Asymptotic behaviour in temporal logic *

Eugene Asarin
LIAFA, University Paris Diderot & CNRS, France
www.liafa.univ-paris-diderot.fr/~asarin

Michel Blockelet
LACL, University Paris Est Créteil, France
michel.blockelet@lacl.fr

Aldric Degorre
LIAFA, University Paris Diderot & CNRS, France
www.liafa.univ-paris-diderot.fr/~adegorre

Cătălin Dima
LACL, University Paris Est Créteil, France
lacl.u-pec.fr/dima/

Chunyan Mu †
School of Computer Science, University of Birmingham, UK
c.mu@cs.bham.ac.uk

Abstract

We study the "approximability" of unbounded temporal operators with time-bounded operators, as soon as some time bounds tend to ∞. More specif cally, for formulas in the fragments $\mathsf{PLTL}_{\Diamond}$ and $\mathsf{PLTL}_{\Box}$ of the Parametric Linear Temporal Logic of Alur et al., we provide algorithms for computing the limit entropy as all parameters tend to ∞. As a consequence, we can decide the problem whether the limit entropy of a formula in one of the two fragments coincides with that of its time-unbounded transformation, obtained by replacing each occurrence of a time-bounded operator into its time-unbounded version. The algorithms proceed by translation of the two fragments of PLTL into two classes of discrete-time timed automata and analysis of their strongly-connected components.

Categories and Subject Descriptors [*Theory of computation*]: Logic—Logic and verif cation; [*Mathematics of computing*]: Information theory

General Terms Theory, Verif cation

Keywords entropy, temporal logic

1. Introduction

Properties of programs, protocols and other systems are often specif ed using temporal logics, e.g., LTL. Such logics involve in most cases temporal operators without time bounds such as $\Diamond$ which means eventually, some time in the future. In some other cases, time-bounded operators, such as $\Diamond_{[0;10]}\varphi$ of MITL (which means that the objective formula φ must be satisf ed within 10 time units) are used instead. In this article we explore the following question: how well do "ideal" properties with unbounded temporal operators approximate "practical" properties with time-bounded operators, as soon as some time bounds tend to ∞? The following example illustrates the problematics addressed.

Consider a basic liveness property $\varphi_\infty = \Box\Diamond p$, saying that predicate p is satisf ed by the system inf nitely often. A time-bounded variant of this property is $\varphi_t = \Box\Diamond_{[0;t]}p$ which says that p happens at most every t time units. How close are properties φ_∞ and φ_t when t is large? The answer depends on the proximity criterion. In the set-theoretic setting they are quite different: the sequence $p\bar{p}p\bar{p}^2p\bar{p}^3p\bar{p}^4\dots$ satisf es φ_∞ but violates φ_t for any value of t. From the probabilistic standpoint they are also different: an inf nite random sequence of p and $\bar{p}$ verif es the unbounded liveness φ_∞ but violates its bounded variant φ_t with probability 1.

For these reasons we argue that set-theoretic and probabilistic settings are much too precise, and propose to consider convergence in terms of *entropy* – i.e. information contents of languages in bits per symbol, or, equivalently, their exponential growth rate. Entropy is an important notion in the study of dynamical systems [7], as it is a conjugacy invariant – i.e. systems related by bijective, homeomorphic encodings have the same entropy. The entropy of a f nite automaton can be computed as the log of the greatest positive root of a polynomial with integer coeff cients, and the same property applies to automata over ω-words.

Returning to our example, it turns out that the entropy $\Box\Diamond_{[0;t]}p$ converges to that of $\Box\Diamond p$ as $t \to \infty$; and this is the case also for many other temporal properties. In this paper, for formulas φ_t in two large fragments of the parametric linear temporal logic (PLTL) [2] – namely the "positive" and "negative" fragments, obtained by excluding either bounded until or bounded release from the syntax of PLTL – we provide algorithms for computing the limit entropy as $t \to \infty$; consequently we show that it can be decided whether the entropy of a formula φ_t coincides with that of the formula φ_∞, obtained by replacing all bounded operators in φ_t with their unbounded version.

The algorithms proceed via translation of each formula in the two fragments of PLTL to a parametric discrete-timed Büchi automaton with counters, and then computing, in associated non-parametric f nite automaton, the entropy of the "largest" strongly-connected component which satisf es some specif c properties related with the possibility to either resetting all counters or pumping

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

† This work was performed when the author was at LIAFA, University Paris Diderot & CNRS, France

CSL-LICS 2014, July 14–18, 2014, Vienna, Austria.

http://dx.doi.org/10.1145/2603088.2603158

counters in the original timed automaton. The proofs of the correctness of the algorithms require building "almost injective" mappings between behaviors of length n in the underlying (non-parametric, untimed) Büchi automaton and behaviors of length $n \pm \epsilon$ in the original timed automaton. The technique is similar with coding techniques in symbolic dynamics and is necessary as we cannot provide inclusions between the behaviors of the first type into behaviors of the second.

This paper is structured as follows: in Section 2 we recall necessary notions concerning entropy of languages, as well as the logic PLTL from [2]. In Section 3 we formulate our main problem on limit entropy and its solution (Theorem 2). In the rest of the paper we present its proof: translation from the logic to automata in Section 4 and algorithms for computation of the limit entropy in Section 5. We draw some perspectives in the concluding section.

2. Preliminaries

2.1 Automata and entropy

We assume the reader is familiar with finite and Büchi automata [9]. Given a finite automaton $\mathcal{A}$ we denote its language by $\mathcal{L}(\mathcal{A})$. For a language $\mathcal{L} \subseteq \Sigma^*$ or $\mathcal{L} \subseteq \Sigma^\omega$, we define $\mathcal{L}_n = \mathcal{L} \cap \Sigma^n$, and $\mathtt{pref}(\mathcal{L})$ denotes the set of (finite) prefixes of (ω-) words in $\mathcal{L}$, while $\mathtt{pref}_n(\mathcal{L})$ denotes the set of prefixes of length n of words in $\mathcal{L}$. The suffix of a (finite or ω-) word w starting from position k is denoted w^k.

2.1.1 Entropy of languages

The *entropy* of a language $\mathcal{L} \subseteq \Sigma^*$ ([3]) is defined as:

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

(with the logarithm taken in base 2). Intuitively, the entropy of a language is the amount of information (in bits per symbol) in typical words of the language. An alternative interpretation is that the entropy of a language is the "growth rate" of the language.

Recall that, for a regular language $\mathcal{L} \in \Sigma^*$ accepted by a finite automaton, its entropy can be effectively computed. More precisely, given a deterministic automaton $\mathcal{A}$, we say that $\mathcal{A}$ is *trim* if all its states are reachable from the initial state and co-reachable to a final state. Furthermore, let $M(\mathcal{A})$ denote its *extended* adjacency matrix, $M(\mathcal{A})_{ij} = |\{a \in \Sigma \mid i \xrightarrow{a} j \in \delta_A\}|$, then:

Theorem 1 ([3])**.** *For any deterministic trim automaton $\mathcal{A}$,*

$$\mathcal{H}(\mathcal{L}(\mathcal{A})) = \log \rho(M(\mathcal{A})).$$

Here $\rho(M)$ denotes the spectral radius of M, i.e., the maximal modulus of its eigenvalues. Note that $\mathcal{H}(\mathcal{A})$ can be found as maximum of $\mathcal{H}(S)$ over all strongly connected components S of $\mathcal{A}$.

2.1.2 Entropy of ω-languages

The entropy of an ω-language $\mathcal{L} \subset \Sigma^\omega$ is defined by $\mathcal{H}(\mathcal{L}) = \mathcal{H}(\mathtt{pref}(\mathcal{L}))$ [8]. In particular, if $\mathcal{L} = \Sigma^\omega$ and $|\Sigma| = k$ then $\mathcal{H}(L) = \log k$.

