Distance on Timed Words and Applications

Eugene Asarin[1], Nicolas Basset[2], and Aldric Degorre[1](✉)

[1] IRIF, University Paris Diderot and CNRS, Paris, France
aldric.degorre@irif.fr
[2] VERIMAG, University Grenoble Alpes and CNRS, Grenoble, France

Abstract. We introduce and study a new (pseudo) metric on timed words having several advantages:
- it is global: it applies to words having different number of events;
- it is realistic and takes into account imprecise observation of timed events; thus it reflects the fact that the order of events cannot be observed whenever they are very close to each other;
- it is suitable for quantitative verification of timed systems: we formulate and solve quantitative model-checking and quantitative monitoring in terms of the new distance, with reasonable complexity;
- it is suitable for information-theoretical analysis of timed systems: due to its pre-compactness the quantity of information in bits per time unit can be correctly defined and computed.

1 Introduction

Timed words are sequences of events (from a finite alphabet Σ) with their dates (from $\mathbb{R}^+$). Such words, sets thereof (timed languages) and automata working on them (timed automata) constitute a relevant abstraction level for modelling and verification of real-time systems and an attractive research area since the founding work [1].

It is commonly accepted that timed words can be produced or observed with a certain precision, and several works considered approximate verification of timed systems, fixing a distance and a precision on timed words. In most cases the uniform distance on dates (or delays) for words with n events is considered, see e.g. tube languages [11], or robustness [14]. In [4,7] we have studied symbolic dynamics of timed systems w.r.t. a similar distance, and in [3] we have applied it to the following question: "what is the amount of information in timed words of language L, of length n, with precision ε".

However, on the information theory side, all the distances considered up to now are well adapted to analysis of quantity of information for a fixed number of events n, or its asymptotic behaviour w.r.t. n. But those distances are not suitable for the analysis on information for a given time T (or asymptotically w.r.t. $T \to \infty$). To perform such an analysis, we need a unique distance on timed words with different numbers of events. Furthermore, this metric should

D. N. Jansen and P. Prabhakar (Eds.): FORMATS 2018, LNCS 11022, pp. 199–214, 2018.
https://doi.org/10.1007/978-3-030-00151-3_12

be compact (for a given duration T), if we want the amount of information transmitted in T seconds with precision ε to be finite.

On the practical side, if we observed timed words with some finite precision (say $0.01s$), then it would be difficult to distinguish the order of close events, e.g. detect the difference between

$$w_1 = (a, 1), (b, 2), (c, 2.001) \text{ and } w_2 = (a, 1.001), (c, 1.999), (b, 2.001).$$

Moreover, it is even difficult to count the number of events that happen in a short lapse of time, e.g. the words w_1, w_2 look very similar to

$$w_3 = (a, 1), (c, 1.999), (c, 2), (b, 2.001), (c, 2.0002).$$

A slow observer, when receiving timed words w_1, w_2, w_3 will just sense an a at the date ≈ 1 and b and c at the date ≈ 2.

As the main contribution of this paper, we introduce a metric on timed words (with non-fixed number of events) for which w_1, w_2, w_3 are very close to each other. We believe that this metric is natural and sets a ground for approximate model-checking and information theory of timed languages w.r.t. time (and not only number of events).

We present the first technical results concerning this distance:

- its simple geometrical properties;
- techniques of quantitative model-checking and monitoring, and complexity estimates thereof (the complexity of standard problems is quite moderate: PSPACE or sometimes NP);
- proof of compactness of this distance, and analysis of information contents of some important languages.

The paper is structured as follows: after some preliminaries in Sect. 2 we introduce our main new notion of distance between timed words in Sect. 3. We analyse problems of quantitative model-checking (with respect to this distance) in Sect. 4 and those of information content in Sect. 5. We conclude with some perspectives in Sect. 6.

2 Preliminaries

We suppose that the reader is acquainted with timed automata (and region equivalence), [1]. Nonetheless, here we fix notation and provide main definitions. We also provide basic facts and notions on pre-compact spaces and two information measures thereof.

2.1 Timed Words and Timed Languages

A *timed word* of length n over an alphabet Σ is a sequence $w = t_1 a_1 \dots t_n a_n$, with $a_i \in \Sigma, t_i \in \mathbb{R}$ and $0 \leq t_1 \leq \dots \leq t_n$. Here t_i represents the dates at which the event a_i occurs (this definition rules out timed words ending by a time delay). We also adopt the convention that $t_0 = 0$. A *timed language* L is a set of timed words.

Timed word projection. The projection $p_\Sigma(u)$ of a timed word u erasing alphabet Σ consists in the word u where the events with labels in Σ are hidden. Recursively:

$$p_\Sigma(\varepsilon) = \varepsilon \ ; \ p_\Sigma(ta) = \begin{cases} ta \text{ if } a \notin \Sigma \\ \varepsilon \text{ if } a \in \Sigma \end{cases} ;$$

$$p_\Sigma(tau) = p_\Sigma(ta)p_\Sigma(u) = \begin{cases} ta \cdot p_\Sigma(u) \text{ if } a \notin \Sigma \\ \quad p_\Sigma(u) \text{ if } a \in \Sigma \end{cases}$$

This definition is lifted to timed languages the natural way: the projection of a language L is the set of the projections of all words $u \in L$.

2.2 Timed Graphs and Timed Automata

A *clock* is a variable ranging over $\mathbb{R}_{\geq 0}$ (non-negative reals). A *clock constraint* $\mathfrak{g} \in G_C$ over a set of clocks C is a conjunction of finitely many inequalities of the form $x \sim c$ or $x \sim y$, where x and y are clocks, $\sim \in \{<, \leq, =, \geq, >\}$ and $c \in \mathbb{Q}_{\geq 0}$. A *clock reset* $\mathfrak{r} \in R_C$ is determined by a subset of clocks $B \subset C$, it resets to 0 all the clocks in B and does not modify the values of the others.

A *timed graph* (TG) is a triple $\Gamma = (V, C, E)$. Its elements are respectively the finite set of locations, the finite set of clocks (let its cardinality be d) and the transition relation (timed edges). A *state* of Γ is a pair $(v, \mathbf{x})$ of a control location $v \in V$ and a vector of clock values $\mathbf{x} \in \mathbb{R}^{\mathrm{d}}$. Elements of $\mathtt{E}$ are *transitions*, i.e. tuples $(v, \mathfrak{g}, \mathfrak{r}, v') \in V \times G_C \times R_C \times V$ denoting the possibility, at location v when the clock vector satisfies the *guard* $\mathfrak{g}$, to apply the clock reset $\mathfrak{r}$ and then go to location v'.

A *timed automaton* (TA) is a tuple $\mathcal{A} = (Q, \Sigma, C, \Delta, q_0, F)$ such that Σ, Q, C are finite sets, $q_0 \in Q$, $F \subset Q \times G_C$ and $\Delta \subset Q \times (\Sigma \cup \{\varepsilon\}) \times G_C \times R_C \times Q$. Hence, if we define $E = \{(q, \mathfrak{g}, \mathfrak{r}, q') | \exists a \in \Sigma \cup \{\varepsilon\} \text{ s.t. } (q, a, \mathfrak{g}, \mathfrak{r}, q') \in \Delta\}$, then (Q, C, E) is a TG called the *underlying timed graph* of $\mathcal{A}$.

Q is the set of locations of the TA, Σ, the alphabet of its symbols, Δ, its transition relation, q_0, its initial location and F, its final condition.

