathfrak{g}_{\delta}$. For any source vertex u, there is a delay $d \geq 0$ such that $u + d \models \mathfrak{g}_{\delta}$ (the automaton is region-split), so \mathfrak{g}_{δ} is satisfied by both v_0 and u + d. Due to the shape of the guards, it means it is also satisfied by all clock vectors that are greater or equal to that of v_0 and smaller or equal to that of u + d. This rectangle contains the coordinates of u, hence $u \models \mathfrak{g}_{\delta}$.

So δ can be traversed in zero time from u. By Lem. 36, there is at least one such run going to some vertex v of the target region, such that $\gamma_d(\delta)[u,v] \in \{\text{instant}, \text{fast}\}.$

- ▶ **Lemma 45** (Self-loops on a cycle). For a cyclic path π , if its d-orbit contains a vertex u with a zero-timed (instant or fast) self-loop, then
- if it contains a vertex v, different from u, with a slow self-loop, then the d-orbit also contains a slow edge from u to v and a zero-timed (instant or fast) edge from v to u;
- it cannot contain a vertex v, different from u, having a **fast** self-loop.

Proof. We assume $v \neq u$; $\gamma_d(\pi)[u, u] \in \{\text{instant}, \text{fast}\}\$ and $\gamma_d(\pi)[v, v] \in \{\text{fast}, \text{slow}\}.$

Let $x = (x_1, \ldots, x_c)$ and $y = (y_1, \ldots, y_c)$ be the clock vectors for u and v respectively. As there is a positive-timed self-loop on v, it is possible to take the cycle π in a non-immediate manner (with duration d > 0). For this to be possible, every clock has to be reset during π (otherwise there is some i such that $y_i = y_i + d$, which is impossible). Since all clocks are reset during the cycle, doing it with duration 0 necessarily leads to the vertex of clock coordinates $(0, \ldots, 0)$, hence $x = (0, \ldots, 0)$. Moreover, the existence of this zero-time self-loop implies the possibility to traverse each transition without waiting. This means, starting from $(0, \ldots, 0)$, each automaton edge can be traversed without waiting and thus staying in $(0, \ldots, 0)$. By Lem. 44, this implies that for each automaton edge in π , for all source vertices, there is an instant or fast outgoing edge. So, if we concatenate the orbits, all vertices of the d-orbit of π have a zero-time outgoing edge. Because the cycle resets all the clocks, these edges all necessarily go to u. This also implies that there cannot be a zero-time realization of the self-loop around v, hence this self-loop was actually slow.

For the edge from u to v, consider the cycle started at u, with the same delays as the ones of the concrete path realizing the self-loop of v (it is possible to take such a cycle, as all clock values are smaller in u than in v). The clock vector after such a run of the cycle is the same as v's (every clock has been reset, so its value in v is the sum of delays after its last reset). So v can be reached from u and there is an edge from u to v. This edge cannot be fast or instant because of the uniqueness of the zero-timed run from u.

▶ Lemma 46. In any idempotent d-orbit, instant loops only exist in singleton SCCs.

Proof. Let o be an idempotent d-orbit. Assuming o has a vertex u with an **instant** loop e in an SCC contains several vertices. Thus u must have another outgoing edge, which must be **slow**. But one can go back from the target of this edge, hence there is a **slow** circuit c of length l going through u, but $o = o^l$ contains both c and e^l , hence the label of the loop through u should have been $\mathbf{slow} + \mathbf{instant}^l = \mathbf{fast}$ and not $\mathbf{instant}$.

Finally, let us state an intermediary result, an immediate corollary of Lem. 45, that we use in several proofs in this subsection.

▶ Lemma 47. A cycle having an idempotent p-orbit and a d-orbit with a zero-timed self-loop and a positive-time self-loop actually has a complete p-orbit. If the d-orbit is also idempotent, then the cycle is structurally obese.

Proof. In an idempotent p-orbit of a region cycle, all initial and terminal vertices have self-loops, so in the d-orbit of the same cycle, all these vertices have loops. But the condition implies, according to Lem. 45, that all vertices with self-loops are in the same SCC. Hence the d-orbit is strongly connected, thus the p-orbit graph is complete. Since the vertices with zero-time loops also belong to positive-time cycles, if the d-orbit is idempotent, it means the self-loop was both zero-time and positive-time, hence **fast**, which is a realization of the obesity pattern.

B.2 Upper bound

Given a region-split automaton that is not structurally obese – or NSOA for short – we want to show that its bandwidth is $O(\log(1/\varepsilon))$ (and hence it is not obese).

At the **first stage** we explore cycles in the automaton corresponding to idempotents of \mathbf{M}_d , with a particular interest to Zeno behaviors (defined below). Let us split idempotent d-orbits of any automaton into four disjoint sets:

 E_0 (null) its elements have no realization;

 E_I (instant) its elements are realizable and all their realizations have duration 0;

 E_F (fast) its elements have at least one realization of duration in (0;1);

 E_S (slow) its elements are realizable and any realization has duration ≥ 1 .

We expect and will show in the following, that elements of E_I should not be able to create much bandwidth, because the distance collapses simultaneous events together; that those of E_S should not be able to do so either, because they are not fast enough; and, for obvious reasons, non-realizable orbits in E_0 do not contribute at all. The case of E_F is more involved. Intuitively, its elements correspond to fast behaviors which, in the case of non-structurally obese automata, should not be able to run for an unbounded duration, as they would otherwise be a source of obese bandwidth. More specifically, we will prove that elements of E_S satisfy the Zeno property:

▶ **Definition 48** (Zeno property). A d-orbit e is Zeno when, for any run factorized as $\zeta_0 \rho_1 \eta_1 \zeta_1 \rho_2 \eta_2 \dots \rho_n \eta_n \zeta_n$ such that $\gamma_d(\rho_i) = \gamma_d(\eta_i) = e$ for all i, we have that $\sum_i \tau(\rho_i) \leq 1$.

I.e. the accumulated duration of all occurrences of cycles with orbit e, except the last one in each pack, is bounded by 1. The last iteration needs to be excluded for purely technical reasons.

To establish that elements of E_f are Zeno, we formulate a sufficient condition (in terms of the monoid) — structural Zenoness, show that in a NSOA, elements of E_F satisfy this property (Lem. 50), and finally that structural Zenoness implies Zenoness (Lem. 53).

- ▶ **Definition 49.** A d-orbit is structurally Zeno iff all its self-loops are **instant** and it has at least one **slow** edge.
- ▶ **Lemma 50.** For an NSOA, all elements e in the set E_F are structurally Zeno.

Proof. First, observe that any edge with a **fast** self-loop would already be an instance of the obesity pattern (with u = v). Moreover, according to Lem. 47, if there were both an **instant** and a **slow** self-loop in an idempotent d-orbit, the orbit would also contain the obesity pattern.

Let us take any element $e \in E_F$.

The d-orbit e, being in E_F , must have both a zero time edge and a positive time edge. If they are the same edge, it is a **fast** edge, which cannot be a self-loop (as we just observed), so there must be 2 distinct vertices. If the zero time and positive time edges are different edges, they must either have different sources or different destinations (otherwise they are the same, **fast**, edge), hence there are at least 2 vertices in all cases. Either both these vertices have a self-loop, or at least one does not (we call such a vertex transient), but in this case, it is reachable and co-reachable from 2 different vertices having self-loops. In all cases, there are 2 vertices with self-loops, which cannot be **fast**.

But it is also impossible for all self-loops to be **slow**: since e is idempotent and contains a zero-time (**instant** or **fast**) edge, then it must contain an arbitrarily long (non-elementary) zero-time path of edges, which may only have zero-time factors. However, sufficiently long paths must contain an (also zero-time) cycle, the starting vertex of which must have a zero-time self-loop (because of idempotency), which contradicts the hypothesis.

Thus necessarily, all loops in e are **instant**. Just notice in this case that all SCC of e are singletons (Lem. 46). Additionally, we also know that non-self-loop outgoing edges of e from vertices having a self-loop must be **slow**.

Before proving that structural Zenoness implies Zenoness, let us describe the precise shape of structurally Zeno orbits.

▶ Lemma 51. Let e be a structurally Zeno d-orbit. Then, for all clocks c there is a natural number low_c such that, in all regions traversed by any realization π , c has values staying in an interval $[low_c, high_c)$ (where $high_c = low_c + 1$).

For clocks c reset in some realization of e low_c = 0 and high_c = 1.

Proof. Any clock c never reset in any realization π must have increasing values along any run of π^* . So it is not possible for π to visit regions with different intervals for c (since different intervals are disjoint): otherwise, let I and J be two different visited intervals; wlog we assume all elements of I are smaller than all elements of I; after visiting I in one iteration of I, I would be visited in the next iteration, with I0 getting a strictly smaller value than one obtained before when visiting I1. Since the common interval is defined by region constraints, the interval must be a singleton or an open interval of length 1 and natural bounds. Singleton is impossible (this would imply all runs have duration 0, so there could not be any slow edge), so it is open of length 1 (indeed, at least the starting region is bounded, and all clocks in this region have a value either contained in a singleton or in an interval of length 1).

Since all paths realizing e share at least a common first region, the interval is common to all regions traversed by all realizations.

So the statement is true for non-reset clocks.

Observe that there has to be at least one non-reset clock in any π realizing e: indeed a zero-time run along $\bar{\pi}$ is possible from any non-transient vertex (all loops are **instant**); but with all clocks being reset, such runs all go to $(0, \ldots, 0)$, implying there is only one self-loop, which is only possible if there is only one vertex, which contradicts there must also be a **slow** edge.

So, there is one non-reset clock, and its value remains contained in an open interval of length 1, common to all regions traversed by any realization of e. This implies the total

duration of any run along any factor of an element of $Paths(e)^*$ is strictly smaller than 1, this is true in particular if the run starts from a region where a clock was just reset. So clocks reset during π can have values in [0,1) only.

- ▶ Lemma 52 (Shape of structurally Zeno d-orbits). Let e be a structurally Zeno d-orbit. We define:
- \blacksquare X_{nr} as the set of clocks not reset by any realization of e;
- \blacksquare and X_r as those that are reset by some realization;
- v_0 , the vertex of e such that for any clock, $(v_0)_c = low_c$ (as defined in Lem. 51);
- V_h , the set of vertices v such that for any clock c in X_{nr} , $v_c = high_c$;
- v_1 , the vertex of V_h such that for any clock in X_r , $(v_1)_c = low_c$.

Then:

- $v_0 \in V(e) \ and \ v_1 \in V(e);$
- the vertices v such that, for all clock $c \in X_r$ $v_c = 0$, have an instant self-loop and are the only ones with a self-loop;
- all other vertices v are transient, belong to V_h and only have one outgoing edge; this edge is **instant** and goes to v_1 ;
- from v_0 to any vertex $v \in V_h$ there is a **slow** edge and all the **slow** edges are like this. Furthermore, if the automaton is not structurally obese, then there is no edge from v_1 to v_0 in any realizable p-orbit.