Whenever $\mathcal{L}$ is an ω-regular language recognized by a Büchi automaton $\mathcal{A}$, its entropy can be computed as follows: compute the (finite) automaton $\mathcal{A}^{fin}$ recognizing $\mathtt{pref}(\mathcal{L})$, determinize it and compute the entropy as the logarithm of a spectral radius using Theorem 1.

2.2 PLTL and its fragments

We study here PLTL, the parametric variant of the Linear Temporal Logic from [2], which is an extension of LTL with bounded operators $\mathcal{U}_t$ and $\mathcal{R}_t$. For technical reasons we prefer the formulas in positive normal form, where all the negations are pushed to atomic propositions. This leads to the logic described below.

2.2.1 Syntax

The formulas of PLTL in positive normal form (PNF) over the set of atomic propositions AP with parameters in $\mathbf{t} = \{t_1, \dots t_k\}$. are defined by the grammar:

$$\varphi ::= p \mid \neg p \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \bigcirc \varphi_1 \\ \mid \varphi_1 \mathcal{U} \varphi_2 \mid \varphi_1 \mathcal{R} \varphi_2 \mid \varphi_1 \mathcal{U}_t \varphi_2 \mid \varphi_1 \mathcal{R}_t \varphi_2,$$

where p ranges over AP and $t \in \mathbf{t}$. We also require that every timed parameter $t \in \mathbf{t}$ appears *only once* in each formula.

2.2.2 Semantics

A *valuation* of parameters from $\mathbf{t}$ is just a function $\mathbf{v} : \mathbf{t} \longrightarrow \mathbb{N}$.

The semantics of PLTL is defined in terms of ω-words over 2^{AP}, $w = w_0 w_1 \dots$ and valuation of parameters $\mathbf{v}$ (denoted by $w, \mathbf{v} \models \phi$).

- $w, \mathbf{v} \models p$ for $p \in AP$ iff $p \in w_0$;
- $w, \mathbf{v} \models \neg p$ for $p \in AP$ iff $p \notin w_0$;
- $w, \mathbf{v} \models \varphi_1 \vee \varphi_2$ iff $w, \mathbf{v} \models \varphi_1$ or $w, \mathbf{v} \models \varphi_2$;
- $w, \mathbf{v} \models \varphi_1 \wedge \varphi_2$ iff $w, \mathbf{v} \models \varphi_1$ and $w, \mathbf{v} \models \varphi_2$;
- $w, \mathbf{v} \models \bigcirc \varphi_1$ iff $w^1, \mathbf{v} \models \varphi$;
- $w, \mathbf{v} \models \varphi_1 \mathcal{U} \varphi_2$ iff $\exists i \geq 0$ s.t. $w^i, \mathbf{v} \models \varphi_2$ and $\forall j < i$, we have $w^j, \mathbf{v} \models \varphi_1$;
- $w, \mathbf{v} \models \varphi_1 \mathcal{R} \varphi_2$ iff $\forall j \in \mathbb{N}$, either $w^j, \mathbf{v} \models \varphi_2$ or $\exists i < j$ such that $w^i, \mathbf{v} \models \varphi_1$.
- $w, \mathbf{v} \models \varphi_1 \mathcal{U}_t \varphi_2$ iff $\exists i \leq \mathbf{v}(t)$ with $w^i, \mathbf{v} \models \varphi_2$ and $\forall 0 \leq j < i$, $w^j, \mathbf{v} \models \varphi_1$;
- $w, \mathbf{v} \models \varphi_1 \mathcal{R}_t \varphi_2$ iff $\forall j < \mathbf{v}(t)$, either $w^j, \mathbf{v} \models \varphi_2$ or $\exists i < j$ such that $w^i, \mathbf{v} \models \varphi_1$.

Derived operators are defined as usual: $\Diamond \varphi = \mathtt{true}\, \mathcal{U} \varphi$; $\Box \varphi = \mathtt{false}\, \mathcal{R} \varphi$; $\Diamond_t \varphi = \mathtt{true}\, \mathcal{U}_t \varphi$; $\Box_t \varphi = \mathtt{false}\, \mathcal{R}_t \varphi$. Given a valuation $\mathbf{v}$ of the parameters, we associate an ω-language to any PLTL formula: $[\![\varphi]\!]_{\mathbf{v}} = \{w \in \Sigma^\omega \mid w, \mathbf{v} \models \varphi\}$.

2.2.3 Fragments

The following four fragments of PLTL will play an important role in the rest of the paper: the *positive* fragment $\mathsf{PLTL}_\Diamond$ does not use $\mathcal{R}_t$ operator, the *negative* fragment $\mathsf{PLTL}_\Box$ does not use $\mathcal{U}_t$, while the *non-parametric* fragment (which corresponds to LTL in positive normal form) does not use either of those.

The following monotonicity properties are immediate from the definitions:

Proposition 1. *For any two valuations $\mathbf{v}_1, \mathbf{v}_2 : \mathbf{t} \to \mathbb{N}$ satisfying $\mathbf{v}_1(t) \leq \mathbf{v}_2(t)$ for all $t \in \mathbf{t}$ we have that:*

- *for any formula φ in $\mathsf{PLTL}_\Diamond$ it holds that $[\![\varphi]\!]_{\mathbf{v}_1} \subseteq [\![\varphi]\!]_{\mathbf{v}_2}$;*
- *for any formula φ in $\mathsf{PLTL}_\Box$ it holds that $[\![\varphi]\!]_{\mathbf{v}_1} \supseteq [\![\varphi]\!]_{\mathbf{v}_2}$;*
- *for φ in LTL, it holds that $[\![\varphi]\!]_{\mathbf{v}_1} = [\![\varphi]\!]_{\mathbf{v}_2}$.*

We finally introduce a notation: given a PLTL formula φ, we denote by φ_∞ the formula obtained by replacing each occurrence of the time-bounded operators $\mathcal{U}_t$, resp. $\mathcal{R}_t$, with their unbounded variants $\mathcal{U}$, resp. $\mathcal{R}$. Note that φ_∞ is an LTL formula. We note first a couple of easy properties concerning the relation between φ and φ_∞.

Proposition 2. *For any valuation $\mathbf{v} : \mathbf{t} \to \mathbb{N}$ we have that*

- *for any formula φ in $\mathsf{PLTL}_\Diamond$ it holds that $[\![\varphi]\!]_{\mathbf{v}} \subseteq [\![\varphi_\infty]\!]$;*
- *for any formula φ in $\mathsf{PLTL}_\Box$ it holds that $[\![\varphi]\!]_{\mathbf{v}} \supseteq [\![\varphi_\infty]\!]$*

with $[\![\cdot]\!]$ standing for the usual LTL semantics.

3. The problem and the main result

We are interested in the study of the asymptotic properties of PLTL formulas. More specif cally, we would like to characterize the existence of limits of the type

$$\lim_{\mathbf{v}} [\![\varphi]\!]_{\mathbf{v}} \tag{1}$$

(hereinafter $\lim_{\mathbf{v}}$ means $\lim_{\mathbf{v}\to\infty}$) and their relationship with the previously-def ned formulas φ_∞.

We start by discussing (informally) the limitations of three approaches to def ning and computing such limits: the set-theoretical, the topological and the probabilistic one.

Set-theoretical approach. The f rst discussion about the limitations inherent in computing the limit (1) concerns its possible interpretation in set-theoretical terms.