Intuitively, a TA reads a timed word, which will make a pebble move from state to state, starting from $(q_0, \mathbf{0})$, and accepts or rejects the word depending on whether the last visited state satisfies F or not.

More formally, a *run* of $\mathcal{A}$ along a *path* $\pi = \delta_1 \ldots \delta_n \in \Delta^n$ has the form

$$(q_{i_0}, \mathbf{x}_0) \xrightarrow{t_1 a_1} (q_{i_1}, \mathbf{x}_1) \xrightarrow{t_2 a_2} \cdots \xrightarrow{t_n a_n} (q_{i_n}, \mathbf{x}_n),$$

where, for all $j \in 1..n$, $\delta_j = (q_{i_{j-1}}, a_j, \mathfrak{g}, \mathfrak{r}, q_{i_j}) \in \Delta$,

- $\mathbf{x}_{j-1} + (t_j - t_{j-1})\mathbb{1} \models \mathfrak{g}$ with $\mathbb{1}$ denoting the vector $(1, \ldots, 1)$,
- and $\mathbf{x}_j = \mathfrak{r}\,(\mathbf{x}_{j-1} + (t_j - t_{j-1})\mathbb{1})$.

When $q_{i_0} = q_0$ is the initial state, $\mathbf{x}_0$ is $\mathbf{0}$ and F contains a couple $(q, \mathfrak{g})$ with $q_{i_n} = q$ and $\mathbf{x}_n$ satisfying $\mathfrak{g}$, then the timed word $p_\varepsilon(t_1 a_1 \ldots t_n a_n)$ is said to be *accepted* by $\mathcal{A}$. The set of all such words is the language $L(\mathcal{A})$ accepted by $\mathcal{A}$.

Finally, the *granularity* of a timed graph G is the largest rational number $g(G)$ such that any constant k appearing in the guards of G satisfies $\frac{k}{g(G)} \in \mathbb{N}$.

2.3 Synchronized Product of TA

The *synchronized product* of the TA $\mathcal{A} = (Q_{\mathcal{A}}, \Sigma_{\mathcal{A}} \cup \Sigma_S, C_{\mathcal{A}}, \Delta_{\mathcal{A}}, q_{0\mathcal{A}}, F_{\mathcal{A}})$ and $\mathcal{B} = (Q_{\mathcal{B}}, \Sigma_{\mathcal{B}} \cup \Sigma_S, C_{\mathcal{B}}, \Delta_{\mathcal{B}}, q_{0\mathcal{B}}, F_{\mathcal{B}})$, with $\Sigma_{\mathcal{A}}, \Sigma_{\mathcal{B}}$ and Σ_S disjoint alphabets and $C_{\mathcal{A}}$ and $C_{\mathcal{B}}$ disjoint clock sets[1], is the TA

$$\mathcal{A} \otimes_{\Sigma_S} \mathcal{B} = \Big(Q_{\mathcal{A}} \times Q_{\mathcal{B}}, \Sigma_{\mathcal{A}} \cup \Sigma_{\mathcal{B}} \cup \Sigma_S, C_{\mathcal{A}} \cup C_{\mathcal{B}}, \dot{\Delta}_{\mathcal{A}} \cup \Delta_S \cup \dot{\Delta}_{\mathcal{B}}, (q_{0\mathcal{A}}, q_{0\mathcal{B}}), F \Big),$$

$$\begin{aligned}
\text{where } \Delta_S &= \{ ((q_{\mathcal{A}}, q_{\mathcal{B}}), s, \mathfrak{g}_{\mathcal{A}} \wedge \mathfrak{g}_{\mathcal{B}}, \mathfrak{r}_{\mathcal{A}} \cup \mathfrak{r}_{\mathcal{B}}, (q'_{\mathcal{A}}, q'_{\mathcal{B}})) \\
&\qquad \mid \exists s \in \Sigma_S \wedge (q_{\mathcal{A}}, s, \mathfrak{g}_{\mathcal{A}}, \mathfrak{r}_{\mathcal{A}}, q'_{\mathcal{A}}) \in \Delta_{\mathcal{A}} \wedge (q_{\mathcal{B}}, s, \mathfrak{g}_{\mathcal{B}}, \mathfrak{r}_{\mathcal{B}}, q'_{\mathcal{B}}) \in \Delta_{\mathcal{B}} \}; \\
\dot{\Delta}_{\mathcal{A}} &= \{ ((q_{\mathcal{A}}, q_{\mathcal{B}}), a, \mathfrak{g}_{\mathcal{A}}, \mathfrak{r}_{\mathcal{A}}, (q'_{\mathcal{A}}, q_{\mathcal{B}})) \\
&\qquad \mid a \in \Sigma_{\mathcal{A}} \wedge (q_{\mathcal{A}}, a, \mathfrak{g}_{\mathcal{A}}, \mathfrak{r}_{\mathcal{A}}, q'_{\mathcal{A}}) \in \Delta_{\mathcal{A}} \wedge q_{\mathcal{B}} \in Q_{\mathcal{B}} \}; \\
\dot{\Delta}_{\mathcal{B}} &= \{ ((q_{\mathcal{A}}, q_{\mathcal{B}}), b, \mathfrak{g}_{\mathcal{B}}, \mathfrak{r}_{\mathcal{B}}, (q_{\mathcal{A}}, q'_{\mathcal{B}})) \\
&\qquad \mid b \in \Sigma_{\mathcal{B}} \wedge q_{\mathcal{A}} \in Q_{\mathcal{B}} \wedge (q_{\mathcal{B}}, b, \mathfrak{g}_{\mathcal{B}}, \mathfrak{r}_{\mathcal{B}}, q'_{\mathcal{B}}) \in \Delta_{\mathcal{B}} \}; \\
\text{and } F &= \{((q_{\mathcal{A}}, q_{\mathcal{B}}), \mathfrak{g}_{\mathcal{A}} \wedge \mathfrak{g}_{\mathcal{B}}) \mid (q_{\mathcal{A}}, \mathfrak{g}_{\mathcal{A}}) \in F_{\mathcal{A}} \wedge (q_{\mathcal{B}}, \mathfrak{g}_{\mathcal{B}}) \in F_{\mathcal{B}} \}
\end{aligned}$$

We remark that $\mathcal{A}_1 \otimes_\Sigma (\mathcal{A}_2 \otimes_\Sigma \mathcal{A}_3)$ and $(\mathcal{A}_1 \otimes_\Sigma \mathcal{A}_2) \otimes_\Sigma \mathcal{A}_3$ are isomorphic (up to relabelling of the locations), hence, with only a slight abuse of notation, operation $\otimes_\Sigma$ is associative. The same reasoning also holds for commutativity. Thus, for iterating $\otimes_\Sigma$ (with Σ fixed), on $A = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n\}$, a finite set of TA, we define the notation $\bigotimes_{\mathcal{A} \in A}^{\Sigma} \mathcal{A} = \mathcal{A}_1 \otimes_\Sigma \mathcal{A}_2 \otimes_\Sigma \cdots \otimes_\Sigma \mathcal{A}_n$.