Proof. Let us define $e' \triangleq \gamma_{np}(e)$ (the p-orbit common to all paths of Paths(e), it is idempotent).

Observe that clocks that are positive for at least one vertex with a self-loop (necessarily **instant**) must be in X_{nr} (otherwise the outgoing zero-time edge would have to target a vertex having 0 as coordinate for this clock, hence this edge would not be a self-loop).

Also, edges in e' can only go from a vertex to another one with equal coordinates on X_{nr} (for an **instant** edge in e) or one with all coordinates on X_{nr} strictly higher (for a **slow** edge in e), this means that for a vertex to have an outgoing **slow** edge in e, it needs to have all coordinates in X_{nr} set to low (let us call the set of such vertices V_l), otherwise a delay of 1 time unit would go outside R, the support region of e. Also, this edge must go to a vertex where all clocks in X_{nr} are high (i.e. a vertex of V_h). Remark that at least one **slow** edge actually exists from V_l to V_h , otherwise e would not be structurally Zeno. Moreover, due to this structure, V_h contains a terminal SCC of e', hence at least one non-transient vertex (idempotency), but such a vertex can only be the vertex v_1 having all clocks in X_r set to 0. Since this is the only vertex in V_h that is terminal for e', it is reachable by all vertices in V_h . By idempotency, it means there is a **slow** edge from V_l to v_1 . Dually, V_l contains an initial SCC of e', hence a non-transient vertex, which is the unique vertex v_0 such that all clocks in X_r have value 0. v_0 has an outgoing **slow** edge into V_h , and repeating the argument of idem-potency, it has a **slow** edge to v_1 .

Moreover, since v_1 has all clocks in X_r equal to 0 and those in X_{nr} equal to high, inside the region, clocks in X_r must have a smaller fractional part than those in X_{nr} . Therefore the component of edges having coordinates in X_{nr} set to low, must also have the resetting ones set to low = 0, hence V_l actually is the singleton $\{v_0\}$.

To finish, concerning non-transient vertices, vertices neither in V_l nor V_h are only connected to other vertices having the same coordinates in X_{nr} . Hence, each class of common coordinates contains at least one initial and one terminal SCC, which have to be non-transient singletons, for which all clocks in X_r are set to 0. Therefore each such class actually is a non-transient, isolated, singleton (so all transient vertices are in V_h).

Concerning transient vertices, let us consider any given region path π realizing e. For any $i=0..|\pi|$, every vertex v of $\gamma_d(\pi[i])$ such that, for at least one $c\in X_{nr}$, $v_c=high_c$, has exactly one successor: its successor by reset. Notice this is actually an **instant** edge. This successor, hence having the same values as v for every clock of X_{nr} . So, by induction on i, we obtain that, in $e=\gamma_d(\pi)$, any vertex of V_h has exactly one outgoing edge, going to another vertex with the same coordinates on X_{nr} and this edge is **instant**. This is, in particular, true for transient vertices, as we already know they belong to V_h . But as they need to have v_1 as a successor (transitivity), v_1 actually is their only successor and it is reachable by an **instant** edge.

Then, we have to prove that there is a **slow** edge from v_0 to every vertex in V_h other than v_1 . Consider v such a vertex. Being transient, it should have an incoming edge from elsewhere (by Cor. 34). This can either be an **instant** edge from another vertex of V_h (which should have its own incoming edge) or a **slow** edge coming from v_0 (as the only element in V_l). As elements in V_h cannot have self-loops, and as the number of vertices is finite (if a vertex from V_h is repeated, by idempotency this will mean a self-loop), eventually the second option will have to be taken, and by idempotency, this means a **slow** edge from v_0 to v.

Lastly, concerning realizable p-orbits, just remark that going from v_1 to v_0 would create an obesity pattern of type II (with v_0 in the role of u and v_1 in the role of v).

Now we established a sufficiently precise characterization of structurally Zeno orbits and thus, thanks to Lem. 36, that of the stochastic matrices that may transform any clock vector, by executing any realization of a given structurally Zeno orbit. We use this knowledge to prove the next lemma:

▶ Lemma 53. In an NSOA, any structurally Zeno d-orbit is Zeno.

Proof. Let us fix a structurally Zeno idempotent e.

To prove the lemma, first, we describe a ranking function η from the support region of e to the real interval [0,1] such that its value cannot decrease along a run; then we show that in any sequence of k+1 repetitions of e (for some positive integer k), the accumulated duration of the first k consecutive realizations of e is smaller or equal to the increase of η over the k+1 realizations. So, on a full run, the total duration spent in all realizations of e (except the last of each repetitive sequence) is a fortiriori smaller than the total increase of η over the run, which is smaller than 1.

In the developments below, we abundantly use the definitions and characterizations from Lem. 52.

Construction of the ranking function η . We partition the vertices of the region into the sets $V_1 \triangleq \{v \mid \exists \pi \text{ s.t. } (v_1, v) \text{ is an edge of } \gamma(\pi)\}$, and $V_0 \triangleq V(R) \setminus V_1$. Since there is no return path from v_1 to v_0 (no structural obesity), this means $v_0 \in V_0$. Note that this partitioning only depends on R, and not on e, knowing R.

We consider the Lyapunov function ℓ_{V_0} , as described in Constr. 38. As V_0 is an initial set of vertices of $\gamma(\pi)$ for any cycle π starting from R (its complement, V_1 , is defined as a reachability cone), for any clock vectors x and x' in R, such that $x' \in Reach_{\pi}(x)$, it holds that $\ell_{V_0}(x) \geq \ell_{V_0}(x')$ (still according to the proof of the same corollary).

Hence, defining η as the function $x\mapsto 1-\ell_{V_0}$, we obtain a non-decreasing ranking function as required.

Proof of the link of η with the accumulated duration of e. For x_0, x_1, x_2 , clock vectors and ρ_1 and ρ_2 , realizations of the structurally Zeno d-orbit e such that x_1 is reached from x_0 by

reading ρ_1 and x_2 is reached from x_1 by reading ρ_2 , we will prove that $\tau(\rho_1) \leq \eta(x_2) - \eta(x_0)$ (which is enough to deduce the Zeno property of e).

According to Lem. 36, there exist two stochastic matrices P_1 and P_2 such that $\lambda(x_1) = \lambda(x_0)P_1$, $\lambda(x_2) = \lambda(x_1)P_2$, where for any clock vector x in the region R, $\lambda(x)$ is the vector of its barycentric coordinates with respect to vertices of R, sorted in clock order. Furthermore, according to the characterization from Lem. 52, P_i must have the following form, for $\mu_j^i \geq 0$ and $\sum_j \mu_j^i = 1$:

$$P_i = \begin{pmatrix} \mu_0^i & 0 & 0 & \dots & 0 & 0 & \mu_1^i & \mu_2^i & \dots & \mu_{\#X_r}^i \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & \ddots & 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 \end{pmatrix} = \begin{pmatrix} \mu_0^i & 0 & \mu_1^i & \mathbf{M}_i \\ 0 & Id_{\#X_{nr}-1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1_{\#X_r-1} & 0 \end{pmatrix},$$

where $Id_{\#X_{nr}-1}$ is the identity matrix of size $\#X_{nr}-1$, \mathbf{M}_i is the line vector $(\mu_2^i, \dots, \mu_{\#X_r}^i)$

and
$$1_{\#X_r-1}$$
 is the column vector $\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$ of height $\#X_r-1$.

Since the matrix P_i describes a barycentric coordinates transform where vertices are sorted in clock order, the first line and column concern v_0 , the next $\#X_{nr} - 1$ (5 here) lines and columns concern the isolated vertices with a self-loop, the $\#X_{nr} + 1$ -th line and column concerns v_1 , and last, the last $\#X_r - 1$ (3 here) lines and columns concern the transient vertices.

Remark that running e in t units of time must increase all non-reset clocks by t, in particular c the one with the smallest fractional part. Hence going from x_0 to x_1 , if we decompose x_i in the form $\lambda_i v_0 + (1 - \lambda_i) v$, with $v \in ConvHull(V(R) \setminus \{v_0\})$ and λ_i stands for $\lambda(x_i)_{v_0}$, then projecting on coordinate c, we get $\tau(\rho_1) = x_c^1 - x_c^0 = (1 - \lambda_1) - (1 - \lambda_0) = \lambda_0 - \lambda_1$. Using the notations from the matrix P_1 , we get $\tau(\rho_1) = \lambda_0 - \lambda_0 \mu_0^1$.

Now, observe that $\eta(x_2) - \eta(x_0) = \lambda(x_2) \mathbb{1}_{V_1} - \lambda(x_0) \mathbb{1}_{V_1} = \lambda(x_0) (P_1 P_2 - Id) \mathbb{1}_{V_1}$ (where $\mathbb{1}_{V_1}$ is the column vector having 1 for all entries corresponding to the index of a vertex of V_1 and having 0 everywhere else). Moreover

and thus

$$(P_1 P_2 - Id) \mathbb{1}_{V_1} = \begin{pmatrix} 1 - \mu_0^1 + \mu_0^1 \sum_{i \in V_1} \mu_i^2 \\ 0 \\ 0 \\ 1 \text{ for } V_0, 0 \text{ for } V_1 \end{pmatrix},$$

where
$$\kappa = \mu_0^1 \mu_1^2 + \sum_{i=1}^{\# X_r} \mu_i^1 = \mu_0^1 \mu_1^2 + 1 - \mu_0^1$$
. Hence

$$\eta(x_2) - \eta(x_0) = \lambda(x_0) (P_1 P_2 - Id) \mathbbm{1}_{V_1} = \lambda_0 (1 - \mu_0^1 + \sum_{i \in V_1} \mu_0^1 \mu_i^2) + \sum_{i \not \in V_1} \lambda_i(x) \geq \lambda_0 (1 - \mu_0^1) = \tau(\rho_1).$$

Now we proceed to the **second stage** of the upper bound proof. Using Simon's theorem, we factorize words in the language into three types of subwords, where type F factors correspond to Zeno subwords.

- ▶ Lemma 54. For any NSOA \mathcal{A} , there exist constants a,d, such that any timed word $w \in L(A)$ admits a factorization $w = u_1u_2 \cdots u_N$ and a labeling $\ell_1\ell_2 \dots \ell_N \in \{B,I,F\}^*$ with $N \leq a\tau(w) + a$ satisfying
- whenever $\ell_i = B$, the corresponding word u_i has a form (t, a) with $t \geq 0, a \in \Sigma$;
- whenever $\ell_i = I$, the word is instant: $\tau(u_i) = 0$;
- time spent with label F is bounded: $\sum_{i \in \ell_i = F} \tau(u_i) \leq d$.