Note f rst that the monotony properties provided by Proposition 1 imply that, for any $\mathsf{PLTL}_\Diamond$ formula φ_1 and any $\mathsf{PLTL}_\Box$ formula φ_2, the following limits exist:

$$\bigcup_{\mathbf{v}:\mathbf{t}\to\mathbb{N}} [\![\varphi_1]\!]_{\mathbf{v}} \qquad\qquad \bigcap_{\mathbf{v}:\mathbf{t}\to\mathbb{N}} [\![\varphi_2]\!]_{\mathbf{v}} \tag{2}$$

This would suggest interpreting the limit in (1) as either the f rst or the second ω-language in (2), depending on the type of the PLTL formula involved.

Unfortunately, even when they exist, the two limits (2) do not preserve regularity, as shown by the following example:

In all examples of this section we suppose $AP = \{p\}$.

Example 1. *Consider the formula $\varphi = \Box\Diamond_t p$. For any $\mathbf{v} : \mathbf{t} \to \mathbb{N}$, the ω-language $[\![\varphi]\!]_{\mathbf{v}}$ is the set of ω-words over 2^{AP} where consecutive p are separated by no more than $\mathbf{v}(t)$ time units. Thus their limit, $L_{\lim} = \bigcup_t [\![\Box\Diamond_t p]\!]$, is the set of words such that the distance between consecutive p admits an upper bound. It is easy to see that $L_{\lim}$ is not ω-regular.*

On the other hand $\varphi_\infty = \Box\Diamond p$ is the set of words with inf nitely many p, and therefore $L_{\lim} \subsetneq [\![\varphi_\infty]\!]$, which is not as desired.

Topological interpretation. A second interpretation can be given to the limit in (1) in terms of the usual product topology on $\left(2^{AP}\right)^\omega$. For a given set $L \subseteq \left(2^{AP}\right)^\omega$ we denote $cl(L)$ the topological closure of L in the product topology.

Then we may replace the limit in (1) with:

$$cl(\lim_{\mathbf{v}} [\![\varphi]\!]_{\mathbf{v}})$$

whenever the ω-language in the argument exists in the set-theoretical setting, and our quest would be to check whether this quantity equals $cl([\![\varphi_\infty]\!])$.

Example 2. *Consider then the formulas in Example 1. Then clearly:*

$$cl([\![\Box\Diamond_t p]\!]) = cl(\Box\Diamond p) = \left(2^{AP}\right)^\omega$$

But, on the other hand, one also has:

$$cl([\![\Diamond\Box_t p]\!]) = \left(2^{AP}\right)^\omega$$

The two above identities would suggest that both formulas $\Box\Diamond_t p$ and $\Diamond\Box_t p$ are "good" approximations of $\Box\Diamond p$, which we believe not to be very intuitive.

Probabilistic interpretation. The third interpretation that could be proposed for the limit in (1) is a probabilistic one.

Consider a Markov chain $M = (Q, A)$ endowed with a state-labeling $\pi : Q \to 2^{AP}$ such that for any state $q \in Q$ and any $S \subseteq AP$ there exists a state $r \in Q$ such that $\pi(r) = S$ and $A(q, r) \neq 0$ and $A(q, r) \neq 1$. Also denote Pr the probability induced by the Markov chain onto measurable sets in $\left(2^{AP}\right)^\omega$.

But then we have that

$$Pr([\![\Box\Diamond_t p]\!]) = 0 \qquad\qquad Pr([\![\Box\Diamond p]\!]) = 1$$

which indicates that interpreting the limit in (1) as $Pr(\lim_{\mathbf{v}} [\![\varphi]\!]_{\mathbf{v}})$ would not make it equal to $Pr([\![\varphi_\infty]\!])$, as desired.

3.1 Convergence in entropy

We therefore propose to interpret the limits in (1) as *convergence in entropy*, that is, the study of the existence of the limit $\lim_{\mathbf{v}} \mathcal{H}\left([\![\varphi]\!]_{\mathbf{v}}\right)$ with the limit taken for all parameter values going to ∞ (in an arbitrary order) and its relationship with $\mathcal{H}([\![\varphi_\infty]\!])$.

A few facts may help gaining some intuition about such limits. As we will show in Section 5.

$$\lim_{\mathbf{v}} \mathcal{H}([\![\Box\Diamond_t p]\!]_{\mathbf{v}}) = \mathcal{H}([\![\Box\Diamond p]\!]) = 1 \tag{3}$$

Exact calculations of the limit for various parameter values in the left hand side show that this convergence is strict, i.e.

$$H([\![\Box\Diamond_t p]\!]_{\mathbf{v}}) < 1$$

for any v.

On the other hand, the following example shows that an identity like (3) does not always hold even when the set-theoretical limits of the (semantics of the) formulas in the left hand side do exist:

Example 3. *For all valuations $\mathbf{v}(t) \in \mathbb{N}$, $\mathcal{H}([\![\Diamond_t\Box(p)]\!]_{\mathbf{v}}) = 0$, while for the non-parametric variant $\mathcal{H}([\![\Diamond\Box p]\!]) = 1$.*

The interpretation of this mismatch between the two entropies above is the following: for each $\mathbf{v}(t)$, there are exponentially less behaviors in $[\![\Diamond_t\Box(p)]\!]_{\mathbf{v}}$ than in $[\![\Diamond\Box p]\!]$. Therefore, the f rst set of behaviors cannot "approximate" the second for large $\mathbf{v}(t)$.

We are ready to formulate the main problem addressed in this article:

Problem 1. *Given a PLTL formula $\varphi_{\vec{t}}$, decide whether*

$$\lim_{\mathbf{v}} \mathcal{H}([\![\varphi]\!]_{\mathbf{v}}) = \mathcal{H}([\![\varphi_\infty]\!]).$$

More generally, decide whether $\lim_{\mathbf{v}} \mathcal{H}([\![\varphi]\!]_{\mathbf{v}})$ exists and, in this case, compute it.

Finally, note that, due to non-existence of set-theoretical limits, nothing can be said about formulas that are neither in $\mathsf{PLTL}_\Diamond$ nor in $\mathsf{PLTL}_\Box$. E.g., for $\varphi = \Box_{t_1} p \wedge \Diamond_{t_2} \neg p$, for which we have:

- if $\mathbf{v}(t_1) \geq \mathbf{v}(t_2)$, then $[\![\varphi]\!]_{\mathbf{v}} = \emptyset$, thus
$$\lim_{\mathbf{v}(t_2)\to\infty} \lim_{\mathbf{v}(t_1)\to\infty} \mathcal{H}([\![\varphi]\!]_{\mathbf{v}}) = 0;$$
- on the other hand,
$$\lim_{\mathbf{v}(t_1)\to\infty} \lim_{\mathbf{v}(t_2)\to\infty} \mathcal{H}([\![\varphi]\!]_{\mathbf{v}}) = 1;$$

therefore, $\lim_{\mathbf{v}} \mathcal{H}([\![\varphi]\!]_{\mathbf{v}})$ does not exist.

The following theorem provides the solution to the problem for $\mathsf{PLTL}_\Diamond$ and $\mathsf{PLTL}_\Box$ formulas:

Theorem 2 (Main)**.** *Given a formula $\varphi_{\vec{t}}$ in $\mathsf{PLTL}_\Diamond$ or $\mathsf{PLTL}_\Box$, the limit $\lim_{\mathbf{v}} \mathcal{H}([\![\varphi]\!]_{\mathbf{v}})$ always exists and is computable as a logarithm of an algebraic real number. Consequently, it is decidable whether $\lim_{\mathbf{v}} \mathcal{H}([\![\varphi]\!]_{\mathbf{v}}) = \mathcal{H}([\![\varphi_\infty]\!])$.*

The following two sections are devoted to the proof of this theorem. The proof goes in two main steps: f rst we translate PLTL into a variant of discrete-time parametric timed automata. Then we show how to compute the entropy limits for special types of automata, corresponding to $\mathsf{PLTL}_\Diamond$ and $\mathsf{PLTL}_\Box$ formulas.