2.4 Pre-compact Spaces, and Their ε-entropy and ε-capacity

We recall some concepts (mostly from [12]). Given a metric space (S, d), i.e. a set S with a distance d, a subset $K \subset S$ is called ε-*net* if for any $x \in S$, there exists some $y \in K$ with $d(x, y) \leq \varepsilon$ (i.e. any point can be ε-approximated by an element of K). The ε-*entropy* of S is defined as the logarithm[2] of the size of the smallest ε-net:

$$\mathcal{H}_\varepsilon(S) = \log \min\{\#K | K \text{ an } \varepsilon\text{-net in } S\}.$$

A subset M of S is called ε-*separated*, if all the distances between points in M are $> \varepsilon$. The ε-*capacity* of S is defined as the logarithm of the size of the largest ε-separated set:

$$\mathcal{C}_\varepsilon(S) = \log \max\{\#M | M \text{ an } \varepsilon\text{-separated set in } S\}.$$

The metric space (S, d) is pre-compact if its ε-entropy (or equivalently its ε-capacity) is finite for any $\varepsilon > 0$.

Both $\mathcal{H}$ and $\mathcal{C}$ characterise the quantity of information needed to describe an arbitrary point in S with precision ε, they give respectively upper and lower bound, as shows the following informal reasoning. Indeed, every point $x \in S$ can

[1] Generally, shared clocks could be considered, but they are not needed in this paper.

[2] As usual in information theory, all logarithms are base 2.

be described with precision ε using $\mathcal{H}_\varepsilon$ bits information: it suffices to pick an ε-approximation y in some standard minimal-size ε-net K, and write the number of y in a standard enumeration of K. On the other hand, ε-precise descriptions of points of a (maximal) 2ε-separated set M should be all distinct, which requires at least $\log \#M = \mathcal{C}_{2\varepsilon}(S)$ bits.

The two information characteristics ($\mathcal{H}$ and $\mathcal{C}$) are tightly related [12]:

$$\mathcal{C}_{2\varepsilon} \leq \mathcal{H}_\varepsilon \leq \mathcal{C}_\varepsilon. \tag{1}$$

3 Distance

Given two timed words $u = (a_1, t_1) \dots (a_n, t_n)$ and $v = (b_1, s_1) \dots (b_m, s_m)$, we define

$$\overrightarrow{d}(u, v) = \overleftarrow{d}(v, u) = \max_i \min_j \{|t_i - s_j| : b_j = a_i\};$$

$$d(u, v) = \max(\overrightarrow{d}(u, v), \overleftarrow{d}(u, v)).$$

In words, $\overrightarrow{d}(u, v)$ is small whenever for each event in u, there exists the same event in v, happening at a close date.

As explained in the introduction, this distance formalizes the idea of a slow observer who cannot distinguish events which are too close to each other.

Function d is strongly related to the classical Hausdorff distance (between sets in metric spaces). Indeed, in the case of a one-letter alphabet, $d((a, t_1) \dots (a, t_n), (a, s_1) \dots (a, s_m))$ coincides with Hausdorff distance between two sets of dates $\{t_1, \dots, t_n\}$ and $\{s_1, \dots, s_m\}$.

Let us state basic geometric properties of d, $\overrightarrow{d}$ and $\overleftarrow{d}$. We need a notation: for a timed word v and a date t we denote by $v(t) \subseteq \Sigma$ the set of all letters a such that v contains an event (a, t); also let $\alpha(u) \subseteq \Sigma$ denote the set of all the letters appearing in u.

Proposition 1. – *d is symmetrical, $\overrightarrow{d}$, $\overleftarrow{d}$ are not.*

- *d, $\overrightarrow{d}$, $\overleftarrow{d}$ satisfy the triangular inequality.*
- *$\overrightarrow{d}(u, v) = 0$ whenever for all t it holds that $u(t) \subseteq v(t)$. The criterion for $\overleftarrow{d}$ is symmetrical. Finally, $d(u, v) = 0$ whenever $u(t) = v(t)$ holds for all t.*
- *$\overrightarrow{d}(u, v) = \infty$ whenever $\alpha(u) \nsubseteq \alpha(v)$. The criterion for $\overleftarrow{d}$ is symmetrical. Finally, $d(u, v) = \infty$ whenever $\alpha(u) \neq \alpha(v)$.*

Thus, in fact, d is a pseudo-distance on timed words. Later on, in Sect. 5 we will prove its pre-compactness on timed words of a duration $\leq T$.

We also extend $\overleftarrow{d}$, $\overrightarrow{d}$ and d to distances between an element and a set, and between two sets the usual way. For $\delta \in \{\overleftarrow{d}, \overrightarrow{d}, d\}$, L, L' two timed languages, u a timed word, we define:

- $\delta(u, L) = \min \{\delta(u, v) | v \in L\}$;
- $\delta(L, u) = \min \{\delta(v, u) | v \in L\}$;
- $\delta(L, L') = \min \{\delta(u, v) | u \in L, v \in L'\}$.

4 Quantitative Verification

In this section, we treat the following verification problems:

Quantitative model-checking. Given two timed automata $\mathcal{A}$ and $\mathcal{B}$ we want to compute one of three distances $\overrightarrow{d}(L_\mathcal{A}, L_\mathcal{B})$; $\overleftarrow{d}(L_\mathcal{A}, L_\mathcal{B})$; $d(L_\mathcal{A}, L_\mathcal{B})$. A practical interpretation is as follows: $\mathcal{A}$ represents a timed system; $\mathcal{B}$ recognises the set of bad (erroneous) behaviours. In this case the distances represent the "security margin" between the system and errors. It is similar to robustness from [14] or [10]. The choice of the most appropriate distance for each practical setting is still to be explored.

Quantitative monitoring. Given a timed word w and a timed automaton $\mathcal{B}$ we want to compute one of three distances $\overrightarrow{d}(w, L_\mathcal{B})$; $\overleftarrow{d}(w, L_\mathcal{B})$; $d(w, L_\mathcal{B})$. One practical interpretation is that w is an execution trace (log file, airplane black box record etc.) of a system, (measured with some finite precision $\varkappa$), and $L_\mathcal{B}$ is the set of good (admissible) behaviours. Whenever $d(w, L_\mathcal{B}) > \varkappa$ we can be sure that the system behaviour was erroneous. Symmetrically, if $L_\mathcal{B}$ is the set of bad behaviours, and $d(w, L_\mathcal{B}) > \varkappa$ we can be sure that the system behaviour was correct.

4.1 Reachability Problems

In our complexity analysis we will use a couple of results about reachability on timed graphs: one is well-known, the other less so.

We say that $(G, I, F) \in \texttt{TREACH}$ whenever G is a timed graph, I, F are subsets of its vertices (for technical reasons we suppose them disjoint), and there exists a path in G which starts in I (with all clocks equal to 0) and terminates by a transition to F.

Theorem 1 ([1,9]). *The problem* TREACH *is* PSPACE*-complete.*

A variant of TREACH with bounded length of path is easier. We say that $(G, I, F, b) \in \texttt{TREACH}_\texttt{B}$ whenever G is a timed graph, I, F, disjoint subsets of its vertices and b a natural number in unary representation; and there exists a feasible path of length $\leq b$ in G which starts in I (with all clocks equal to 0) and terminates by a transition to F.

Proposition 2. $\texttt{TREACH}_\texttt{B}$ *is in* NP*.*

Proof. The non-deterministic algorithm will first guess a path (of some length $\ell \leq b$) in the timed graph G from I to F. Feasibility of the path corresponds to existence of a sequence of dates $t_1, \ldots, t_\ell$, satisfying a polynomially sized system of difference constraints, which can be checked polynomially. □

Last, the following result concerns reachability in small time.