Proof. Consider the set P of all paths in \mathcal{A} , we have that $P \subset \gamma^{-1}(\mathbf{M}_d \setminus \{0\})$. To alleviate notation, given $e \in \mathbf{M}_d$ we write P_e for $\gamma^{-1}(e)$.

Thus, by Simon's theorem, $P \subset P_K$ for some level K of the following recurrence:

$$P_0 = P_B = \Delta_A \cup \{\epsilon\}; P_{h+1} = P_h \cdot P_h \cup \bigcup_{e \in E_I} (P_e \cap P_h)^* \cup \bigcup_{e \in E_F} (P_e \cap P_h)^* \cup \bigcup_{e \in E_S} (P_e \cap P_h)^*.$$
 (5)

We will slightly modify this inductive construction: consider the recurrence

$$P_0' = P_B \cup P_I \text{ (with } P_I = \bigcup_{e \in E_I} P_e \text{)}; \quad P_{h+1}' = P_h' \cdot P_h' \cup \bigcup_{e \in E_F} P_e \cdot (P_e \cap P_h') \cup \bigcup_{e \in E_S} (P_e \cap P_h')^*.$$

By induction at each level $P_h \subset P_h'$, and thus $P \subset P_K'$. In the union over E_F we overapproximate all the iterations except the last one by P_e and use the fact that $P_e^+ = P_e$ for idempotent e.

The previous recurrence concerned paths in \mathcal{A} , we need its version for time-bounded runs. For any T, let R_T be the set of all runs of \mathcal{A} with duration $\leq T$. We also define run counterparts of all sets of paths defined above, so $R'_{hT} = Runs(P'_h) \cap R_T$ etc.

First, we consider short runs (of duration <1). Applying $Runs(\cdot) \cap R_{<1}$ to (5) we obtain

$$R'_{0,<1} = R_{B,<1} \cup R_I; \qquad R'_{h+1,<1} = ((R'_{h,<1} \circ R'_{h,<1}) \cap R_{<1}) \cup \bigcup_{e \in E_F} R_e \circ (R_e \cap R'_{h,<1})$$

(we have omitted the last term of (5) since it does not produce short runs). The level K of the induction generates all the short runs: $R_{<1} \subset R'_{K,<1}$.

For any short run $\rho \in R_{<1}$, by induction on h we produce a factorization and labeling required in the statement of the lemma as follows:

- for the base case $\rho \in R'_{0,<1}$ we take only one factor, and label it I if $\rho \in R_I$ or B otherwise;
- for the inductive case $\rho \in R'_{h+1,<1}$, whenever $\rho = \rho_1 \circ \rho_2$ (with $\rho_i \in R'_{h,<1}$) we factorize ρ_i by inductive hypothesis and merge the factorizations and labelings;
- finally, for the inductive case of $\rho = \rho_1 \circ \rho_2$ with $\rho_1 \in R_e$ and $\rho_2 \in R_e \cap R'_{h,<1}$ with $e \in E_F$, we label the whole ρ_1 by F. We factorize and label ρ_2 using the inductive hypothesis.

32

We obtain at level h at most 2^h factors, hence at most 2^K factors for any short run $\rho \in R_{<1}$. The labeling of the short runs has the following Zeno-like property: in any run $\zeta_0 \rho_1 \zeta_1 \rho_2 \zeta_2 \dots \rho_n \zeta_n$ with all $R'_{h,<1}$ short, let us label factors of ρ_i as described above. Then the accumulated duration of F-labeled ones is bounded by $h \# E_F$. This property can be proved by induction on h, and implies the global bound of $d = K \cdot \# E_F$.

For each e, on level h+1, the accumulated duration of new F-factors is ≤ 1 due to Zeno property (Lemmas 50 and 53); summing over all possible e this gives $\#E_F$. The accumulated duration of old factors (inherited from level h) is at most $h\#E_F$. This implies the announced bound.

Consider now the set of runs R_T for an arbitrary T. Applying $Runs(\cdot) \cap R_T$ to the recurrence (5) we obtain a recurrence for runs:

$$R'_{0T} = R_{BT} \cup R_I; \qquad R'_{h+1,T} = \left(R'_{hT} \circ R'_{hT} \cup \bigcup_{e \in E_F} (R_e \cap R'_{hT})^* \cup \bigcup_{e \in E_S} (R_e \cap R'_{hT})^* \right) \cap R_T,$$

and $R_T \subset R'_{KT}$. We overapproximate it as follows

$$R_{0T}'' = R_{BT} \cup R_{<1}; \qquad R_{h+1,T}'' = R_{hT}^{" \le T+2} \cap R_T.$$

We omitted the union over E_F because it is included in $R_{<1}$ already injected at level 0. The star in the union over E_S has been truncated up to power T since each run R_e with $e \in E_S$ has a duration of at least 1.

Again we have $R_T \subset R''_{KT}$. Now we can produce the factorization and labeling of each run in R_T :

- for the base case $\rho \in R''_{0,<1}$ such that $\rho \in R_{BT}$ we take one factor, and label it B;
- otherwise, in the base case such that $\rho \in R_{<1}$ we apply the factorization for short runs described above;
- for the inductive case $\rho \in (R''_{hT})^{\leq T} \cap R_T$, in this case $\rho = \rho_1 \rho_2 \dots \rho_n$ with $n \leq T + 2$ with $\rho_i \in R''_{hT}$ satisfying the inductive hypothesis. We factorize and label each of ρ_i and obtain a factorization of ρ .

We will now prove by induction over h that the size of factorization obtained $N \leq 2^{h+K}T + 2^{h+K}$.

Indeed, for the base level, the statement is evident. For level h+1, whenever $\rho \in R''_{h+1,T}$, it can be factorized as $\rho = \rho_1 \rho_2 \dots \rho_n$ with $n \leq T$, and ρ_i of some durations T_i with $\sum_{i=1}^n T_i \leq T$ and satisfying the inductive hypothesis. Combining factorizations of ρ_i we get a factorization of ρ of size

$$\sum_{i=1}^{n} (2^{h+K}T_i + 2^{h+K}) \le 2^{h+K}T + 2^{h+K}n \le 2^{h+K}T + 2^{h+K}(T+2) = 2^{h+1+K}T + 2^{h+1+K},$$

which concludes the induction. Hence each run $\rho \in R_T \subset R''_{KT}$ has a factorization of size aT + a with $a = 2^{2K}$. Factors labeled by I have duration 0, and factors labeled by B correspond to one transition. As observed above, the sum of durations of factors labeled by F does not exceed d. Each word $w \in L_T(A)$ corresponds to some run $\rho \in R_T$, projecting its factorization yields a required factorization of w.

We can now construct an ε -net for L_T .

▶ Construction 55. The set $\mathcal{N}(\Sigma, T, \varepsilon, a, d)$ of timed words with duration $\leq T$ is constructed as follows:

- **a** at some aT + a positions multiple of ε we put a set of letters in Σ ;
- at some d/ε positions multiple of ε we put an element of $\mathcal{N}_{U\varepsilon}$,

where $\mathcal{N}_{U\varepsilon}$ is a finite ε -net for the set of all timed words of duration ≤ 1 as can be found in [4, Thm 2], its size is $2^{\#\Sigma/\varepsilon}$.

▶ **Lemma 56.** The set $\mathcal{N}(\cdot)$ is an ε -net for $L_T(\mathcal{A})$.

Proof. By virtue of Lem. 54, every word $w \in L_T$ can be factorized in $\leq aT + a$ factors with labels B, I, F. The total duration of factors with label F is bounded by d, thus only d/ε among them can have a "non-negligible" duration $\geq \varepsilon$.

The former bullet of Constr. 55 allows approximating letters labeled by B, instant words labeled by I, and factors with label F shorter than ε ; the latter — factors with label F longer than ε .

▶ **Lemma 57.** The cardinality $S(T,\varepsilon)$ of the set $\mathcal{N}(\cdot)$ satisfies the estimate

$$\lim_{T \to \infty} \log S(T, \varepsilon) / T = O(\log(1/\varepsilon)).$$

Proof. The following bound is immediate from Constr. 55:

$$S(T,\varepsilon) \leq \binom{T/\varepsilon}{aT+a} 2^{\#\Sigma(aT+a)} \cdot \binom{T/\varepsilon}{d/\varepsilon} 2^{(\#\Sigma/\varepsilon)(d/\varepsilon)}.$$

This gives an upper bound for the bandwidth:

$$\lim_{T \to \infty} \frac{\log S(T, \varepsilon)}{T} \le \lim_{T \to \infty} \frac{\log \binom{T/\varepsilon}{aT+a} + \#\Sigma(aT+a) + \log \binom{T/\varepsilon}{d/\varepsilon} + \log \left(\frac{\#\Sigma}{\varepsilon} \cdot \frac{d}{\varepsilon}\right)}{T} = \\
= \lim_{T \to \infty} \frac{\log \binom{T/\varepsilon}{aT+a}}{T} + \lim_{T \to \infty} \frac{\log \binom{T/\varepsilon}{d/\varepsilon}}{T} + \#\Sigma \cdot a. \quad (6)$$

We notice that whenever $A \gg B \gg 1$, then due to Stirling's formula $\binom{A}{B} \leq (Ae/B)^B$. Thus the first term of the sum in (6) does not exceed

$$\lim_{T \to \infty} \frac{(aT + a) \log \left(\frac{Te}{\varepsilon} \cdot \frac{1}{aT + a}\right)}{T} = a \log \frac{1}{\varepsilon} + a \log \frac{e}{a},$$

and the latter term gives

$$\lim_{T \to \infty} \frac{\frac{d}{\varepsilon} \cdot \log\left(\frac{Te}{\varepsilon} \cdot \frac{\varepsilon}{d}\right)}{T} = 0.$$

Gathering all the estimates we obtain the required bound:

$$\lim_{T \to \infty} \frac{\log S(T, \varepsilon)}{T} \le a \log \frac{1}{\varepsilon} + a \log \frac{e}{a} + \#\Sigma \cdot a = O\left(\log \frac{1}{\varepsilon}\right).$$

From Lemmas 56 and 57 follows the upper bound of Prop. 24.

B.3 Lower bound

▶ Proposition 58. If π is a structurally obese cycle whose obesity pattern involves vertices u and v, then the language $L_{\bar{\pi}^*}(u,\cdot)$ is obese.

Proof. We exhibit a construction for the two subcases of structural obesity. The second construction is actually the same as the first, with some runs through a foreign cycle inserted regularly along the runs of the ε -separated set.

Case of Type I structural obesity. For a cycle π exhibiting Type I structural obesity, vertex u can be the starting point of some realizations of duration 0, going back to u, and of some realizations of duration ≥ 1 , also going back to u. Hence by convex combination, it is possible to go from u to u with runs of any duration λ with $0 \leq \lambda \leq 1$.