4. Building automata for PLTL

The algorithmic computation of the entropy of PLTL formulas requires translating PLTL formulas into a discrete-time variant of timed automata. The translation is similar to that of [4], and counters are used to model time-bounded operators. We provide here two slightly asymmetrical variants of the timed automata which correspond to the $\mathsf{PLTL}_\Diamond$ and $\mathsf{PLTL}_\Box$ fragments. The asymmetry amounts to requiring that all counters are tested on each transition in the first variant but not in the second.

4.1 Generalized Büchi Automata with Counters and Parametric Constraints (BüAPC)

The automata to which PLTL is translated are parameterized discrete timed automata with generalized Büchi acceptance conditions. These automata use simple constraints of the form $c \leq t$ or $c \geq t$, where c is a counter and t is an integer parameter. We denote by $G_{\mathtt{Ctr},\mathbf{t}}$, or simply by $G_{\mathtt{Ctr}}$ when the set of parameters is understood from the context, the set of conjunctions of simple constraints over the set of counters $\mathtt{Ctr}$ with formal parameters $\mathbf{t}$. For a given constraint g and parameter valuation $\mathbf{v}$, we denote by $g[\mathbf{v}]$ the result of substituting each formal parameter in g with its actual value given by $\mathbf{v}$. The set of constraints with formal parameters $\mathbf{t}$ substituted with values given by $\mathbf{v}$ is denoted by $G_{\mathtt{Ctr},\mathbf{t},\mathbf{v}}$.

Given an integer valuation of counters $\mathbf{c} : \mathtt{Ctr} \to \mathbb{N}$, we denote by $\mathbf{c}+1$ the valuation defined as $(\mathbf{c}+1)(x) = \mathbf{c}(x)+1$ for all $x \in \mathtt{Ctr}$. Furthermore, given a subset of counters $X \subseteq \mathtt{Ctr}$, we denote by $\mathbf{c}[X := 1]$ the valuation with $\mathbf{c}[X := 1](x) = 1$ if $x \in X$ and $\mathbf{c}[X := 1](x) = \mathbf{c}(x)$ otherwise. We also denote by $\models \subseteq ([\mathtt{Ctr} \to \mathbb{N}] \times G_{\mathtt{Ctr},\mathbf{t},\mathbf{v}})$ the usual satisfiability relation of constraints by integer valuations of counters.

Definition 1. *A Generalised Büchi Automaton with Counters and Parametric Constraints (*BüAPC*) with parameter set* $\mathbf{t}$ *is a tuple* $\mathcal{A} = (Q, \Sigma, \Delta, \mathtt{Ctr}, Q_0, \mathtt{Acc})$*, where*

- Q *is a finite set of states;*
- Σ *is the finite* alphabet *of the automaton;*
- $\mathtt{Ctr}$ *is a finite set of time counters;*
- $Q_0 \subseteq Q$ *is the set of initial states;*
- $\Delta \subseteq Q \times \Sigma \times G_{\mathtt{Ctr}} \times 2^{\mathtt{Ctr}} \times Q$ *is the finite* transition relation*;*
- $\mathtt{Acc} \subseteq 2^{\Delta}$ *is a finite set of accepting conditions (referred to as colours).*

Transitions in Δ *are then tuples* $q \xrightarrow{a,g,X} q'$ *in which* $g \in G_{\mathtt{Ctr}}$ *is the* guard*,* $a \in \Sigma$ *is the* action *and* $X \subseteq \mathtt{Ctr}$ *is the* reset component*. It is assumed that guards in* Δ *use only parameters in* $\mathbf{t}$*.*

As usual, the semantics of a BüAPC is given in terms of the transition system, parameterized by the instantiations of the parameters $\mathbf{t}$ occurring along the transition guards. Formally, given $\mathbf{v} : \mathbf{t} \to \mathbb{N}$, we associate to $\mathcal{A}$ and $\mathbf{v}$ the *counter transition system* $Tr(\mathcal{A}, \mathbf{v}) = (\overline{Q}, \overline{Q}_0, \theta, \overline{\mathtt{Acc}})$ where:

$$\begin{aligned}
\overline{Q} &= \{(q, \mathbf{c}) \mid \mathbf{c} : \mathtt{Ctr} \to \mathbb{N}\} \\
\overline{Q}_0 &= \{(q_0, \mathbf{c_1}) \mid q_0 \in Q_0, \mathbf{c_1} : \mathtt{Ctr} \to \mathbb{N} \\
&\qquad \text{with } \mathbf{c_1}(c) = 1 \text{ for all } c \in \mathtt{Ctr}\} \\
\theta &= \{(q, \mathbf{c}) \xrightarrow{a} (q', \mathbf{c}') \mid \exists q \xrightarrow{a,g,X} q' \\
&\qquad \text{s.t. } \mathbf{c} \models g[\mathbf{v}] \text{ and } \mathbf{c}' = (\mathbf{c}+1)[X := 1]\} \\
\overline{\mathtt{Acc}} &= \Big\{ \{(q, \mathbf{c}) \xrightarrow{a} (q', \mathbf{c}') \mid q \xrightarrow{a} q' \in A\} \cap \theta \mid A \in \mathtt{Acc} \Big\}.
\end{aligned}$$

Note that all counters that are not reset on a transition are incremented on that transition.

A run ρ in $Tr(\mathcal{A}, \mathbf{v})$ is *accepting* if it starts in $\overline{Q}_0$ and, for each $A \in \mathtt{Acc}$, the set of *transitions* occurring infinitely often in ρ has a nonempty intersection with A.

An ω-word $w \in \Sigma^\omega$ is *associated* with a run ρ in $Tr(\mathcal{A}, \mathbf{v})$ if it is the (infinite) concatenation of the labels of the transitions in ρ. The *language* $\mathcal{L}(\mathcal{A}, \mathbf{v})$ of the automaton $\mathcal{A}$ under actual parameter values given by $\mathbf{v}$ is defined as the set of ω-words associated with accepting runs in $Tr(\mathcal{A}, \mathbf{v})$.

When the given BüAPC has all guards equal to $\mathtt{true}$, we denote by $\mathcal{L}(\mathcal{A})$ its language.

Definition 2. *A* BüAPC+ *is a* BüAPC *in which there exists a bijection between the counters and the parameters, such that each transition is labeled with the same guard* $\bigwedge_{x \in \mathtt{Ctr}} x \leq t_x$*.*

A BüAPC− *is a* BüAPC *whose guards are conjunctions of constraints of the form* $x \geq t_x$ *(with a similar bijection), and have the property that, on each transition* $q \xrightarrow{a,g,X} q'$*, all the clocks used in the guard* g *are in* X*.*

Theorem 3. *Given a* PLTL *formula* φ *with parameter set* $\mathbf{t}$ *and using atoms in* AP*, there exists a* BüAPC $\mathcal{A}$ *with the same parameter set* $\mathbf{t}$ *and over the alphabet* 2^{AP} *such that for any valuation* $\mathbf{v}$ *of* $\mathbf{t}$ *we have* $[\![\varphi]\!]_{\mathbf{v}} = \mathcal{L}(\mathcal{A}, \mathbf{v})$*.*

Moreover, this association is constructive and is such that, if φ *is in* $\mathsf{PLTL}_\Diamond$ *then* $\mathcal{A}$ *is a* BüAPC+*, and if* φ *is in* $\mathsf{PLTL}_\Box$ *then* $\mathcal{A}$ *is a* BüAPC−*.*

Proof sketch. The construction of the BüAPC equivalent with the formula φ is an adaptation to the discrete time domain of the construction of a (generalized) timed Büchi automaton equivalent with a MITL formula [1]: each state of the automaton $\mathcal{A}$ is labeled with subformulas of φ representing *obligations* that all runs starting in the state have to accomplish, while transitions must respect the inductive part of the formulas involving the fixpoint operators $\mathcal{U}, \mathcal{R}, \mathcal{U}_t$ and $\mathcal{R}_t$ and the operand of each nexttime formula which labels the source state. A distinct counter is created for each time-bounded operator and guards on counter values are used to model the "timeout" part in the two time-bounded operators. The guards which are created for modelling $\mathcal{U}_t$ obligations involve a constraint of the form $c \leq t$, while the guards created for $\mathcal{R}_t$ obligations involve a constraint of the form $c \geq t$. As a corollary, we will get the third and fourth statement of the theorem. Translation details can be found in Appendix A.1. □

In the sequel, given a BüAPC $\mathcal{A}$, we denote by $\mathcal{A}_\infty$ the automaton obtained from $\mathcal{A}$ by replacing all its guards with $\mathtt{true}$. Note that $\mathcal{A}_\infty$ is a generalized Büchi automaton in which colours label transitions.