Proposition 3. *In a timed graph G, if a state $(q', \mathbf{x}')$ is reachable from a state $(q, \mathbf{x})$, within time $t < g(G)$, by some path π, then it is also reachable within exactly the same time via a path π' of polynomial size. Moreover, such a π' can be chosen such that it contains the same set of transitions as π.*

Proof. Without loss of generality, we assume $g(G) = 1$. Let k be the number of clocks, ℓ the number of locations (vertices) and tr the number of transitions (edges). Consider a path π in G from $(q, \mathbf{x})$ to $(q', \mathbf{x}')$ of duration t. We define the subset of *important* transitions in π which includes, for each clock c (with initial value x)

- the first transition such that $c > x$ (just before taking the transition);
- the same for $c = \lceil x \rceil$ and for $c > \lceil x \rceil$;
- the transition when c is reset for the first or last time.

For each clock there are at most five important transitions (some of them can be absent or coincide), thus altogether there are at most $5k$ of those.

Without changing the important transitions in π, we simplify the periods between those as follows.

During the period between two important transitions the clock vector stays in the same region. Thus if in such a period some location p of G is entered twice, the segment of π of duration τ from p to p can be removed, and replaced by staying in p during the same time τ. After removing all such useless fragments we get a new path π' without repeated locations between two important transitions. Its maximal size is $5k + (5k + 1)\ell$. The duration of π' is t, by construction.

To prove that π' leads again from $(q, \mathbf{x})$ to $(q', \mathbf{x}')$, it suffices to notice that all resets removed from π are not important because they happen between the first and the last reset of the same clock.

If we want π' and π to contain the same set of edges, the simplification should be a bit less aggressive: instead of removing the whole path fragment from p to p we preserve enough loops (at most tr) to visit the same set of transitions. $\square$

4.2 Timed Automata for Neighbourhoods

We present now the key construction for quantitative verification with respect to our distances: for each distance $\delta \in \{\overleftarrow{d}, \overrightarrow{d}, d\}$, for any TA $\mathcal{B}$ and any rational number $\varkappa > 0$, we want to construct an automaton (resp. $\overleftarrow{\mathcal{B}}_\varkappa$, $\overrightarrow{\mathcal{B}}_\varkappa$ and $\mathcal{B}_\varkappa$) that recognizes the $\varkappa$-neighbourhood of $L_\mathcal{B}$, *i.e.* the language

$$\mathcal{N}_\delta(L_\mathcal{B}, \varkappa) = \{w | \exists v \, (v \in L_\mathcal{B} \wedge \delta(w, v) < \varkappa)\} .$$

We build these three automata as products of several components:

- $\mathcal{A}_v$ guesses the word v, check that $v \in L_\mathcal{B}$ and communicates with other components about timed events in v. No wonder, it is very similar to $\mathcal{B}$: it has the same states and clocks. To each transition in $\mathcal{B}$ corresponds a transition in $\mathcal{A}_v$ as presented on Fig. 1.
- For each letter $a \in \Sigma$, automaton $\mathcal{A}_{a\leftarrow}$ checks that every occurrence of a in the guessed v is $\varkappa$-close to its occurrence in the input w, see Fig. 2, left.
- Similarly, for each $a \in \Sigma$, automaton $\mathcal{A}_{a\rightarrow}$ checks that every occurrence of a in the input w is $\varkappa$-close to its occurrence in the guessed v, see Fig. 2, right.

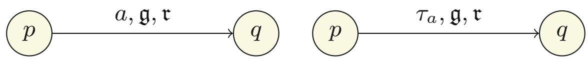

Fig. 1. A transition in $\mathcal{B}$ (left) and its variant in $\mathcal{A}_v$ (right)

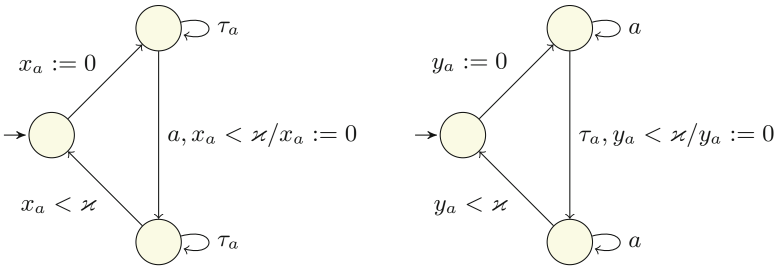

Fig. 2. Timed automata $\mathcal{A}_{a\leftarrow}$ (left) and $\mathcal{A}_{a\rightarrow}$ (right)

Formally, let $\mathcal{A}_{\leftarrow} = \bigotimes^{\emptyset}_{a\in\Sigma} \mathcal{A}_{a\leftarrow}$ and $\mathcal{A}_{\rightarrow} = \bigotimes^{\emptyset}_{a\in\Sigma} \mathcal{A}_{a\rightarrow}$. Then $\mathcal{B}_{\varkappa}$, $\overrightarrow{\mathcal{B}}_{\varkappa}$ and $\overleftarrow{\mathcal{B}}_{\varkappa}$ are defined respectively as the products $\big(\mathcal{A}_{\leftarrow} \otimes_{\{\tau_a|a\in\Sigma\}} \mathcal{A}_v \otimes_{\{\tau_a|a\in\Sigma\}} \mathcal{A}_{\rightarrow}\big)$, $\big(\mathcal{A}_v \otimes_{\{\tau_a|a\in\Sigma\}} \mathcal{A}_{\rightarrow}\big)$ and $\big(\mathcal{A}_{\leftarrow} \otimes_{\{\tau_a|a\in\Sigma\}} \mathcal{A}_v\big)$ where we replace all τ_a by ε.

Proposition 4. *Timed automata $\overrightarrow{\mathcal{B}}_{\varkappa}$; $\overleftarrow{\mathcal{B}}_{\varkappa}$; $\mathcal{B}_{\varkappa}$ recognise respectively $\varkappa$-neighbourhoods of $L_{\mathcal{B}}$ with respect to distances $\overleftarrow{d}$, $\overrightarrow{d}$, d.*

4.3 Quantitative Timed Model-Checking

Proposition 5. *Given timed automata $\mathcal{A}$ and $\mathcal{B}$ and a precision $\varkappa > 0$ (represented as ratio of two integers), deciding whether $d(L_A, L_{\mathcal{B}}) < \varkappa$ is* `PSPACE`*-complete. The same is true for $\overrightarrow{d}$ and $\overleftarrow{d}$.*

Proof. **`PSPACE`-easyness.** Consider first the case of distance d. The inequality $d(L_{\mathcal{A}}, L_{\mathcal{B}}) < \varkappa$ holds iff $L_{\mathcal{A}}$ and $L_{\mathcal{B}_{\varkappa}}$ have a nonempty intersection. We can build a timed automaton C for this intersection using the standard product construction. The size of C is polynomial, it uses ε-transitions and constants proportional to 1 and to $\varkappa$. To get rid of non-integers, we can multiply everything by the denominator of $\varkappa$, and the problem is just reduced to reachability between (initial and final states) in a polynomial-sized timed graph, which is in `PSPACE`.

The cases of $\overrightarrow{d}$ and $\overleftarrow{d}$ are similar, they just use $\overrightarrow{\mathcal{B}}_{\varkappa}$ and $\overleftarrow{\mathcal{B}}_{\varkappa}$ instead of $\mathcal{B}_{\varkappa}$.