In particular, for any ε , $0 < \varepsilon < \frac{1}{3|\pi|}$, there exists at least a run $\sigma(x)$ of $\bar{\pi}$ of length ε and a run $\rho_0(x)$ of length $3\varepsilon|\pi|$, both going back to u. Let us define $\rho_1(x) = \sigma(x)^{3|\pi|}$ (so that $\rho_0(x)$ and $\rho_1(x)$ have the same duration) and construct a function $f_x:\{0,1\}^* \to Runs_{\bar{\pi}^*}$, inductively: $f_x(\varepsilon) = \varepsilon$; for $b \in \{0,1\}$, $f_x(b) = \rho_b(x)$ and, for $b \in \{0,1\}$, $f_x(wb) = f_x(w)\rho_b(dst(f_x(w)))$. We show the language of the labels of runs in the image is ε -separated. Indeed, for 2 different words of this language, there are 2 different runs that are labeled by these words, and these runs are of the form $f_x(w)$ and $f_x(w')$ with $w \neq w'$; i.e. there is $i \in \mathbb{N}$ such that w[i] = 0 and w'[i] = 1. This means that the slices of $f_x(w)$ and $f_x(w')$ for the time interval $[3i\varepsilon|\pi|, 3(i+1)\varepsilon|\pi|]$ are $\rho_0(dst(f_x(w[0..i-1]))$ and $\rho_1(dst(f_x(w[0..i-1])))$. But $\rho_0(\dots)$ necessarily has a gap of length ε without any event, while $\rho_1(\dots)$ has at least one event in every interval of length ε , so at least one event of $\rho_1(\dots)$ cannot be matched in $\rho_0(\dots)$ by an event with a date closer than ε . Therefore this is also true of $f_x(w')$ and $f_x(w)$, thus $d(word(f_x(w)), word(f_x(w')) > \varepsilon$. Hence $\{word(f_u(w)) \mid w \in \{0,1\}^*\}_{\leq T}$ is an ε -separated subset of $L(\bar{\pi}^*)_{\leq T}$ and its size is at least $2^{\left\lfloor \frac{T}{3\varepsilon|\pi|}\right\rfloor}$.

Case of Type II structural obesity. For a Type II cycle, the construction is similar. For technical reasons, for a given ε , we will build an ε' -separated set with ε' the next value (slightly) larger than ε such that $\frac{1}{3\varepsilon'|\pi|}$ is integer. The difference with previous construction is that as long as only π is executed, instead of going back to u, we go farther and farther from u and closer and closer to v: a run of duration t from $(1-\lambda)u+\lambda v$ goes to $(1-\lambda-t)u+(\lambda+t)v$, provided $\lambda+t\leq 1$. So the analogs of runs ρ_0 and ρ_1 can only be executed $\frac{1}{3\varepsilon'|\pi|}$ times at most before it is necessary to "recharge" the clocks through the execution of a resetting cycle.

More precisely, the vertex u can be the starting point of a run of duration 0 through π , going back to u, as well of runs of duration ≥ 1 , going to v; while, through π , vertex v can only be the starting point of realizations of duration 0, all going back to v. By convex combinations, from any $x(\lambda) = (1 - \lambda)u + \lambda v$ it is thus possible to start a run of any duration $\mu \leq 1 - \lambda$ and reach the point $x(\lambda + \mu)$.

In particular, with ε' such that $0 < \varepsilon' < \frac{1}{3|\pi|}$, for any point $x(\lambda)$ of the segment $[u, \tilde{v}]$ (where $\tilde{v} = x(1 - 3\varepsilon'|\pi|)$), there exists at least a run $\sigma(x(\lambda))$ of $\bar{\pi}$ of duration ε' and a run $\rho_0(x(\lambda))$ of duration $3\varepsilon'|\pi|$. Let us define $\rho_1(x) = \sigma(x)^{3|\pi|}$ (so that $\rho_0(x)$ and $\rho_1(x)$ have the same duration) and remark that both $\rho_0(x(\lambda))$ and $\rho_1(x(\lambda))$ reach $x(\lambda + 3\varepsilon'|\pi|)$. Before we construct the new function $f_x : \{0,1\}^* \to Runs_{\bar{\pi}^*}$, let us choose a resetting cycle π_r such that its p-orbit has an edge from v to u, a run ρ_r that goes from v to u through π_r and τ_r the duration of this run.

Now we construct f_x : $f_x(\epsilon) = \epsilon$; with $b \in \{0,1\}$, if $x \in [u,\tilde{v}]$, $f_x(b) = \rho_b(x)$ and $f_x(bw) = \rho_b(x) f_{dst(\rho_b(x)}(w)$; otherwise $f_v(w) = \rho_r(v) f_u(w)$ (note that values of x in (\tilde{v},v) cannot be reached with this strategy; hence f is left undefined for this interval).

Here again, the language of the labels of the image is ε' -separated for the same reason as in the previous case. So $S_{\varepsilon}(T) \triangleq \{word(f_u(w)) \mid w \in \{0,1\}^*\}_{\leq T}$ is an ε' -separated (thus ε -separated) subset of $L((\bar{\pi}^{\frac{1}{3\varepsilon'|\bar{\pi}|}}\bar{\pi}_r)^*)_{\leq T}$ equal to the product of its slices corresponding to intervals $[ip, i(p+1)], i \in \mathbb{N}$, with $p = \left\lfloor \frac{1}{3\varepsilon'|\bar{\pi}|} \right\rfloor 3\varepsilon'|\bar{\pi}| + \tau_r = 1 + \tau_r$. Each slice contains exactly $2^{\frac{1}{3\varepsilon'|\bar{\pi}|}}$ elements, hence $\log \#S_{\varepsilon}(T) \geq \frac{1}{3\varepsilon'|\bar{\pi}|} \left\lfloor \frac{T}{p} \right\rfloor \geq \frac{1}{3\varepsilon'|\bar{\pi}|} \left\lfloor \frac{T}{1+\tau_r} \right\rfloor \geq \left(\frac{1}{3\varepsilon|\bar{\pi}|} - 1\right) \left\lfloor \frac{T}{1+\tau_r} \right\rfloor$. Hence $\mathcal{BH}_{\varepsilon}(L(\bar{\pi}^*)_{\leq T}) = \Theta(\frac{1}{\varepsilon})$ (the upper bound is given by comparison with the universal

Listing 2 Deciding structural obesity

```
Boolean function isAPathOrbit(e, h) if h = 0 return (e=1 || \exists \delta : \gamma_d(\delta) =e) for all orbits e1, e2 if e1·e2 = e && isAPathOrbit(e1, h-1) && isAPathOrbit(e2, h-1) return true return false Boolean function isStrucObese() for all orbits e if (obeseType1(e) && isAPathOrbit(e, \log \# M_d)) return true for all states q, int i,j \leq d_q for all orbits e1,e2 from q to q if (obeseType2(e1,i,j) && resettingPath(e2,j,i)&& isAPathOrbit(e1, \log \# M_d)) return true return false
```

language, having a bandwidth $\Theta(\frac{1}{\varepsilon})$).

▶ **Theorem 3.** Structural obesity of an RSTA of description size n with m clocks is decidable in $SPACE(\log^2(n)poly(m))$.

Proof sketch. The RsTA contains at most n locations. Each orbit can be represented as a couple of locations and an $(m+1) \times (m+1)$ -matrix with elements in Λ_d (a semiring of 4 elements). Thus $\#\mathbf{M}_d \leq n^2 \cdot 4^{(m+1)^2} = \mathsf{poly}(n)\mathsf{exp}(m)$. An automaton is structurally obese whenever it contains a path π with orbit $\gamma_d(\pi)$ having an obesity pattern. According to Lem. 17, only cycles up to length $\#\mathbf{M}_d$ need to be checked. We use now the reachability method [21, Sect. 7.3] to detect the obesity pattern in a space-efficient way, see Listing 2. An auxiliary function $\mathsf{isAPathOrbit(e,h)}$, true iff there is a path $\pi \in \gamma_d^{-1}(e)$ of length $\leq 2^h$ in the RsTA, is computed by a divide-and-conquer algorithm. The algorithm requires a call stack of depth $\log \#\mathbf{M}_d$, the size of each stack frame is $O(\log n) + \mathsf{poly}(m)$, thus the computation takes space $(O(\log n) + \mathsf{poly}(m)) \cdot \log \#\mathbf{M}_d = \log^2(n)\mathsf{poly}(m)$.

▶ **Proposition 25.** The language of a structurally obese RsTA is obese.

Proof. This is a direct corollary of Prop. 58. We obtain an ε -separated set S' from S, the ε -separated set of $L_{\bar{\pi}^*,T}(u,\cdot)$ constructed in the previous proof, by prepending a common prefix w to all its elements, such that w is the labeling of a run ρ going from an initial vertex of an initial location to the vertex u, through some path p.

This shows that $L_{\bar{p}\bar{\pi}^*}$ is obese, but since for each ε -separated subset of $L_{\bar{p}\bar{\pi}^*,T}$, $L_{p\pi^*,T}$ admits a ε' -separated subset of the same size (according to Lem. 35), this actually also proves that $L_{p\pi^*}$ is obese and hence its superset L(A).

C Proofs on meagerness

C.1 Monoid properties

▶ Proposition 15. γ_f is a monoid morphism.

Proof. Here again, we only need to prove, for any two edge sequences π_1, π_2 , that $\gamma_f(\pi_1 \circ \pi_2) = \gamma_f(\pi_1)\gamma_f(\pi_2)$. It is still trivially true if any of the two sequences is ϵ , or if $\pi_1 \circ \pi_2$ is not a path. So assume again both π_1 and π_2 are paths of length ≥ 1 such that $\pi_1 \circ \pi_2$ also is a path and let us call R the destination region of π_1 .