5. Asymptotic behaviour of PLTL

We now turn to proving our Main Theorem 2. The proof goes first by translating the given formula φ into a BüAPC $\mathcal{B}$, and then addressing the question of computing $\lim_{\mathbf{v}} \mathcal{H}(\mathcal{L}(\mathcal{B}, \mathbf{v}))$. The algorithms and the proofs are quite different for BüAPC+ and BüAPC−, and we treat each case separately.

5.1 Limit entropy of BüAPC+ automata

5.1.1 Presenting the algorithm

The computation of the entropy of BüAPC+ requires analysing the strongly-connected components of its transition graph. We say that a BüAPC+ is *strongly connected* if its transition graph is strongly connected. A strongly connected BüAPC+ is called *nontrivial* if it contains a cycle; *accepting* if it contains all the colours of $\mathtt{Acc}$; *good* if for each counter there exists some transition resetting it.

Computation of limit entropy of a BüAPC+ is based on the following observations: for $\mathbf{v}$ large enough a run can stay during an unbounded lapse of time only in good SCCs of the automaton. Thus, the entropy is produced in some good SCC, and to be accepted, a run should arrive to a good accepting SCC. However, the run can use the rest of the automaton for reaching the two components mentionned above.

Furthermore, the entropy produced by a good SCC in a BüAPC+ is the same as the entropy produced in the same SCC of the underlying Büchi automaton, obtained by removing all the counter information (resetting and/or constraints). This essential fact, which is proved in Lemma 1, requires building an injection between runs of length n of the untimed good SCC and runs of length "slightly longer than" n in the original automaton.

Data: a BüAPC+ $\mathcal{B}$
Result: $\mathcal{H} = \lim_{\mathbf{v}} \mathcal{H}(\mathcal{L}(\mathcal{B}, \mathbf{v}))$ as log of algebraic number
$\mathsf{SCC} \leftarrow$ `Tarjan`$(\mathcal{B})$;
$\mathsf{SCC}_G \leftarrow$ set of good non-trivial components;
$\mathsf{SCC}_A \leftarrow$ set of accepting non-trivial components;
$\mathcal{B}_1 \leftarrow$`trim`$(\mathcal{B}, Q_0, \mathsf{SCC}_A \cap \mathsf{SCC}_G)$;
$\mathcal{B}_2 \leftarrow$ `finAut(`*restrict*`(`B_1, SCC_G`))`;
return $\mathcal{H}(\mathcal{L}(\mathcal{B}_2))$.

Figure 1: Algorithm for computing the limit entropy of a BüAPC+

The algorithm on Figure 1 for computing the entropy of BüAPC+ is based on the considerations above and uses the following simple subroutines:

- `Tarjan`$(\mathcal{B})$ returns the set of all non-trivial strongly-connected components of the automaton $\mathcal{B}$.
- `trim`$(\mathcal{B}, I, S)$ returns the "useful" part of $\mathcal{B}$, i.e., reachable from I and co-reachable to S.
- `restrict` $(\mathcal{B}, S)$ returns the subautomaton of $\mathcal{B}$ containing only the states belonging to components in S.
- `finAut`$(\mathcal{B})$ removes from $\mathcal{B}$ all the information about counters. In the obtained finite automaton all the states are considered as initial and final.

Let us apply the algorithm to Example 1: $\Box\Diamond_t p$. An automaton (BüAPC+) corresponding to the formula is given in Figure 2, left. Here, the only SCC is good, since it has a cycle that resets the counter without testing it. As a result, both transitions contribute to the entropy, and $\mathcal{B}_2$ (the same figure, right) contains both. Hence, $\lim_{\mathbf{v}} \mathcal{H}([\![\Box\Diamond_t p]\!]_{\mathbf{v}}) = \mathcal{H}(\mathcal{B}_2) = 1$.

We remark, that for a given $\mathbf{v}(t)$, the testing transition cannot be taken as often as the other one, but as $\mathbf{v}(t)$ goes to infinity, the behavior of the automaton on the left becomes closer and closer to that of the automaton on the right (without counters).

5.1.2 Proving correctness

We prove the following:

Proposition 3. *For any* BüAPC+ $\mathcal{B}$*, the limit entropy*

$$\lim_{\mathbf{v}} \mathcal{H}(\mathcal{L}(\mathcal{B}, \mathbf{v}))$$

exists and equals to $\mathcal{H}$ *computed by the algorithm above.*

We will first state that the required limit property holds in every good strongly-connected component.

Lemma 1. *Given* $\mathcal{S}$ *a strongly connected good* BüAPC+*, it holds that* $\lim_{\mathbf{v}}(\mathcal{H}(\mathcal{L}(\mathcal{S}, \mathbf{v}))) = \mathcal{H}(\mathcal{L}(\mathcal{S}_\infty))$.

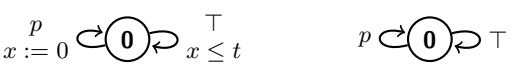

Figure 2: Left: BüAPC+ $\mathcal{B}$ recognizing the language of formula $\Box\Diamond_t p$. Right: finite automaton $\mathcal{B}_2$ computed by the algorithm.

Proof. By construction, every finite run π in $Tr(\mathcal{S}, \mathbf{v})$ is also a run in $\mathcal{S}_\infty$. Thus $\mathcal{L}(\mathcal{S}, \mathbf{v}) \subseteq \mathcal{L}(\mathcal{S}_\infty)$, hence

$$\mathcal{H}(\mathcal{L}(\mathcal{S}, \mathbf{v})) \leq \mathcal{H}(\mathcal{L}(\mathcal{S}_\infty)). \tag{4}$$

For the opposite inequality, we start by fixing, for each state, a cycle resetting all the clocks. Let λ denote the maximum of the lengths of all these cycles. Note that, by assumption of $\mathcal{S}$ being good strongly connected, λ is correctly defined. Also choose some integer T with $T > 2\lambda$ and consider any valuation $\mathbf{v}$ such that $\mathbf{v}(t) \geq T$ for all $t \in \mathbf{t}$.

Let $u \in \mathcal{L}_n(\mathcal{S}_\infty)$, and π the first run (in lexicographic order) in $\mathcal{S}_\infty$ with label u. We build a run π' as follows. We cut the run π into $k-1$ blocks $\pi_1 \cdots \pi_{k-1}$, of length $T - 2\lambda$ and a last block π_k of length $\leq T - 2\lambda$. By hypothesis, for each of the $k-1$ first blocks, there exists a cycle σ_i looping over the final state of π_i, of length $\leq \lambda$ and such that all counters are reset along σ_i.