`PSPACE`-hardness. We reduce the `TREACH` question for a timed graph G to the question of the form $d(L_{\mathcal{A}}, L_{\mathcal{B}}) < \varkappa$. We choose the trivial event alphabet $\Sigma = \{a\}$, take $\mathcal{A}$ the same as G with all transitions labeled by a; and $\mathcal{B}$ an automaton for the universal timed language over Σ and choose the constant $\varkappa > 0$ arbitrarily. In this case $d(L_{\mathcal{A}}, L_{\mathcal{B}}) = 0 < \varkappa$ whenever the final state is reachable in G, otherwise $d(L_{\mathcal{A}}, L_{\mathcal{B}}) = \infty > \varkappa$. This concludes the reduction and the `PSPACE`-hardness for d; the cases of $\overrightarrow{d}$ and $\overleftarrow{d}$ are similar. $\square$

4.4 Quantitative Timed Monitoring

Proposition 6. *Given a timed word w (with timings represented as rationals), a timed automaton $\mathcal{B}$ and a precision $\varkappa > 0$, deciding whether $d(w, L_\mathcal{B}) < \varkappa$ is* PSPACE*-complete. The same is true for $\overrightarrow{d}$ and $\overleftarrow{d}$.*

Proof. **PSPACE-easyness.** Build a timed automaton $\mathcal{A}$ (with rational constants) recognizing only w, thus $d(w, L_\mathcal{B}) < \varkappa$ iff $d(L_\mathcal{A}, L_\mathcal{B}) < \varkappa$, the latter condition is PSPACE-easy as stated in the previous proposition.

PSPACE-hardness. Again we reduce TREACH. We chose the trivial event alphabet $\Sigma = \{a\}$, take $\mathcal{B}$ the same as G with all transitions labeled by a; and $w = (a, 0)$. The constant $\varkappa$ is chosen very large, an upper bound for the diameter of the region graph of $\mathcal{B}$ ($\varkappa$ can still be written in a polynomial number of bits). Whenever the final state is reachable in G, the language $L_\mathcal{B}$ contains some word v (containing only letters a) of duration smaller than $\varkappa$. In this case $d(w, L_\mathcal{B}) \leq d(w, v) < \varkappa$. If the final state is unreachable in G, then $L_\mathcal{B} = \emptyset$ and hence $d(w, L_\mathcal{B}) = \infty > \varkappa$. This concludes the reduction and the PSPACE-hardness for d; the cases of $\overrightarrow{d}$ and $\overleftarrow{d}$ are similar. □

The case when $\varkappa$ is small is easier. Let us define the granularity $g(w)$ of a timed word $w = (a_1, t_1) \ldots (a_n, t_n)$ as $\min\{t_{s+1} - t_s | t_s < t_{s+1}\}$ (with $t_0 = 0$), i.e. the minimal non-0 interval between events.

Proposition 7. *Given a timed word w (with timings represented as rationals), a timed automaton $\mathcal{B}$ and a positive precision $\varkappa < \min(g(\mathcal{B}), g(w))/2$, deciding whether $d(w, L_\mathcal{B}) < \varkappa$ is* NP*-complete. The same is true for $\overleftarrow{d}$. For $\overrightarrow{d}$, the problem is* PSPACE*-complete even for small $\varkappa$.*

Proof. NP**-easyness for $\overleftarrow{d}$.** We first show (with $\varkappa$ as small as required) that (*) the inequality $\overleftarrow{d}(w, L_\mathcal{B}) < \varkappa$ is equivalent to existence of some $v \in L_\mathcal{B}$ of polynomially bounded length, such that $\overleftarrow{d}(w, v) < \varkappa$.

Let $t_1 < \cdots < t_N$ be distinct dates of events in w. We denote $t_i^- = t_i - \varkappa$ and $t_i^+ = t_i + \varkappa$. Due to the bound on $\varkappa$, the intervals (t_i^-, t_i^+) are of length < 1 and disjoint. The inequality $\overleftarrow{d}(w, L_\mathcal{B}) < \varkappa$ is equivalent to existence of $u \in L_\mathcal{B}$ with $\overleftarrow{d}(w, u) < \varkappa$. In other words, u should satisfy three requirements:

- u is accepted by $\mathcal{B}$;
- for each i, within the interval (t_i^-, t_i^+), the word u only contains letters from $w(t_i)$;
- no events happen in u outside of intervals (t_i^-, t_i^+).

The fragment accepting run of $\mathcal{B}$ on u within the interval (t_i^-, t_i^+) can be considered as a path in timed graph consisting of transitions in $\mathcal{B}$ labeled by letters from $w(t_i)$. This path (and the corresponding fragment of u) can thus be simplified (using Proposition 3) to a polynomially bounded length. Proceeding with such a simplification for every i will transform the totality of u to the required polynomially bounded v. The property (*) is proved.

As in previous easiness proofs, we build a timed automaton $\mathcal{A}$ for $\{w\}$, then another one for $L_\mathcal{A} \cap L_{\overleftarrow{\mathcal{B}}_\varkappa}$, and check existence of a polynomially bounded accepted word v with complexity NP as stated by Proposition 2.

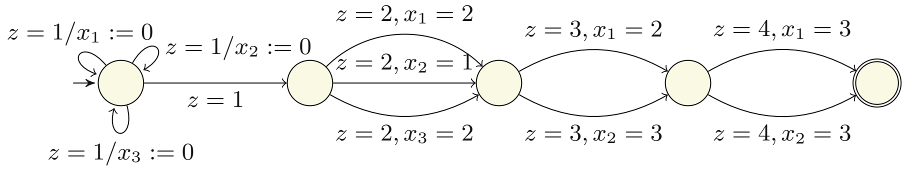

Fig. 3. Timed automaton encoding the 3CNF $(p_1 \vee \bar{p}_2 \vee p_3) \wedge (\bar{p}_1 \vee p_2) \wedge (\bar{p}_1 \vee \bar{p}_3)$

`NP`**-easyness for** d is proved similarly, but the variant of Proposition 2 where the simplified path has the same set of transitions should be used.

`NP`**-hardness for** d **and** $\overleftarrow{d}$**.** We proceed by reduction of `3SAT`. Given a 3CNF $\bigwedge_{i=1}^{k} C_i$ with n boolean variables $p_1, \dots, p_n$, we take $w = (a,1)(a,2)\dots(a,k+1)$. The timed automaton $\mathcal{B}$ over alphabet $\{a\}$ has clocks $x_1, \dots, x_n$ (which will encode boolean variables) and a special clock z which is never reset and ensures that all transitions of $\mathcal{B}$ happen every time unit. A gadget works during the first time unit and gives to each clock x_i a value 0 or 1 (this corresponds to possible boolean values of p_i). Afterwards, there are no more resets, hence $x_i = z-1$ encodes $p_i = 0$ while $x_i = z$ encodes $p_i = 1$. At transition $i+1$ the clause C_i is checked (one of boolean literals should have a required value), see example on Fig. 3.

Whenever the formula is satisfiable, the language of $\mathcal{B}$ contains w, and $d(w, L_{\mathcal{B}}) = 0 < \varkappa$, otherwise $L_{\mathcal{B}} = \emptyset$ and hence $d(w, L_{\mathcal{B}}) = \infty > \varkappa$. This concludes the reduction. The case of $\overleftarrow{d}$ is similar.