Let us choose any source vertex u and any destination vertex v for $\pi_1 \circ \pi_2$. Then $(\gamma_f(\pi_1)\gamma_f(\pi_2))[u,v]$ has to take one of the three values of Λ_d :

- If it is 0: the proof is the same as the case 0 in Prop. 22 (actually $\gamma_f(\pi)[u,v] = 0$ iff $\gamma_d(\pi)[u,v] = 0$ iff $\gamma(\pi) = 0$).
- If it is **narrow**, there is a single vertex w of R such that $\gamma_f(\pi_1)[u,w] = \gamma_f(\pi_2)[w,v] \neq 0$, and this value has to be **narrow**. But according to Lem. 33, a full region must be reachable from u through π_1 , and v must be reachable from a full region through π_2 ; the intersection of these two regions must also be a region, but this region contains only the vertex w, hence it is equal to $\{w\}$. So the only run from u to v through $\pi_1 \circ \pi_2$ is the concatenation of the only run from u to w and the only run from w to v. Hence $\gamma_f(\pi_1 \circ \pi_2)[u,v] = \mathbf{narrow} = (\gamma_f(\pi_1)\gamma_f(\pi_2))[u,v]$.
- If it is wide, because of semiring rules Table 1, there are 2 possibilites. Either wide is obtained by virtue of f-orbit multiplication, meaning there is w such that $\gamma_f(\pi_1)[u,w] =$ wide and $\gamma_f(\pi_2)[w,v] \neq 0$; since, by definition, $L_{\overline{\pi_1}}(u,w)$ contains several words, and thus so is the case of $L_{\overline{\pi_1}\circ\pi_2}(u,v)$. Or we obtained wide by virtue of f-orbit addition, meaning that there are w_1 and w_2 , $w_1 \neq w_2$, such that $\gamma_f(\pi_1)[u,w_1] = \mathbf{narrow}$, $\gamma_f(\pi_2)[w_1,v] = \mathbf{narrow}$, $\gamma_f(\pi_1)[u,w_2] = \mathbf{narrow}$ and $\gamma_f(\pi_2)[w_2,v] = \mathbf{narrow}$; this implies you can go from u to v by at least two different orbit paths, with different timings, hence $L_{\overline{\pi}(u,v)}$ contains at least two distinct words. Hence $\gamma_f(\pi_1\circ\pi_2)[u,v] = \mathbf{wide} = (\gamma_f(\pi_1)\gamma_f(\pi_2))[u,v]$.

C.2 Upper bound

Throughout this section, we reason about a structurally meager region-split automaton, we will abbreviate it as SMA. As we said, the case when the language $L_{\bar{\pi}}(x,x)$ is a singleton (typical for structurally meager automata) will play the key role. First, we prove that whenever π is an "idempotent cycle", the language mentioned above is indeed a singleton.

▶ **Lemma 59.** In an SMA, for any path π with idempotent orbit and a clock vector x in the closure of starting region of π , the language $L_{\bar{\pi}}(x,x)$ is either empty or a singleton.

Proof. First note that it is already known [22], for p-orbits, that $L_{\bar{\pi}}(x,x)$ is non-empty iff x is in the convex hull of some set of vertices with self-loops. So, it remains to prove that, in the non-empty case, the language is a singleton, which we do by using the fact that, in the f-orbit, all these self-loops are **narrow** when π is structurally meager.

Before going on, we show it is possible to assume w.l.o.g. that, for all factorization $\pi = \pi_1 \circ \pi_2$, the "shifted" cycle $\pi' = \pi_2 \circ \pi_1$ always has an idempotent f-orbit. If not, reason with π^p , where p is sufficiently large, observe it is still structurally meager and then, prove its polytope is a singleton. It implies that $L_{\bar{\pi}}(x,x)$ also is a singleton: indeed, if it were some non-singleton polytope, then $L_{\bar{\pi}^p}(x,x)$ would also include several words.

So we assume that π and all its "shifted" versions have idempotent p-orbits.

Suppose, towards a contradiction, that $L_{\bar{\pi}}(x,x)$ is neither empty nor a singleton. Then, this language has at least two words with different timings and, thus, with a first disagreeing timing, so there exist two paths π_1 and π_2 and an edge $\delta \in \Delta$ such that $\pi = \pi_1 \circ \delta \circ \pi_2$ a word w_1 through π_1 , two delays t and t' with 0 < t < t' and two words w_2 and w'_2 through

 π_2 such that both $w_1 \circ (lbl_{\delta}, t) \circ w_2$ and $w_1 \circ (lbl_{\delta}, t') \circ w'_2$ belong to $L_{\overline{\pi}}(x, x)$. Let us call y and y' the two clock vectors reached after reading respectively $w_1 \circ (lbl_{\delta}, t)$ and $w_1 \circ (lbl_{\delta}, t)$ through $\pi_1 \circ \delta$ from x.

If $y \neq y'$, then observe both vectors are reachable from each other through $\pi' = \pi_2 \circ \pi_1 \circ \delta$. The smallest affine set that contains the two vectors has dimension 1, hence, according to Cor. 42, the starting region of π' has at least one non-singleton SCC S. Since $\gamma_f(\pi')$ is idempotent, all self-loops $\gamma_f(\pi')[u, u]$ for $u \in S$ are **wide**, which is impossible in a structurally meager automaton.

If y=y', this means the edge δ can reach the same vector with different delays from the same originating vector, and hence that \mathfrak{g}_{δ} is not punctual and \mathfrak{r}_{δ} contains at least all clocks such that their value is not constrained inside a singleton in \mathfrak{g}_{δ} . In this case $S(dst_{\delta})$ also is a singleton, of the form $\{v\}$ with $v=(0,\ldots,0)$. So v is the only vertex of $\gamma_d(\pi_2\circ\pi_1\circ\delta)$. It necessarily has a self-loop, and since $L_{\pi_2\circ\pi_1\circ\delta}(v,v)$ contains several words, this loop is wide, which is the non-meagerness pattern.

We establish now a continuity result for the vicinity of a single-word path:

▶ Lemma 60. Given a path π in an RSTA and clock vectors x^0, y^0 , such that $L_{\bar{\pi}}(x^0, y^0) = \{w^0\}$ (a singleton), for any clock vectors x^1, y^1 such that both $||x^0 - x^1||_{\infty} < \varepsilon$, $||y^0 - y^1||_{\infty} < \varepsilon$, and for any $w^1 \in L_{\bar{\pi}}(x^1, y^1)$ we have $d(w^0, w^1) < 3\varepsilon$.

Proof. We fix notations for start and end regions of the path: $S(src_{\pi}) = R_s$, $S(dst_{\pi}) = R_d$, clearly $x^0, x^1 \in \bar{R}_s$ and $y^0, y^1 \in \bar{R}_d$ (otherwise the languages $L_{\bar{\pi}}(\cdot)$ would be empty.)

For any $x \in \bar{R}_s$, $y \in \bar{R}_d$ all words in $L_{\bar{\pi}}(x,y)$ have the form $(a_1,t_1),\ldots(a_n,t_n)$, where the untiming is always the same $a_1 \ldots a_n = lbl_w$, and the set of timings is defined by a parametric DBM A(x,y), as described in Prop. 28, and its projection to t_j is an interval $[L_j(x,y);U_j(x,y)]$ as described in Prop. 29. As we know $L_{\bar{\pi}}(x^0,y^0)$ is a singleton w^0 , thus for every j, the interval for t_j^0 is also a singleton: $L_j(x^0,y^0) = U_j(x^0,y^0) = t_j^0$. DBMs $A(x^1,y^1)$ and $A(x^0,y^0)$ satisfy the hypotheses of Cor. 30: they are equal in the middle and ε -close on border, thus by this Corollary,

$$t_j^0 - 3\varepsilon < L_j(x^1, y^1) \le t_j^1 \le U_j(x^1, y^1) < t_j^0 + 3\varepsilon,$$

hence $|t_j^0 - t_j^1| < 3\varepsilon$ (for any j). We conclude that $d(w^0, w^1) < 3\varepsilon$.

In order to characterize the duration of singletons we will naturally use the monoid \mathbf{M}_d again and define a new structural property. An idempotent e in \mathbf{M}_d will be called quick whenever all its self-loops are **instant**.

- ▶ **Lemma 61.** In an SMA, if for a cycle π with idempotent $e = \gamma_d(\pi)$ and a clock vector x the language $L_{\bar{\pi}}(x,x)$ is a singleton $\{w\}$, then
- if e is quick then $\tau(w) = 0$;
- \bullet otherwise $\tau(w) \geq 1$.

Proof. Let π be a cycle such that $\gamma_f(\pi)$ is idempotent and x a clock vector such that $L_{\pi}(x,x)$ is not empty. Necessarily, x is in a facet of the starting region of π , delimited by vertices all having self-loops in the (p, f, d)-orbits (the convex sums of triples (x, t, x) that satisfy the linear system of constraints of π also satisfy it; hence if x is a convex combination of vertices v such that at least one is such that (v, t, v) satisfies the system for several t, then x also accepts several t such that (x, t, x) satisfies the system). Let us call V the set of vertices of this facet.

If $\gamma_d(\pi)$ is quick (only has **instant** self-loops), then any element w of $L_{\bar{\pi}}(x,x)$ is such that $\tau(w) = 0$. If $\gamma_d(\pi)$ only has **slow** self-loops, then $\tau(w) \geq 1$. If $\gamma_d(\pi)$ has a **fast** self-loop, then P(x,x) is not a singleton. If $\exists u,v \in V$ s.t. $\gamma_d(\pi)[u,u] =$ **instant** and $\gamma_d(\pi)[v,v] =$ **slow**, then Lem. 47 applies, and we deduce that $\gamma_p(\pi)$ is complete and that therefore all the self-loops of $\gamma_f(\pi)$ are **wide** and that therefore $L_{\bar{\pi}}(x,x)$ is not a singleton.

We will now need a small technical lemma.

▶ Lemma 62. Let e be a realizable idempotent of \mathbf{M}_f . If u and v are transient vertices such that $e[u,v] = \mathbf{narrow}$, then there is a non-transient vertex w such that $e[u,w] = \mathbf{narrow}$ and $e[w,v] = \mathbf{narrow}$.

Proof. Assume there is no such vertex w. Then all paths from u to v only pass through transient vertices. Paths through transient vertices cannot go back or stay in place, thus by taking n the length of the longest such path, e^{n+1} does not contain a path from u to v, which contradicts e being idempotent.

The following lemma states that in any singleton run along a structurally meager idempotent cycle, all iterations except the first and the last one loop on a same clock configuration.

▶ Lemma 63. In an SMA, let $e \neq 0$ be an idempotent, let $\pi = \pi_1 \dots \pi_k$ such that all $\gamma(\pi_i) = e$ and $L_{\bar{\pi}}(x,y) = \{w\}$. Then there exists a factorization $w = w_1 \dots w_k$ and a clock vector z such that: $L_{\bar{\pi}_1}(x,z) = \{w_1\}$, for i = 2..k - 1 it holds that $L_{\bar{\pi}_i}(z,z) = \{w_i\}$, and finally $L_{\bar{\pi}_k}(z,y) = \{w_k\}$.

Proof. Assume such a path π and call $\rho = \rho_1 \dots \rho_k$ its unique run such that $Runs_{\bar{\pi}}(x,y) = \{\rho\}$ and $\rho_i \in Runs_{\bar{\pi}_i}$.

Necessarily there is $x_0 = x, x_1, x_2, \dots x_k = y$ such that $Runs_{\bar{\pi}}(x_i, x_{i+1}) = \{\rho_{i+1}\}.$

Moreover, the timing vector of ρ_i necessarily is a convex combination of timings of runs from a vertex u to another v in the support of e. Note that if there were several such runs from u to v, there would be several runs from x_i to x_{i+1} through π_{i+1} , hence several from x_i to x_{i+1} through x_i , which would contradict the hypothesis. Hence the convex combination has positive coefficients only for runs from u to v such that $e[u, v] = \mathbf{narrow}$.