Let $\pi' = \pi_1\sigma_1\pi_2\sigma_2 \cdots \pi_{k-1}\sigma_{k-1}\pi_k$. The run π' is well-defined by construction, and its length is $\leq n + k\lambda \leq \beta_T n$, with $\beta_T = \frac{T-\lambda}{T-2\lambda}$. Moreover, since all counters are reset at most every T transitions, the counters never exceed value T and therefore π' is also a run in $\mathcal{S}$ with valuation $\mathbf{v}$. Thus the label u' of π' belongs to $\mathcal{L}_{\lfloor \beta_T n \rfloor}(\mathcal{S}, \mathbf{v})$.

The above association of a run π' to each run π defines an *injection* from the runs of length n on $\mathcal{S}_\infty$ to the runs of length $\lfloor \beta_T n \rfloor$ in $\mathcal{S}$ under the parameter valuation given by $\mathbf{v}$. It is important to note that this injection also projects to an injection from $\mathcal{L}_n(\mathcal{S}_\infty)$ to $\mathcal{L}_{\lfloor \beta_T n \rfloor}(\mathcal{S}, \mathbf{v})$, since distinct words in $\mathcal{L}_n(\mathcal{S}_\infty)$ are associated with distinct runs. We then have, for each n, $|\mathcal{L}_n(\mathcal{S}_\infty)| \leq |\mathcal{L}_{\lfloor \beta_T n \rfloor}(\mathcal{S}, \mathbf{v})|$. Thus

$$\frac{1}{\beta_T} \frac{\log |\mathcal{L}_n(\mathcal{S}_\infty)|}{n} \leq \frac{\log |\mathcal{L}_{\lfloor \beta_T n \rfloor}(\mathcal{S}, \mathbf{v})|}{\beta_T n},$$

which implies that

$$\frac{1}{\beta_T} \mathcal{H}(\mathcal{L}(\mathcal{S}_\infty)) \leq \mathcal{H}(\mathcal{L}(\mathcal{S}, \mathbf{v})). \tag{5}$$

Thus according to (4) and (5), we have, for each $\mathbf{v} \geq T$, $\mathcal{H}(\mathcal{L}(\mathcal{S}, \mathbf{v})) \leq \mathcal{H}(\mathcal{L}(\mathcal{S}_\infty)) \leq \beta_T \mathcal{H}(\mathcal{L}(\mathcal{S}, \mathbf{v}))$. As

$$\lim_{T \to \infty} \beta_T = \lim_{T \to \infty} \frac{T - \lambda}{T - 2\lambda} = 1,$$

we can conclude that $\lim_{\mathbf{v}} \mathcal{H}(\mathcal{L}(\mathcal{S}, \mathbf{v})) = \mathcal{H}(\mathcal{L}(\mathcal{S}_\infty))$. □

The next lemma states that $\mathcal{B}$ can only spend a bounded amount of time outside of good components.

Lemma 2. *In a* BüAPC+ $\mathcal{B}$ *with* $\mathbf{v} \leq T$*, any run* $\pi \in Tr(\mathcal{B}, \mathbf{v})$ *which does not visit good components satisfies* $|\pi| \leq \alpha T + e$ *with* α *the number of strongly connected components and* e *the total number of transitions.*

Proof. The run can visit bad SCC and spend at most T time units within each of those. It can also visit some transitions outside SCCs, no more than once each. The estimate is now immediate. □

We can now proceed to the proofs of the upper of and lower bounds giving together Proposition 3, let $\mathcal{H}$ be the result returned

by the algorithm (entropy of $\mathcal{B}_2$ - union of useful good components).

Lemma 3. *The lower bound holds:* $\lim_{\mathbf{v}} \mathcal{H}(\mathcal{L}(\mathcal{B},\mathbf{v})) \geq \mathcal{H}$.

Proof. The proof idea is as follows: according to Lemma 1 some good useful component produces a set of words with entropy tending to $\mathcal{H}$ as $\mathbf{v} \to \infty$, we embed these words into accepting runs of $\mathcal{B}$ and obtain the required lower bound.

The following objects necessarily exist:

- $\mathcal{S}_1$ the useful good SCC of entropy $\mathcal{H}$, and p_1 a state within it;
- π_1 a path from q_0 to p_1 and u_1 its trace, $k_1 = |\pi_1|$;
- σ_1 a cycle within $\mathcal{S}_1$ from p_1 to p_1 resetting all counters, v_1 its trace; $\ell_1 = |\sigma_1|$;
- $\mathcal{S}_2$ an accepting good SCC, and p_2 a state within it;
- σ_2 a cycle within $\mathcal{S}_2$ from p_2 to p_2 resetting all counters and visiting all colours in $\mathtt{Acc}$, v_2 its trace; ℓ_2 its length.
- π_2 a path from p_1 to p_2 and u_2 its trace, k_2 its length.;

Consider a valuation $\mathbf{v} \geq T_1 = \max\{k_1 + \ell_1, \ell_1 + k_2 + \ell_2\}$. Let $w \in \mathcal{L}(\mathcal{S}_1)$ of length n accepted by a run ρ starting in p_1, feasible in $Tr(\mathcal{S}_1, \mathbf{v} - \ell_1)$. Consider an infinite run $\zeta = \pi_1\sigma_1\rho\sigma_1\pi_2\sigma_2^\omega$. It is accepting (since σ_2 contains all the colours); it is easy to see that ζ is feasible for valuation $\mathbf{v}$. Thus, the trace of ζ, i.e., $u_1v_1wv_1u_2v_2^\omega \in \mathcal{L}(\mathcal{B},\mathbf{v})$. We deduce that $u_1v_1w \in \mathtt{pref}(\mathcal{L}(\mathcal{B},\mathbf{v}'))$, thus we have found an injection from $\mathcal{L}(\mathcal{S}_1, \mathbf{v} - \ell_1)$ to $\mathtt{pref}(\mathcal{B},\mathbf{v})$. For a given length, this yields for cardinalities:

$$|\mathcal{L}_n(\mathcal{S}_1, \mathbf{v} - \ell_1)| \leq |\mathtt{pref}_{n+k_1+\ell_1}(\mathcal{L}(\mathcal{B},\mathbf{v}))|$$

(for $\mathbf{v}$ big enough), and for entropies

$$\mathcal{H}(\mathcal{L}(\mathcal{S}_1, \mathbf{v} - \ell_1)) \leq \mathcal{H}(\mathcal{L}(\mathcal{B},\mathbf{v})).$$

Passing to the limit as $\mathbf{v} \to \infty$ and using Lemma 1 we obtain the required estimate $\mathcal{H} \leq \lim_{\mathbf{v}} \mathcal{H}(\mathcal{L}(\mathcal{B},\mathbf{v}))$. □

Lemma 4. *For any* $\mathbf{v}$ *the upper bound holds:* $\mathcal{H}(\mathcal{L}(\mathcal{B},\mathbf{v})) \leq \mathcal{H}$.