`PSPACE`**-hardness for** $\overrightarrow{d}$**.** The hardness argument is quite similar to that of Proposition 6, and also proceeds by reduction of `TREACH` in a timed graph G to the question of the form $\overrightarrow{d}(w, L_{\mathcal{B}}) < \varkappa$ (for any positive $\varkappa$). We chose the trivial event alphabet $\Sigma = \{a\}$, the word $w = (a,1)$. The timed automaton $\mathcal{B}$ has its first transition (labeled by a and resetting all clocks) at time 1, afterwards it follows the graph G with all transitions labeled by a. Whenever the final state is reachable in G, the language $L_{\mathcal{B}}$ contains some word v starting with $(a,1)$. In this case $\overrightarrow{d}(w, L_{\mathcal{B}}) = 0 < \varkappa$. If the final state is unreachable in G, then $L_{\mathcal{B}} = \emptyset$ and hence $\overrightarrow{d}(w, L_{\mathcal{B}}) = \infty > \varkappa$. This concludes the reduction. □

5 Information in Timed Words

In [3–5,7] we have answered the following question:

> Given a timed regular language L, what is the maximal amount of information in $w \in L$, observed with precision ε (as function of ε and **the number of events** n in w)?

We have explored potential applications including information transmission, data compression, and random generation. However, in the timed setting it is more natural and important to consider the quantity of information (or bandwidth) w.r.t. **time elapsed**, i.e. the duration T of w. Unfortunately, it was not possible in previous settings, because considering together timed words of the same duration but different number of events is tricky. We proposed a first solution in [6] based on formal summation of volumes in various dimensions but this turns out to be a bit artificial.

In this section, we show that our new distances on timed words provide a natural framework for a thorough study of quantity of information in timed words and languages w.r.t. **time elapsed**. This quantity can be characterised as ε-capacity (or ε-entropy), as described in Sect. 2.4, of sets of words in L of duration $\leq T$. Two features of our distances are crucial: (1) they can be applied to words with different numbers of events; (2) the relevant sets endowed with these distances are compact.

We cannot yet compute the ε-capacity (or ε-entropy) of an arbitrary timed regular language. Instead, we address two important examples:

- the most important one is the universal timed language U_T^Σ consisting of all timed words on event alphabet Σ (with s elements) of duration $\leq T$. We show that it is indeed compact (and hence all other closed languages of bounded duration are); and compute its ε-entropy. Naturally, since all timed words are considered, the entropy is quite large.
- In many practical situations, the words to consider have some special properties and thus are constrained to belong to some sublanguage, and their bandwidth is lower. The most natural constraint concerns the minimal interval separating the events (or the maximal frequency thereof). For that reason, we consider the timed language F_b^Σ of all the words over an alphabet Σ with the minimal interval b between events (in particular, simultaneous events are forbidden). We come up with tight bounds on its ε-entropy, without surprise it grows much more slowly than for the universal language.

Information in the Universal Timed Language. We consider the universal timed language U_T^Σ and start with a simple construction of a finite subset $M_{T,N}^\Sigma \subset U_T^\Sigma$, explore its net/separated properties and deduce tight bounds on ε-capacity and ε-entropy of U_T^Σ.

The set of timed words $M_{T,N}^\Sigma$ is built as follows. We distribute uniformly N active instants t_k on the interval $[0, T]$, taking $t_k = \frac{(2k-1)T}{2N}$ for $k = 1..N$. Then a timed word w belongs to $M_{T,N}^\Sigma$ whenever

- all events in w happen at active instants t_k only;
- within any fixed active instant t_k, the events in Σ happen in alphabetic order, and each letter in Σ happens at most once (in this instant).

It is easy to see that the cardinality of $M_{T,N}^\Sigma$ is 2^{sN}, indeed for each of N active instants there are 2^s possible choices (corresponding to subsets of Σ).

Proposition 8. *The set $M_{T,N}^\Sigma$ described above is an ε-separated in U_T^Σ whenever $\varepsilon < T/N$. It is an ε-net whenever $\varepsilon \geq T/2N$.*

Given T and ε, we can choose $N = \lceil T/2\varepsilon \rceil$, which yields $N-1 < T/2\varepsilon \leq N$ which implies two bounds:

- $\varepsilon \geq T/2N$, hence $\varepsilon \geq T/2N$ and $M^{\Sigma}_{T,N}$ is an ε-net, and thus $\mathcal{H}_\varepsilon(U^{\Sigma}_T) \leq \log \# M^{\Sigma}_{T,N} = sN$.
- On the other hand, $2\varepsilon < T/(N-1)$, so $M^{\Sigma}_{T,N-1}$ is 2ε-separated, hence $\mathcal{C}_{2\varepsilon}(U^{\Sigma}_T) \geq \log \# M^{\Sigma}_{T,N-1} = s(N-1)$.

Together with (1) this gives $s(N-1) \leq \mathcal{C}_{2\varepsilon}(U^{\Sigma}_T) \leq \mathcal{H}_\varepsilon(U^{\Sigma}_T) \leq sN$.
This implies the following result

Theorem 2. *ε-capacity and ε-entropy of U^{Σ}_T satisfy:*

$$s(\lceil T/2\varepsilon \rceil - 1) \leq \mathcal{H}_\varepsilon(U^{\Sigma}_T) \leq s\lceil T/2\varepsilon \rceil;$$
$$s(\lceil T/\varepsilon \rceil - 1) \leq \mathcal{C}_\varepsilon(U^{\Sigma}_T) \leq s\lceil T/\varepsilon \rceil.$$

Consequently, for $\varepsilon \to 0$ this gives the asymptotical estimates:

$$\mathcal{H}_\varepsilon(U^{\Sigma}_T) \approx sT/2\varepsilon; \quad \mathcal{C}_\varepsilon(U^{\Sigma}_T) \approx sT/\varepsilon.$$

Thus, the maximal bandwidth in timed words on Σ, observed with precision ε, equals $|\Sigma|/\varepsilon$ bits per time unit.

Information in bounded frequency language F^{Σ}_b. We recall that the timed language F^{Σ}_b contains all the words over an alphabet Σ with the minimal interval b between events (in particular, simultaneous events are forbidden). To simplify a bit the reasoning, we also suppose that the first event occurs after b seconds. We let $F^{\Sigma}_{b,T,\varepsilon}$ be the restriction of $F^{\Sigma}_{b,T}$ to timed words with events happening at dates multiple of ε. This set constitutes both an $\varepsilon/2$-net and an ε'-separated subset of $F^{\Sigma}_{b,T}$ for every $0 < \varepsilon' < \varepsilon$:

Lemma 1. *For every $0 < \varepsilon' < \varepsilon$, the set $F^{\Sigma}_{b,T,\varepsilon}$ is an $\varepsilon/2$-net and is ε'-separated.*

Hence, by evaluating the cardinality of $F^{\Sigma}_{b,T,\varepsilon}$, we will learn about $\varepsilon/2$-entropy and ε-entropy of $F^{\Sigma}_{b,T}$.