Additionally, we show that, for i < k, ρ_i cannot be a convex combination including runs from vertex u to vertex v when v is transient.

Indeed, **narrow** edges can only go from u to v such that either u = v or at least one of the two vertices is transient (otherwise it would violate the idempotency of e).

Assume ρ_i can be written as a linear combination of vertex-to-vertex runs such that one goes from u to v with v transient. Hence x_i has a positive barycentric coordinate in v. So ρ_{i+1} is a convex combination of vertex-to-vertex runs which must include a run from v to another vertex w. Remark that $\rho_i\rho_{i+1}$ is also a realization of π and so it can be written as a linear combination of vertex-to-vertex runs too, this one such that one run goes from u to w (connecting the run from u to v and the one from v to w). By Lem. 62, there are non-transient vertices in the path from u to v and from v to w. But in this case, $e[u,w] = \mathbf{wide}$, so it means other runs than $\rho_i \rho_{i+1}$ can go from x_{i-1} to x_{i+1} which contradicts the hypothesis.

This means that the barycentric coordinate of x_i , 0 < i < k, for any transient vertex is 0. So for 1 < i < k, ρ_i is a convex combination of vertex-to-vertex runs starting from each non-transient vertex u and going to themselves (going to another non-transient is impossible because the edge is **wide**, going to a transient is forbidden, as we just proved), and the coefficient of these runs is necessarily equal to the weight of u in the barycentric coordinates of x_{i-1} , hence $x_i = x_{i-1}$.

Given a timed word $w = (a_1, t_1) \dots (a_n, t_n)$ we denote $||w|| = \#\{i \mid t_{i-1} < t_i\}$, i.e. the number of non-instant events in w. We will now estimate the number of non-instant delays in singletons w.r.t. their duration.

▶ **Lemma 64.** For any path π and clock vectors x, y in an SMA, if the language $L_{\bar{\pi}}(x, y) = \{w\}$ (a singleton), then $||w|| \le a\tau(w) + a$, with the constant a depending on the automaton.

Proof. We will use Simon's theorem for monoid \mathbf{M}_d and prove by induction, that whenever $\pi \in X_h$ (with X_h as in (4)) satisfies the Lemma's hypothesis for some x, y, w, then $||w|| \le 2^h \tau(w) + 2^h$. Since $h \le 3 \# \mathbf{M}_d$, this would imply the required result with $a = 2^{3 \# \mathbf{M}_d}$.

- The base case $\pi \in X_0$ is evident, $||w|| \le 1$.
- The first inductive case is $\pi = \pi_1 \pi_2$ with $\pi_1, \pi_2 \in X_h$ satisfying the inductive hypothesis. On the unique run along π from x to y, let z be the vector of clock values at the end of the prefix π_1 . It is easy to see that languages $L_{\bar{\pi}_1}(x,z)$ and $L_{\bar{\pi}_2}(z,y)$ should be some singletons w_1 and w_2 . Thus by inductive hypothesis $||w_1|| \leq 2^h \tau(w_1) + 2^h$, similarly for w_2 . As required

$$||w|| = ||w_1|| + ||w_2|| \le (2^h \tau(w_1) + 2^h) + (2^h \tau(w_2) + 2^h) = 2^h \tau(w) + 2^{h+1}.$$

The second inductive case is $\pi = \pi_1 \dots \pi_k$ with all $\pi_i \in X_h$ and $\gamma(\pi_i) = e$ for some non-quick idempotent e. By Lem. 63 we find a clock vector z, and timed words $w_1, \dots w_k$ such that $w = w_1 \dots w_k$ and $L_{\bar{\pi}_1}(x,z) = \{w_1\}$, $L_{\bar{\pi}_k}(z,y) = \{w_k\}$, and $L_{\bar{\pi}_i}(z,z) = \{w_i\}$ for each i = 2..k - 1. $||w_i|| \leq 2^h \tau(w_i) + 2^h$. By Lem. 61, $\tau(w_i) \geq 1$ for all i = 2..k - 1, hence $k-2 \leq \tau(w)$. By inductive hypothesis, $||w_i|| \leq 2^h \tau(w_i) + 2^h$ for any i. Summarizing, we get the required estimate:

$$|||w||| = \sum_{i=1}^{k} |||w_i||| \le \sum_{i=1}^{k} (2^h \tau(w_i) + 2^h) \le 2^h \tau(w) + 2^h (\tau(w) + 2) = 2^{h+1} \tau(w) + 2^{h+1}.$$

The last inductive case is $\pi = \pi_1 \dots \pi_k$ with all $\pi_i \in X_h$ and $\gamma(\pi) = \gamma(\pi_i) = e$ for a quick idempotent e. Let us factorize $w = w_1 \dots w_k$ according to Lem. 63. By Lem. 61, $\tau(w_i) = 0$ for i = 2..k - 1, thus $||w|| = ||w_1|| + ||w_k||$. Applying the inductive hypothesis to w_1 and w_k , we obtain again

$$||w|| = ||w_1|| + ||w_2|| \le (2^h \tau(w_1) + 2^h) + (2^h \tau(w_2) + 2^h) = 2^h \tau(w) + 2^{h+1}.$$

- ▶ **Definition 65.** Given a run ρ (on w) we define its signature $\sigma(\rho) = I_0 \delta_0 I_1 \delta_1 \dots \delta_k I_k$ with $k = ||w||, \forall j \in \{0..k\} I_j \subset \Sigma, \ \delta_j \in \Delta$ as follows:
- for every (maximal) sequence of instant (0 time) transitions in ρ we keep only the set of labels $I \subset \Sigma$;
- for every non-instant transition we keep only the edge $\delta \in \Delta$.

There are at most $2^{(k+1)\#\Sigma} \cdot \#\Delta^k$ different signatures of size k. Consequently, for some constant c the number of signatures of size $\leq k$ can be bounded by 2^{ck} .

▶ **Lemma 66.** In an SMA, for an idempotent $e \in \mathbf{M}_d$, the following holds. If we have two runs ρ_1, ρ_2 , with $\gamma(\operatorname{path}(\rho_i)) = e$, $\operatorname{src}_{\rho_i} = \operatorname{dst}_{\rho_i} = x$, and with the same signature $\sigma(\rho_1) = \sigma(\rho_2)$, then $\operatorname{d}(\operatorname{word}(\rho_1), \operatorname{word}(\rho_2)) = 0$.

Proof. The factors of the runs ρ_i corresponding to the I_j have exactly the same effects on clocks: they reset all clocks that are not 0 in the ending region of δ_{j-1} and become 0 in the

starting region of δ_j and leave all other clocks unchanged (an instant run cannot change a clock it does not reset).

So, after the prefix of ρ_1 up to δ_{j-1} , it is possible to run the sequence of instant transitions corresponding to I_j in ρ_2 , instead of the instant transitions corresponding to I_j in ρ_1 , and still reach the same state (and thus continue with the rest of the suffix of ρ_1 and still go back to x). By induction it is thus possible to build a run ρ'_1 from x to x such that $d(word(\rho_1), word(\rho'_1)) = 0$ and $path(\rho'_1) = path(\rho_2)$. But since $L_{path(\rho_2)}(x, x)$ is a singleton (structural thinness), it is necessary that $\rho'_1 = \rho_2$. Hence $d(word(\rho_1), word(\rho_2)) = 0$.

We are ready to upper bound information in singletons.

▶ Lemma 67. For any SMA, there exists a constant b such that, given any idempotent non-zero $e \in \mathbf{M}_d$ and clock vector x, the language $L_{eT}(x) \triangleq \bigcup_{\pi \in \gamma^{-1}(e)} L_{\pi T}(x,x)$ admits a 0-net $\mathcal{N}_1(e,x,T)$ of cardinality no more than 2^{bT+b} .

By definition, a 0-net of a set S has an element at distance 0 from each element of S.

Proof. By virtue of Lem. 59, $L_{\pi T}(x,x)$ is empty or singleton, thus for any timed word $w \in L_{eT}(x)$, Lem. 64 can be applied and $||w|| \le aT + a$. Thus for any run ρ on w, the size of the signature $\sigma(\rho)$ is also bounded by aT + a.

Now, by Lem. 66, to build the 0-net $\mathcal{N}_1(e, x, T)$, it suffices to take the trace of one representative run for each possible signature, there are $2^{c(aT+a)}$ possibilities.

▶ Construction 68 (Partition of regions). Given an SMA, an ε , and an idempotent non-zero $e \in \mathbf{M}_d$, take a path $\pi \in \gamma^{-1}(e)$ and its initial region $R = S(src_{\pi})$. According to Constr. 40 and Lem. 41, we select a family Λ of dim R independent Lyapunov functions on R. We partition the closed region \overline{R} into cells of clock vectors having the same ε -integer part for each Lyapunov function (i.e. the same integer vector $\lfloor \Lambda(x)/\varepsilon \rfloor$). We notice now that for an idempotent e the partition does not depend on the choice of the path π .

Note that each cell can be characterized by a tuple of integers $\varkappa = (\varkappa_1, \ldots, \varkappa_d) \in \mathbb{N}^d$ with d = dim(R) such that for each $x \in c$ and $1 \le i \le d$, $\varkappa_i = \lfloor \ell_i(x)/\varepsilon \rfloor - \lfloor \ell_{i-1}(x)/\varepsilon \rfloor$ where by convention we put $\ell_0(x) = 0$.

Given a vertex v, we denote by ℓ_v the smallest function of Λ such that $\ell_t(v_t) = 1$, and I_v its support. Abusively, we denote ℓ_{v-1} the element of vector Λ preceding ℓ_v (in the topological order of Constr. 40), we have that (for an SMA) $\lambda_v = \ell_v - \ell_{v-1}$. Finally, we denote the corresponding component of \varkappa by \varkappa_v .

▶ **Lemma 69.** With the hypotheses and notations from the construction above, if t is a transient vertex, and if x and y are clock vectors such that $P_{\bar{\pi}}(x,y) \neq \emptyset$ then $\ell_t(y) \leq \ell_t(x) - \lambda_t(x)$.