Proof. The proof is based on the following observation: any infinite accepting run of $\mathcal{B}$ (1) remains in $\mathcal{B}_1$ and (2) spends almost all its life (except a bounded amount of time) in $\mathcal{B}_2$. Denote $\alpha = |\mathsf{SCC}|; \beta = |\Sigma|$

We need an upper bound for the number of words accepted by good components. We fix $\mathbf{v}$ and some $\eta > 0$, then, by definition of entropy of $\mathcal{B}_{2\infty}$, there exists γ such that for any good component $\mathcal{S}$, for any n the upper bound holds:

$$|\mathcal{L}_n(\mathcal{B}_{2\infty})| \leq \gamma \cdot 2^{n(\mathcal{H}+\eta)}.$$

Consider now any infinite accepting run π of $Tr(\mathcal{B},\mathbf{v})$. First, it should finally arrive in some accepting SCC $\mathcal{S}_f$ and spend the rest of eternity there. Hence this $\mathcal{S}_f$ is accepting and good. Thus any such run stays in $\mathcal{B}_1$ (as defined in the algorithm: reachable from q_0 and co-reachable from accepting good SCCs). The run π spends some time in good SCCs, and some time outside these good SCC. By Lemma 2 the length of every run fragment outside good SCCs is bounded by some constant d. Hence every infinite word ζ in $L(\mathcal{B},\mathbf{v})$ has a form $u_1w_1u_2w_2\dots u_k\xi$ with $|u_i| \leq d$; $w_i \in \mathcal{L}(\mathcal{B}_{2\infty})$ and $\xi \in \mathcal{L}^\omega(\mathcal{B}_{2\infty})$ (clearly, in this decomposition $k \leq |\mathsf{SCC}|$). The same decomposition holds for its prefixes. We obtain that the language of prefixes can be written as follows (with $\mathcal{L}_n$ standing for $\mathcal{L}_n(\mathcal{B}_{2\infty})$):

$$\mathtt{pref}_n(\mathcal{B}_\mathbf{v}) \subset \bigcup_{k\leq\alpha} \bigcup_{\substack{m_1,\dots,m_k\leq d;\\ \sum m_i+\sum n_i=n}} \Sigma^{m_0}\mathcal{L}_{n_1}\Sigma^{m_1}\mathcal{L}_{n_2}\dots\Sigma^{m_k}.$$

The cardinality of each term does not exceed

$$\prod_{i=1}^{k} \beta^{m_i} \cdot 2^{n_i(\mathcal{H}+\eta)} = \beta^{\alpha d} 2^{n(\mathcal{H}+\eta)};$$

the number of terms does not exceed $\alpha d^{\alpha+1}\binom{n}{\alpha} = O(n^\alpha)$, thus we have for the entropy:

$$\mathcal{H}(\mathcal{B}_\mathbf{v}) \leq \lim_{n\to\infty} \frac{\log n^\alpha + O(1) + n(\mathcal{H}+\eta)}{n} = \mathcal{H} + \eta.$$

Since η was arbitrary, the statement of the lemma follows. □

This concludes the proof of Proposition 3.

5.2 Limit entropy of BüAPC− automata

For BüAPC−, unlike the case of BüAPC+, increasing parameter values will *remove* behaviours. While we were previously looking for transitions that will eventually become fireable for big enough parameter values, now the key problem is to find what transitions will still be playing a role for big values.

More precisely, the transitions that can be used for all parameter values are those that can be reached in a way such that all the counters they test (from below) can have arbitrarily big values, i.e. those that can be reached through a "pumping" cycle that does not reset these counters.

Hence, as long as all relevant counter values remain "big", a BüAPC− behaves roughly like a Büchi automaton without counters. This is why we consider a symbolic automaton containing copies of the original automaton for all possible subsets of "big"-valued counters. In this automaton, one can "jump" into an area with a larger subset of big counters by using special transitions we call *slow*, corresponding to the availability of a "pumping" cycle for this subset.

We prove (Lemma 5) that, in the sense of reachability and acceptance, a BüAPC− $\mathcal{B}$ and its symbolic automaton $\mathcal{E}$ "simulate" each other well (accepting runs of one correspond to accepting runs of the other). Then, for each parameter valuation $\mathbf{v}$, we exhibit a subset of the accepting symbolic runs, spending most of their time in the non-slow part of $\mathcal{E}$ (Lemma 6) and whose entropy is close to that of the accepting runs of $Tr(\mathcal{B},\mathbf{v})$ (Lemmata 8,10). We deduce that the limit entropy of $\mathcal{B}$ is that of the non-slow and useful part of $\mathcal{E}$.

5.2.1 The symbolic automaton and the algorithm

The algorithm solving our problem has the form sketched in Figure 3, where the function $\mathtt{symbolic}(\mathcal{B})$ transforms a BüAPC− into the symbolic automaton $\mathcal{E}$ we describe below, the function $\mathtt{remSlow}(\mathcal{E})$ removes its *slow* transitions (see below) and the functions $\mathtt{trim}(\mathcal{B},\mathtt{I},\mathtt{S})$ and $\mathtt{finAut}(\mathcal{B})$ are as in Figure 1.

Data: a BüAPC− $\mathcal{B}$
Result: $\mathcal{H} = \lim_{\mathbf{v}} \mathcal{H}(\mathcal{L}(\mathcal{B},\mathbf{v}))$ as log of algebraic number
$\mathcal{E} \leftarrow \mathtt{symbolic}(\mathcal{B})$;
$\mathcal{E}_1 \leftarrow \mathtt{trim}(\mathcal{E}, Q_0 \times \emptyset, \mathtt{Acc})$;
$\mathcal{E}_2 \leftarrow \mathtt{finAut}(\mathtt{remSlow}(\mathcal{E}_1))$;
return $\mathcal{H}(\mathcal{L}(\mathcal{E}_2))$;

Figure 3: Algortithm for computing the limit entropy of a BüAPC−.

The key construction is that of this counterless symbolic automaton. Instead of having counters, it stores in its states the information about which counters of $\mathcal{B}$ take a big value. The precise construction follows. If $\mathcal{B} = (Q,\Sigma,\Delta,\mathtt{Ctr},Q_0,\mathtt{Acc})$, then the symbolic automaton $\mathcal{E}$ has the form $(Q \times 2^{\mathtt{Ctr}},\Sigma,\Delta_1 \cup \Delta_2, Q_0 \times$

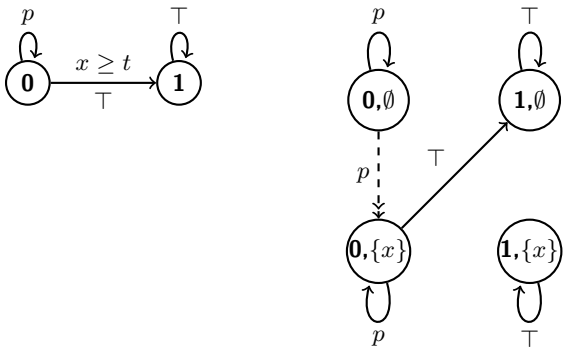

Figure 4: Concrete and symbolic automaton recognizing the language of the $\mathsf{PLTL}_\Box$ formula $\Box_t p$. The dashed arrow represents a slow transition.

$\emptyset, \mathtt{Acc}$). The transition relation of $\mathcal{E}$ is composed of *normal* transitions Δ_1 and *slow* transitions Δ_2. They are obtained as follows: for every transition $\delta = p \xrightarrow{a,g,X} q$ of $\mathcal{B}$ where $g = \bigwedge_{c \in G} c \geq t_c$ with $G \subseteq \mathtt{Ctr}$ and the reset counter set X,

- Δ_1 contains a transition $(p, C) \xrightarrow{a} (q, C \setminus X)$ for every $C \subset \mathtt{Ctr}$ such that $G \subset C$. Informally, if all the clocks tested by the guard are big, the transition can be fired, and all the reset clocks become small after the transition.
- Δ_2 contains a transition $(p, C) \xrightarrow{a} (q, Y)$ whenever it is not in Δ_1 and there exists an elementary cycle π in $\mathcal{B}$ starting from p such that
  - the transitions of π together exactly reset $\mathtt{Ctr} \setminus Y$;
  - transitions of π only test counters in $C \cap Y$ – the cycle can be done once and then iterated many times, thus the counters in Y will be pumped;
  - $G \subset Y$ – after pumping the clocks, the transition δ is fireable.

In both cases, the transitions of $\mathcal{E}$ constructed from δ have the same colours as δ.