Lemma 2. *For every $\varepsilon > 0$, it holds that $|F^{\Sigma}_{b,T,\varepsilon}| = \sum_{n=0}^{\lfloor T/b \rfloor} s^n \binom{\lfloor (T-nb)/\varepsilon \rfloor + n}{n}$.*

Proof. We decompose the set $F^{\Sigma}_{b,T,\varepsilon}$ w.r.t. the number n of events per timed word. This number n goes from 0 to $\lfloor T/b \rfloor$ and for each n, there are s^n choices for the events that should be multiplied by the number of possible choices of a sequence of dates belonging to the set $\{b \leq t_1 \leq \cdots \leq t_n \leq T \mid t_{i+1} - t_i \geq b \wedge t_i \in \varepsilon\mathbb{N} \text{ for } i \leq n\}$. The latter set can be mapped to the set $\{0 < k_1 < \cdots < k_n \leq (T-nb)/\varepsilon + n \mid k_i \in \mathbb{N} \text{ for } i \leq n\}$ by the bijection $(t_1, t_2, \ldots, t_n) \mapsto ((t_1 - b)/\varepsilon + 1, (t_2 - 2b)/\varepsilon + 2, \ldots, (t_n - nb)/\varepsilon + n)$. This set has $\binom{\lfloor (T-nb)/\varepsilon \rfloor + n}{n}$ elements. Summing up over all n from 0 to $\lfloor T/b \rfloor$ we get the desired result.

As a consequence of the two previous lemmas we obtain:

Theorem 3. *For every $0 < \varepsilon' < \varepsilon$, the set $F^{\Sigma}_{b,T,\varepsilon}$ is an $\varepsilon/2$-net and is ε'-separated. The information measurements for $F^{\Sigma}_{b,T}$ are tightly linked as follows: for every $0 < \varepsilon' < \varepsilon$,*

$$\mathcal{C}_{\varepsilon}(F^{\Sigma}_{b,T}) \leq \mathcal{H}_{\varepsilon/2}(F^{\Sigma}_{b,T}) \leq \log|F^{\Sigma}_{b,T,\varepsilon}| \leq \mathcal{C}_{\varepsilon'}(F^{\Sigma}_{b,T}).$$

The following asymptotic equality holds when $\varepsilon \to 0$:

$$\log|F^{\Sigma}_{b,T,\varepsilon}| = \mathfrak{n}\log(1/\varepsilon) + \log\left(\frac{(se\left((T/\mathfrak{n}) - b\right))^{\mathfrak{n}}}{\sqrt{2\pi\mathfrak{n}}}\right) + o(1) \text{ with } \mathfrak{n} = \lceil T/b\rceil - 1.$$

Proof. The sequence of inequalities is a consequence of Lemma 1, the definitions of entropy and capacity and the classical inequalities (1). To find the asymptotic expansion of $\log|F^{\Sigma}_{b,T,\varepsilon}|$ up to $o(1)$ we start from Lemma 2.

For $n \leq \lceil T/b\rceil - 1 < T/b$ we have

$$\binom{\lfloor (T-nb)/\varepsilon\rfloor + n}{n} \sim \left(\lfloor (T-nb)/\varepsilon\rfloor + n\right)^n /n! \sim (T-nb)^n/(n!\varepsilon^n).$$

When $n = T/b$, the term is equal to $s^n = O(1)$ which is negligible compared to the other terms.

Thus

$$|F^{\Sigma}_{b,T,\varepsilon}| \sim \sum_{n=0}^{\lceil T/b\rceil -1} s^n \left(\frac{T-nb}{\varepsilon}\right)^n /n!,$$

and this polynomial in $1/\varepsilon$ is equivalent to its last term when $\varepsilon \to 0$:

$$|F^{\Sigma}_{b,T,\varepsilon}| \sim s^{\mathfrak{n}}\left(\frac{T-\mathfrak{n}b}{\varepsilon}\right)^{\mathfrak{n}} /\mathfrak{n}! \text{ with } \mathfrak{n} = \lceil T/b\rceil - 1.$$

Using Stirling formula $\mathfrak{n}! \sim (\mathfrak{n}/e)^{\mathfrak{n}}\sqrt{2\pi\mathfrak{n}}$ we obtain:

$$|F^{\Sigma}_{b,T,\varepsilon}| \sim \frac{(se\left((T/\mathfrak{n}) - b\right))^{\mathfrak{n}}}{\sqrt{2\pi\mathfrak{n}}}\left(\frac{1}{\varepsilon}\right)^{\mathfrak{n}} \text{ with } \mathfrak{n} = \lceil T/b\rceil - 1.$$

Taking logarithms gives the desired asymptotic expansion. □

Note that the second term of the asymptotic estimate is not bounded when T is allowed to vary and to approach a multiple of b from below. For this reason, below we also provide hard bounds for $\log|F^{\Sigma}_{b,T,\varepsilon}|$ that are not as tight w.r.t. variations of parameter ε but behave better w.r.t. parameter T.

Proposition 9. *For every $\varepsilon < 1/2b$ and[3] $T \geq b$, the following inequalities hold:*

$$\left(\left\lfloor\frac{T}{b}\right\rfloor - 1\right)\log\left\lfloor\frac{1}{\varepsilon}\right\rfloor - \left\lfloor\frac{T}{b}\right\rfloor\log\left\lfloor\frac{T}{b}\right\rfloor \leq \log|F^{\Sigma}_{b,T,\varepsilon}| \leq \left\lfloor\frac{T}{b}\right\rfloor\log\frac{1}{\varepsilon} + \left\lfloor\frac{T}{b}\right\rfloor\log 6bes.$$

[3] when $T < b$, the set of interest $F^{\Sigma}_{b,T}$ is empty.

Proof. First, let us state an equality about binomial coefficients: it holds that

$$\sum_{n=0}^{N} \binom{A+n}{n} = \binom{A+N+1}{N}, \tag{2}$$

where A, N and $n < N$ are given natural numbers. The above is true because

$$\sum_{n=0}^{N} \binom{A+n}{n} = \sum_{n=0}^{N} \binom{A+n}{A} = \sum_{n=A}^{A+N} \binom{n}{A} = \binom{A+N+1}{A+1} = \binom{A+N+1}{N},$$

where the third equality is known as the Hockey-Stick identity.

Now we prove the upper-bound using Lemma 2. We first treat the case where $\lfloor T/b \rfloor = 1$, that is $b \leq T < 2b$. In this case, Lemma 2 gives

$$|F^{\Sigma}_{b,T,\varepsilon}| = s\binom{\lfloor (T-b)/\varepsilon \rfloor + 1}{1} = s(\lfloor (T-b)/\varepsilon \rfloor + 1) \leq s(b/\varepsilon + 1) \leq 2sb/\varepsilon.$$

So $\log |F^{\Sigma}_{b,T,\varepsilon}| \leq \log(1/\varepsilon) + \log 2sb \leq \lfloor T/b \rfloor \log(1/\varepsilon) + \lfloor T/b \rfloor \log 6bes$.

Now we treat the case where $T \geq 2b$ (still proving the upper bound).

$$\begin{aligned}&\sum_{n=0}^{\lfloor T/b \rfloor} s^n \binom{\lfloor (T-nb)/\varepsilon \rfloor + n}{n} \\ &\qquad\qquad \leq s^{\lfloor T/b \rfloor} \sum_{n=0}^{\lfloor T/b \rfloor} \binom{\lfloor T/\varepsilon \rfloor + n}{n} = s^{\lfloor T/b \rfloor} \binom{\lfloor T/\varepsilon \rfloor + \lfloor T/b \rfloor + 1}{\lfloor T/b \rfloor},\end{aligned}$$

where the last equality is given by (2). By using the inequality $\binom{N}{m} \leq N^m/m!$, we obtain the following upper bound:

$$\binom{\lfloor T/\varepsilon \rfloor + \lfloor T/b \rfloor + 1}{\lfloor T/b \rfloor} \leq (\lfloor T/\varepsilon \rfloor + \lfloor T/b \rfloor + 1)^{\lfloor T/b \rfloor} / \lfloor T/b \rfloor! \leq \lfloor 3T/\varepsilon \rfloor^{\lfloor T/b \rfloor} / \lfloor T/b \rfloor!,$$

where the latter inequality holds because by assumption $\varepsilon \leq 1/2b \leq 1$, and so $\lfloor 3T/\varepsilon \rfloor \geq 1 \geq \lfloor T/b \rfloor$.