Proof. By virtue of Lem. 36, there exists a stochastic matrix P such that $\lambda(y) = \lambda(x)P$, with P_{uv} being positive only if there is an edge (u, v) in $\gamma(\pi)$, where $\lambda(x)$ is the vector of

barycentric coordinates of the clock vector x relative to the vertices of $S(src_{\pi})$.

$$1 - \ell_t(y) = \sum_{u \notin I_t} \lambda_u(y) = \sum_{u \notin I_t} \sum_v \lambda_v(x) P_{vu} = \sum_v \lambda_v(x) \left(\sum_{u \notin I_t} P_{vu} \right)$$

$$= \sum_{v \in I_t \setminus \{t\}} \lambda_v(x) \left(\sum_{u \notin I_t} P_{vu} \right) + \lambda_t(x) \left(\sum_{u \notin I_t} P_{tu} \right) + \sum_{v \notin I_t} \lambda_v(x) \left(\sum_{u \notin I_t} P_{vu} \right)$$

$$= \sum_{v \in I_t \setminus \{t\}} \lambda_v(x) \left(\sum_{u \notin I_t} P_{vu} \right) + \lambda_t(x) + \sum_{v \notin I_t} \lambda_v(x) \ge \lambda_t(x) + \sum_{v \notin I_t} \lambda_v(x)$$

$$= \lambda_t(x) + 1 - \sum_{v \in I_t} \lambda_v(x).$$

We conclude.

- ▶ **Lemma 70.** For any cell $c_{e\varepsilon\varkappa}$ the following alternative holds:
- if for some transient vertex t we have $\varkappa_t \geq 1$, then for any $x, y \in c_{e\varepsilon\varkappa}$ it holds that $L_{\pi}(x, y) = \emptyset$. We call such cell $c_{e\varepsilon\varkappa}$ transient.
- Otherwise (if for all transient vertices v_t we have $\varkappa_t = 0$), then there exists an $x \in c_{e\varepsilon\varkappa}$ having $L_{\bar{\pi}}(x,x) \neq \emptyset$ (and thus a singleton). We denote it $x_{e\varepsilon\varkappa}$ and call such cells recurrent.

Proof. In the former case, following the previous lemma, we have that $\ell_t(y) \leq \ell_t(x) - \lambda_t(x) = \ell_{t-1}(x)$. Then

$$\left| \frac{\ell_t(y)}{\varepsilon} \right| \le \left| \frac{\ell_{t-1}(x)}{\varepsilon} \right| < \left| \frac{\ell_{t-1}(x)}{\varepsilon} \right| + 1 \le \left| \frac{\ell_{t-1}(x)}{\varepsilon} \right| + \varkappa_t.$$

But in order for y to be in $c_{e\varepsilon\varkappa}$, we must have, among other properties, that $\left\lfloor \frac{\ell_{t-1}(y)}{\varepsilon} \right\rfloor = \left\lfloor \frac{\ell_{t-1}(x)}{\varepsilon} \right\rfloor$ and $\left\lfloor \frac{\ell_{t}(y)}{\varepsilon} \right\rfloor = \left\lfloor \frac{\ell_{t-1}(y)}{\varepsilon} \right\rfloor + \varkappa_{t}$, which, as per the above inequality, does not hold.

In the latter case, we can choose values of $\Lambda(\cdot)$ corresponding to the cell $c_{e\varepsilon\varkappa}$ with $\ell_t(\cdot) = \ell_{t-1}(\cdot)$ (and thus $\lambda_t(\cdot) = 0$) for all transient vertices t. By Lem. 43 there exists a point x in the cell corresponding to these values of Λ . Such an x is a combination of vertices with self-loops, which gives $L_{\bar{\pi}}(x,x) \neq \emptyset$, as explained in the proof of Lem. 59.

- ▶ **Lemma 71.** For fixed e and ε , cells $c_{e\varepsilon\varkappa}$ have the following properties:
- the number of cells is at most $f_1(\varepsilon) \triangleq \begin{bmatrix} 1 \\ \varepsilon \end{bmatrix}^{\dim R}$;
- the diameter of each cell is at most $K\varepsilon$, with K from Lem. 43;
- there is a partial order over cells: $c^1 \succeq c^2$, whenever some $x^1 \in c^1$, $x^2 \in c^2$ satisfy $\Lambda(x^1) \geq \Lambda(x^2)$. The language $L_{\bar{\pi}}(c^1, c^2)$ can be non-empty only if $c^1 \succeq c^2$.

Proof. The first statement is trivial, the second is a consequence of Lem. 43, and the third follows from Def. 37.

- ▶ **Definition 72.** A non-empty run ρ is called ε -still if
- $= e = \gamma_f(path(\rho))$ is an idempotent;
- and ρ goes from a recurrent cell $c_{e\varepsilon\varkappa}$ to the same cell.

We denote the set $word(\rho)$ over all ε -still runs by $Still_{\varepsilon}$, and its restriction to words of duration $\leq T$ by $Still_{\varepsilon T}$.

▶ Construction 73 (Net for Still). The net $\mathcal{N}_2(\varepsilon,T)$ is built as follows. For each recurrent cell $c_{e\varepsilon\varkappa}$ let $x_{e\varepsilon\varkappa}$ be as in Lem. 71. Take

$$\mathcal{N}_2 \triangleq \bigcup_{\substack{e=ee \ c_{e\varepsilon\varkappa} recurrent}} \mathcal{N}_1(e, x_{e\varepsilon\varkappa}, T+1).$$

▶ **Lemma 74.** Suppose $3K\varepsilon < 1$. The set $\mathcal{N}_2(\varepsilon, T)$ is a $3K\varepsilon$ -net for the language $Still_{\varepsilon T}$ of size $f_2(\varepsilon)2^{bT}$ (for some function f_2).

Proof. Let $w \in Still_{\varepsilon T}$, by definition $w = word(\rho)$ with ρ from some $x \in c_{e\varepsilon \varkappa}$ to some $y \in c_{e\varepsilon \varkappa}$ (in the same recurrent cell). Let $\pi = path(\rho)$. By Lem. 70, $L(\pi, x_{e\varepsilon \varkappa}, x_{e\varepsilon \varkappa})$ is some singleton $\{w'\}$. By Lem. 71, $||x - x_{e\varepsilon \varkappa}||_{\infty} \le K\varepsilon$ and $||y - x_{e\varepsilon \varkappa}||_{\infty} \le K\varepsilon$, thence by Lem. 60 $d(w, w') \le 3K\varepsilon$. This implies that $\tau(w') \le T + 3K\varepsilon < T + 1$, and by Lem. 67 there exists $w'' \in \mathcal{N}_1(e, x_{e\varepsilon \varkappa}, T + 1)$, such that d(w', w'') = 0. Hence $d(w, w'') \le 2K\varepsilon$ and by construction $w'' \in \mathcal{N}_2$. Thus \mathcal{N}_2 is indeed a $3K\varepsilon$ -net for the language $Still_{\varepsilon T}$. The size estimation is obtained by multiplying the size of \mathcal{N}_1 by the number of cells.

We proceed now with factorizing words in $Still_{\varepsilon}$ fragments and transitions between them.

▶ Lemma 75. For an SMA A there exists a function $f_3(\varepsilon)$, such that for any ε every word $w \in L(A)$ admits a factorization $w = w_1 \dots w_N$ with $N \leq f_3(\varepsilon)$, with each w_i either having one letter or belonging to $Still_{\varepsilon}$.

Proof. We will use Simon's theorem (for monoid \mathbf{M}_d) and prove by induction that for every path $\pi \in X_h$, every $w \in L_{\pi}$ admits a factorization with $N \leq (2f_1(\varepsilon))^h$. Since the height h is bounded by K, this yields the required bound $N < f_3(\varepsilon)$ with $f_3(\varepsilon) = (2f_1(\varepsilon))^K$

- In the base case $\pi \in X_0$, the word w has one letter and its factorization is trivial (N=1).
- In the first inductive case $\pi = \pi_1 \pi_2$, with $\pi_1, \pi_2 \in X_h$. Consequently $w = w_1 w_2$ with $w_i \in L_{\pi_i}$, the words w_1, w_2 admitting factorizations with $(2f_1(\varepsilon))^h$ factors. Concatenating those we obtain a factorization for w with $2(2f_1(\varepsilon))^h < (2f_1(\varepsilon))^{h+1}$ factors.
- In the other inductive case, there exists an idempotent e and $\pi = \pi_1 \dots \pi_J$, with $\gamma_d(\pi_i) = e$ and $\pi_i \in X_h$. Passing to runs, we get $\rho = \rho_1 \dots \rho_J$, with $w = word(\rho)$, and $path(\rho_i) = \pi_i$. The factorization in ρ_i is too long and too precise, we will partition it into fewer clusters. Let z_i be clock vectors (traversed by the run ρ), such that $src_{\rho_i} = z_{i-1}$ and $dst_{\rho_i} = z_i$. Let c_i be the cell of z_i , that is the unique $c_{e\varepsilon\varkappa}$ containing z_i . By Lem. 71 $c_0 \succeq c_1 \succeq \cdots \succeq c_J$, and there are at most $f_1(\varepsilon)$ different cells in this sequence, let us denote them c^i . It means that the sequence can be written as

$$c^{0} = c^{0} = \dots c^{0} \succ c^{1} = \dots c^{1} \succ c^{2} \dots c^{M}$$

with $M < f_1(\varepsilon)$. Notice that only recurrent cells can be repeated. We will concatenate together all runs ρ_j from c^i to itself, and call the result of concatenation ζ^i ; we also denote ρ_j from c^i to c^{i+1} by ρ^i . This yields a new factorization:

$$\rho = \zeta^0 \rho^0 \zeta^1 \rho^1 \dots \zeta^M, \quad M < f_1(\varepsilon)$$

with all ζ^i from a cell to the same cell, and all $path(\rho^i) \in X_h$. Applying the projection $word(\cdot)$ we get a factorization $w = u^0v^0u^1v^1\dots u^M$ with all $u^i \in Still_{\varepsilon}$ (or equal to the empty word) and all v^i satisfying the inductive hypothesis. Factorizing each v^i according to the inductive hypothesis, we get a required factorization of w into $f_1(\varepsilon) + f_1(\varepsilon)(2f_1(\varepsilon))^h < (2f_1(\varepsilon))^{h+1}$ factors. This concludes the proof.

At the last step we build an $O(\varepsilon)$ -net for all the words $w \in L_T(\mathcal{A})$:

- ▶ Construction 76 (Net for L_T). The elements of the net $\mathcal{N}_3(\varepsilon,T)$ are generated as follows
- choose $N \le f_3(ε)$ and N dates $0 \le t_1 \le t_2 \le \cdots t_N \le T$ which are multiples of ε;
- \blacksquare at some of those dates put a letter in Σ ;
- at all other t_i insert an element of $\mathcal{N}_2(\varepsilon, t_{i+1} t_i + \varepsilon)$.

We remark that elements of the second kind can slightly overlap with the rest of the word, i.e., go beyond t_{i+1} .

▶ **Lemma 77.** $\mathcal{N}_3(\varepsilon, T)$ is a $(3K + 1/2)\varepsilon$ -net for L_T (for ε small enough).

Proof. Let $w \in L_T(\mathcal{A})$, we will explain how to find a world in $\mathcal{N}_3(\varepsilon, T)$ at a small distance from w.

To that aim, consider the factorization $w = w_1 \dots w_N$ provided by Lem. 75. For each factor w_i of the first kind (containing a delay and one letter a_i) let $t_i = \tau(w_1 \dots w_i)$, i.e. the date when this letter happens in the word w. For each factor of the second kind $(w_i \in Still_{\varepsilon})$ let $t_i = \tau(w_1 \dots w_{i-1})$ the starting date of factor w_i . We notice that $0 \le t_1 \le t_2 \le t_N$, and that for any factor w_i of the second kind $\tau(w_i) \le t_{i+1} - t_i$.

For every t_i , let t_i' be the closest ε -integer. We have that $0 \le t_1' \le t_2' \le t_N'$, and for any factor w_i of the second kind $\tau(w_i) \le t_{i+1}' - t_i' + \varepsilon$. For any such w_i of the second kind let w_i' be its $2K\varepsilon$ -approximation from the net $\mathcal{N}_2(\varepsilon, t_{i+1}' - t_i' + \varepsilon)$.

We are ready to build an approximated word w'. For each factor w_i of the first kind, the word w' contains a_i at date t'_i . For each factor w_i of the second kind, the word w' contains the approximation w'_i starting at t'_i .

By construction, $w' \in \mathcal{N}_3(\varepsilon, T)$. On the other hand, $d(w, w') \leq \varepsilon$, indeed each factor of the first kind in w is shifted by at most $\varepsilon/2$, and each factor of the second kind in w is $3K\varepsilon$ -approximated and shifted by at most $\varepsilon/2$.

▶ Lemma 78. The cardinality of $\mathcal{N}_3(\varepsilon,T)$ is bounded by $T^{f_3(\varepsilon)}f_4(\varepsilon)2^{bT}$.

Proof. There are $\binom{T/\varepsilon}{f_3(\varepsilon)}$ ways of choosing $f_3(\varepsilon)$ dates, which can be bounded by $(T/\varepsilon)^{f_3(\varepsilon)}$. For each of those one can choose a letter in Σ , or decide to put an element of \mathcal{N}_2 , or put nothing (if there are less than $f_3(\varepsilon)$ factors), this gives $(\#\Sigma + 2)^{f_3(\varepsilon)}$ possibilities. For elements of \mathcal{N}_2 the number of possibilities is

$$\prod_{\text{some } j < f_3(\varepsilon)} f_2(\varepsilon/2) 2^{b(t_{j+1} - t_j + \varepsilon)} \le 2^{bT} 2^{b\varepsilon f_3(\varepsilon)} f_2(\varepsilon/2)^{f_3(\varepsilon)}.$$

Multiplying everything together we get an expression of the required form $T^{f_3(\varepsilon)}f_4(\varepsilon)2^{bT}$ for an appropriate f_4 .

▶ Proposition 18. The timed language of a structurally meager RSTA is meager.

Proof. Let K' = 3K + 1/2. By the previous lemma, the bandwidth is bounded by

$$\lim_{T \to \infty} \frac{\log \left(T^{f_3(\varepsilon/K')} f_4(\varepsilon/K') 2^{bT} \right)}{T} = b.$$

C.3 Proof details for lower bound

▶ Lemma 79. If a cyclic path π in an RSTA is not structurally meager, then $\mathcal{BC}_{\varepsilon}(L_{\pi^*}) = \Omega(\log(1/\varepsilon))$, and hence L_{π^*} is not a meager language.

Proof. Without loss of generality, we assume $\gamma_f(\pi)$ is idempotent. It is not structurally meager, hence it has a **wide** self-loop on some vertex v, implying dim $P_{\pi}(v,v) \geq 1$. Now consider $\gamma_d(\pi)[v,v]$. It cannot be **instant** or 0, since $\gamma_f[v,v] = \text{wide}$. If it is **fast**, then π is structurally obese, thus its language is obese (and not meager). So we consider the remaining case: $\gamma_d(\pi)[v,v] = \text{slow}$: i.e. the cycle cannot be executed in less than one time unit. So not only the set of possible timings has a dimension of at least 1, but also all timings have a total duration of at least 1. Without loss of generality, we can assume we can select a segment of realizations such that all its timing vectors have the same total duration (if not, take any segment: take 2 runs ρ_1 and ρ_2 in this segment, going from v to itself, with different total duration; from now on instead of π , we consider π^2 and the segment delimited by the timings of the runs $\rho_1\rho_2$ and $\rho_2\rho_1$).

We consider the set of runs $\sigma_k = \frac{k}{n}\rho_1 + \frac{n-k}{n}\rho_2$ and look at the first transition occurring at a different date in ρ_1 and ρ_2 (call its index m), assuming wlog it occurs earlier in ρ_1 and let us call τ the difference between the date of m-th event in ρ_2 and in ρ_1 . If k > j, in σ_j , the mth event cannot be matched in σ_k with a time difference smaller than τ/n , implying that $d(\sigma_j, \sigma_k) \geq \vec{d}(\sigma_j, \sigma_k) \geq \tau/n$. Now, considering arbitrary repetitions, we define the language $S_n \triangleq ((word(\sigma_k))_{k=1..n})^*)$. We remark there is also a distance of at least τ/n between any two words of S_n : indeed the factors σ_k having all the same duration, the $i|\pi| + m$ th letter of a word cannot be matched with any letter of index smaller than $i|\pi|$ in another word. Also, no letter of index greater than $(i+1)|\pi|$ can be matched with this letter. Hence S_n is τ/n separated. So, for any arbitrary ε (smaller than τ), there is an ε -separated subset of $L(\bar{\pi}^*)_T$ of size $\left|\frac{\tau}{\varepsilon}\right|^{\lfloor \frac{T}{\tau(\rho_1)}\rfloor}$.

Using Lem. 35, it follows this is also true of $L(\pi^*)_T$.

Prop. 19 follows immediately.

D PSPACE-hardness is not so hard

▶ Theorem 5. A timed language accepted by a DTA is either meager, normal, or obese. Given a DTA, the three problems "is it meager?", "is it normal?", "is it obese?" are PSPACE-complete.

We prove here only the PSPACE-hardness of the three problems, denoted Meager, Normal and Obese. PSPACE membership is proved in Sect. 5.

Proof. We proceed by reduction from the following PSPACE-complete problem [1]:

Reach: given a DTA \mathcal{A} and two locations p, q, does there exist a run from state (p, 0) to location q?

From an instance of Reach we build two DTA ${\mathcal B}$ and ${\mathcal C}$ as follows:

- \mathcal{B} is obtained from \mathcal{A} by adding one more location f, and two edges $q \to f$ and $f \to f$ with a new label d, no reset and the guard **true**. The initial condition I(p) is x = 0, the final condition F(f) is **true**, and all the other initial and final conditions are **false**.
- \mathbf{C} is obtained from \mathcal{B} by adding one more location s, and two edges $(s, s, d, x_1 > 1, \{x_1\})$, $(s, p, d, x_1 = 1, X)$ where X is the set of clocks in \mathcal{A} . The initial state is I(s) = (x = 0), the final ones are $F(f) = F(s) = \mathbf{true}$.

By construction

whenever $\mathsf{Reach}(\mathcal{A}, p, q)$, both automata \mathcal{B} and \mathcal{C} are obese: indeed, accepting runs can traverse \mathcal{A} from p to q and then go to the obese loop on location f;

otherwise, \mathcal{B} has empty language, and \mathcal{C} is normal: indeed, the former automaton has no accepting run, and the latter can loop on s producing a normal language.

Thanks to the polynomial reduction $(\mathcal{A}, p, q) \mapsto \mathcal{B}$ we have Reach \leq Obese and $(\neg \text{Reach}) \leq$ Meager. The other reduction $(\mathcal{A}, p, q) \mapsto \mathcal{C}$ yields $(\neg \text{Reach}) \leq$ Normal. Thus the three problems Obese, Meager and Normal are PSPACE-hard.

E Extending to more general automata

▶ Corollary 26. The same classification and complexity results also hold for nondeterministic timed automata without ϵ -transitions.

Proof sketch. Given a non-deterministic timed automaton \mathcal{B} (without ϵ -transitions) with alphabet Σ , it is always possible to relabel its transitions with a larger alphabet Γ to make it deterministic (for example choosing distinct letters for each transition), let us call this deterministic automaton \mathcal{A} . It holds that $L(\mathcal{B}) = \mu(L(\mathcal{A}))$, where μ is a letter-to-letter renaming from Γ to Σ . Thm. 5 applies to \mathcal{A} , and we claim that $L(\mathcal{B})$ belongs to the same class as $L(\mathcal{A})$.

To prove the claim we notice first that if \mathcal{N} is an ε -net for $L_T(\mathcal{A})$, then $\mu(\mathcal{N})$ is a smaller ε -net for $L_T(\mathcal{B})$, hence

$$\mathcal{BH}(L(\mathcal{B})) \le \mathcal{BH}(L(\mathcal{A})). \tag{7}$$

if \mathcal{A} is meager, then \mathcal{B} is also meager by virtue of (7);

- if \mathcal{A} is obese, then it is structurally obese by Thm. 4, and thus the proof of Prop. 25 can be applied, yielding one cycle π in \mathcal{A} and a big ε -separated subset $\mathcal{S} \subset L_{T,\pi^*}(\mathcal{A})$. Since all elements of S follow the same cycle, the restriction of μ to S is injective, moreover $\mu(S)$ is also a big ε -separated subset in $L_T(\mathcal{B})$, thus \mathcal{B} is obese.
- if \mathcal{A} is normal, then by (7) $\mathcal{BH}(\mathcal{B}) = O(\log 1/\varepsilon)$. On the other hand, \mathcal{A} is not structurally meager, and the proof of Prop. 19 yields a big ε -separated subset along one cycle. Similarly to the previous case this implies that $L_T(\mathcal{B})$ is big enough: $\Omega(\log 1/\varepsilon)$.

F Completing the proof of the last theorem

▶ **Theorem 6.** An RSTA with only bounded non-punctual guards cannot be both thick and meager. Also, it cannot be thin and obese of Type I. All other combinations are possible.

The only missing element in the proof sketch in Sect. 5.2 concerned the incompatibility of obesity Type I with thinness.

Proof. Consider a thick automaton, with its forgetful cycle π . If its orbit has a single vertex, since π has a non-punctual transition it can be realized by several runs, hence the only edge of $\gamma_f(\pi)$ is **wide**. Otherwise, $\gamma(\pi)$ has a complete SCC of size at least 2, and then all its vertices have **wide** self-loops in $\gamma_f(\pi^2)$.

Now consider an automaton with structurally obese cycle π of Type I. By definition, $\gamma_d(\pi)$ has both an **instant** and a **slow** self-loop. According to Lem. 47, π must be forgetful and the automaton cannot be thin.