We now use a formula due to Robbins [13]: $N! \geq N^N e^{-N} \sqrt{2\pi N} e^{1/(12N+1)}$ for every $N > 0$. We instantiate this formula for $N = \lfloor T/b \rfloor$ and take its log. We obtain the following upper bound: $-\log(\lfloor T/b \rfloor!) \leq -\lfloor T/b \rfloor \log \lfloor T/b \rfloor + \lfloor T/b \rfloor \log e$ and deduce $\log |F^{\Sigma}_{b,T,\varepsilon}| \leq \lfloor T/b \rfloor \log (\lfloor 3T/\varepsilon \rfloor / \lfloor T/b \rfloor) + \lfloor T/b \rfloor \log e + \lfloor T/b \rfloor \log s$.

Note that $1/\lfloor T/b \rfloor \leq 1/(T/b - 1) = (1/T)/(1/b - 1/T) \leq (1/T)/(1/b - 1/2b) = (1/T)/(1/2b)$, where the second inequality holds due to the assumption $T \geq 2b$ which is equivalent to $1/b - 1/T \geq 1/2b$. Hence $\log (\lfloor 3T/\varepsilon \rfloor / \lfloor T/b \rfloor) \leq \log(1/\varepsilon) + \log 6b$. Summing up we get the desired inequality:

$$\log |F^{\Sigma}_{b,T,\varepsilon}| \leq \lfloor T/b \rfloor \log(1/\varepsilon) + \lfloor T/b \rfloor \log 6bes.$$

Now we prove the left-most inequality:

$$|F^{\Sigma}_{b,T,\varepsilon}| = \sum_{n=0}^{\lfloor T/b \rfloor} s^n \binom{\lfloor (T-nb)/\varepsilon \rfloor + n}{n} \geq \sum_{n=0}^{\lfloor T/b \rfloor - 1} \binom{\lfloor 1/\varepsilon \rfloor + n}{n}.$$

We use again (2) and now the fact that $\binom{N}{m} \geq (N-m)^m/m!$

$$|F^{\Sigma}_{b,T,\varepsilon}| = \sum_{n=0}^{\lfloor T/b \rfloor -1} \binom{\lfloor 1/\varepsilon \rfloor + n}{n} = \binom{\lfloor 1/\varepsilon \rfloor + \lfloor T/b \rfloor}{\lfloor T/b \rfloor - 1} \geq \lfloor 1/\varepsilon \rfloor^{\lfloor T/b \rfloor -1}/(\lfloor T/b \rfloor - 1)!$$

Taking the log we obtain the sought inequality. □

6 Conclusion and Further Work

The three contributions of this paper constitute only the beginning of exploration of a new kind of distance on timed words. Below we draw some research perspectives.

Distance definition. We believe that depending on practical setting, variants of our distance could be appropriate. First, a sort of cost matrix allowing replacement of an a by a b at some cost can be allowed (to compare with edit distance from [8]). Second, a dependence structure over events can be introduced, so that the observer cannot notice swapping independent a and b (when they are close in time), but does observe a swap of a and c.
This work can be seen as a step towards resolution of Open question 5 in the research program of [2], we refer the reader to that work for a general discussion.

Quantitative verification. To make practical quantitative model-checking and monitoring sketched in Sect. 4, methodology, practical algorithms and tools should be developed.

Information in timed languages. The approach introduced opens the way to a thorough study of quantity of information in timed words and languages w.r.t. time elapsed; extending the analysis of Sect. 5 to all timed regular languages is the first challenge. The practical applications to timed data transmission should also be explored.

Acknowledgements. The authors thank James Worrell and François Laroussinie for their valuable advice on complexity analysis.

References

1. Alur, R., Dill, D.L.: A theory of timed automata. Theor. Comput. Sci. **126**, 183–235 (1994). https://doi.org/10.1016/0304-3975(94)90010-8
2. Asarin, E.: Challenges in timed languages: from applied theory to basic theory (column: concurrency). Bull. EATCS **83**, 106–120 (2004). https://eatcs.org/images/images/bulletin/beatcs83.pdf
3. Asarin, E., Basset, N., Béal, M.P., Degorre, A., Perrin, D.: Toward a timed theory of channel coding. In: Jurdziński, M., Nic̆ković, D. (eds.) FORMATS 2012. LNCS, vol. 7595, pp. 27–42. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33365-1_4

4. Asarin, E., Basset, N., Degorre, A.: Entropy of regular timed languages. Inf. Comput. **241**, 142–176 (2015). https://doi.org/10.1016/j.ic.2015.03.003
5. Asarin, E., Degorre, A.: Volume and entropy of regular timed languages: discretization approach. In: Bravetti, M., Zavattaro, G. (eds.) CONCUR 2009. LNCS, vol. 5710, pp. 69–83. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04081-8_6
6. Asarin, E., Degorre, A.: Two size measures for timed languages. In: Lodaya, K., Mahajan, M. (eds.) Proceedings of FSTTCS. LIPIcs, vol. 8, pp. 376–387 (2010). https://doi.org/10.4230/LIPIcs.FSTTCS.2010.376
7. Basset, N.: Timed symbolic dynamics. In: Sankaranarayanan, S., Vicario, E. (eds.) FORMATS 2015. LNCS, vol. 9268, pp. 44–59. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22975-1_4
8. Chatterjee, K., Ibsen-Jensen, R., Majumdar, R.: Edit distance for timed automata. In: Fränzle, M., Lygeros, J. (eds.) Proceedings of HSCC, pp. 303–312. ACM (2014). https://doi.org/10.1145/2562059.2562141
9. Courcoubetis, C., Yannakakis, M.: Minimum and maximum delay problems in real-time systems. Form. Method. Syst. Des. **1**(4), 385–415 (1992). https://doi.org/10.1007/BF00709157
10. Donzé, A., Maler, O.: Robust satisfaction of temporal logic over real-valued signals. In: Chatterjee, K., Henzinger, T.A. (eds.) FORMATS 2010. LNCS, vol. 6246, pp. 92–106. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15297-9_9
11. Gupta, V., Henzinger, T.A., Jagadeesan, R.: Robust timed automata. In: Maler, O. (ed.) HART 1997. LNCS, vol. 1201, pp. 331–345. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0014736
12. Kolmogorov, A., Tikhomirov, V.: ε-entropy and ε-capacity of sets in function spaces. Uspekhi Mat. Nauk **14**(2), 3–86 (1959). Russian, English translation in [15]
13. Robbins, H.: A remark on Stirling's formula. Am. Math. Mon. **62**(1), 26–29 (1955). https://doi.org/10.2307/2308012
14. Sankur, O., Bouyer, P., Markey, N., Reynier, P.-A.: Robust controller synthesis in timed automata. In: D'Argenio, P.R., Melgratti, H. (eds.) CONCUR 2013. LNCS, vol. 8052, pp. 546–560. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40184-8_38
15. Shiryayev, A. (ed.): Selected works of A.N. Kolmogorov, vol. 3. Springer, Dordrecht (1993). https://doi.org/10.1007/978-94-017-2973-4