To illustrate the algorithm, consider the $\mathsf{PLTL}_\Box$ formula $\Box_t p$. Its BüAPC$-$is presented on the left of Figure 4 and its corresponding symbolic automaton is on the right. After removing slow transitions, we obtain an automaton $\mathcal{E}_2$ with entropy 1. Thus $\lim_{\mathbf{v}} \mathcal{H}(\llbracket \Box_t p \rrbracket_{\mathbf{v}}) = 1$.

Proposition 4. *Let $\mathcal{B}$ be a* BüAPC$-$ *and $\mathcal{E}_2$ the automaton constructed by the algorithm. Then* $\lim_{\mathbf{v}} \mathcal{H}(\mathcal{L}(\mathcal{B}, \mathbf{v})) = \mathcal{H}$.

Here, and in the rest of this section devoted to the proof of Proposition 4, $\mathcal{H}$ stands for $\mathcal{H}(\mathcal{E}_2)$ returned by the algorithm 3.

5.2.2 Simulation properties

The proof of Proposition relies on an *approximate simulation* between the counter automaton $Tr(\mathcal{B}, \mathbf{v})$ for big values of the parameters, and the symbolic automaton $\mathcal{E}$. Formally, for a given $T \in \mathbb{N}$, we may define $\sim_T$ between the state spaces of $Tr(\mathcal{B}, \mathbf{v})$ and $\mathcal{E}$ as follows: $(q, \vec{c}) \sim_T (q, C)$ if $\forall x \in C\ (\vec{c}(x) > T)$, that is the values of all the counters in C are really big.

Lemma 5 (on simulation)**.** *If $T \geq \mathbf{v}$, it holds for $\sim_T$ that*

- *$Tr(\mathcal{B}, \mathbf{v})$ simulates $\mathcal{E}$ restricted to normal transitions: for any run $s_1 \xrightarrow{w} s_2$ in $\mathcal{E}$ using only normal (Δ_1) transitions and any $p_1 \sim_T s_1$ there exists $p_2 \sim_T s_2$ and a run $p_1 \xrightarrow{w} p_2$ on the same word in $Tr(\mathcal{B}, \mathbf{v})$;*
- *$Tr(\mathcal{B}, \mathbf{v})$ weakly simulates $\mathcal{E}$: for any run $s_1 \xrightarrow{w} s_2$ in $\mathcal{E}$ and any $p_1 \sim_T s_1$, there exists $p_2 \sim_T s_2$ and a run $p_1 \xrightarrow{w'} p_2$ (on some other word) in $Tr(\mathcal{B}, \mathbf{v})$;*
- *if some infinite word ζ is accepted from a state s of $\mathcal{E}$, then some (other) infinite word ζ' is accepted in $Tr(\mathcal{B}, \mathbf{v})$ from any $p \sim_T s$.*

Proof. The first statement follows immediately from the definitions. As for the second one, in order to simulate a slow transition $s \xrightarrow{a} s'$ it suffices to iterate in $\mathcal{B}$ the cycle π (from the definition of Δ_2) T times and take the transition $\xrightarrow{a,\cdots}$ afterwards. Every colour of $s \xrightarrow{a} s'$ is also visited by the simulating path of $\mathcal{B}$, which implies the third statement. □

As for simulation in the opposite direction, it is much weaker and can be stated only with respect to paths. We say that $\pi \sim \sigma$ for two paths if they have the following forms: $\pi = (q_0, \vec{x_0}) \xrightarrow{a_1,\cdots} (q_1, \vec{x_1}) \xrightarrow{a_2,\cdots} (q_2, \vec{x_2}) \dots$ in $Tr(\mathcal{B}, \mathbf{v})$ and $\sigma = (q_0, C_0) \xrightarrow{a_1} (q_1, C_1) \xrightarrow{a_2} (q_2, C_2) \dots$ in $\mathcal{E}$ with the same labels and colours.

We call a run of $\mathcal{E}$ *T-low density* if it contains few slow transitions, i.e., on any its interval of length T there are at most $\alpha = |\mathtt{Ctr}|$ such transitions.

Lemma 6. *Let $\mathbf{v} \geq 2|\Delta_1|$, then the following holds for $\sim$*

- *for any path π in $Tr(\mathcal{B}, \mathbf{v})$ starting at Q_0 with counters equal to 0 there exists a path σ in $\mathcal{E}$ such that $\pi \sim \sigma$;*
- *if π is an accepting infinite path, then σ is also accepting;*
- *σ is T-low density for $T = \min \mathbf{v} - |\Delta_1|$.*

Proof. We construct the run σ, using the deterministic algorithm in Figure 5. The index k denotes the last time a slow transition was

Data: bound $T \geq |\Delta_1|$, run π of $Tr(\mathcal{B}, \mathbf{v})$, with $T + |\Delta_1| \leq \mathbf{v}$
Result: run σ of $\mathcal{E}$ such that $\pi \sim \sigma$ and σ is T-low density
$k \leftarrow 0, \sigma \leftarrow \varepsilon$;
while $|\sigma| <$ $|\pi|$ /*No termination if π is infinite. */
do
  $(p, a, g, R, q) \leftarrow \pi[|\sigma|]$;
  if $\exists j \in (k, |\sigma|)$, $\sigma[j] = ((p, C), a, (q, C \setminus R))$ *and* $\exists l \in [j, j + T - 1]$, $g_{\pi[l]}$ *tests* $x \notin C$ **then**
    /*This is a slow transition. */
    $C' \leftarrow \mathtt{Ctr} \setminus reset(\sigma[j, |\sigma| - 1])$;
    $k \leftarrow |\sigma|$;
  else
    /*This is a normal transition. */
    $C' \leftarrow C \setminus R$;
  end
  $\sigma \leftarrow \sigma \cdot ((p, C), a, (q, C'))$;
end

Figure 5: Simulation of a run in the symbolic automaton

taken. The strategy is to take slow transitions as soon as possible, but only when it is actually needed to pass a guard within the next T transitions. We prove that this strategy ensures

(i) that the transition we choose at every iteration exists in $\mathcal{E}$;
(ii) that accepting runs are simulated by accepting runs;
(iii) and that slow transitions are chosen rarely enough.

Item (ii) is immediate, as the construction preserves colouring. The proofs of items (i) and (iii) are detailed in Appendix A.2. □

5.2.3 Correctness proof for the algorithm

To terminate the proof of Proposition 4 we have two inequalities to prove, and they follow from Lemmata 7 and 8 below.

Lemma 7 (lower bound)**.** *For any valuation* $\mathbf{v}$*,* $\mathcal{H}(\mathcal{L}(\mathcal{B},\mathbf{v})) \geq \mathcal{H}$.

Proof. Let M be a strongly connected component of $\mathcal{E}_2$ of maximal entropy and s an arbitrary state therein. Consider the finite automaton $\mathcal{M} = (M, \Sigma, \Delta_1|_M, s, s)$, i.e. the restriction of $\mathcal{E}_2$ to M with some fixed initial and finite state. Thus, by choice of M and, in virtue of usual properties of entropy, $\mathcal{H}(\mathcal{L}(\mathcal{M})) = \mathcal{H}(\mathcal{L}(\mathcal{E}_2)) = \mathcal{H}$. By construction, the automaton $\mathcal{E}_1$ was trimmed (and $\mathcal{E}_2$ has the same states), thus some run leads in $\mathcal{E}$ from $(q_0, \emptyset)$ to s and some infinite word is accepted in $\mathcal{E}$ from s. Given a valuation $\mathbf{v}$, take $T = \max_i v_i$ and apply Lemma 5. This provides us three facts about $\mathcal{B}$ under valuation $\mathbf{v}$:

- for some finite word u and some state $p_1 \sim_T s$ there is a run $q_0 \xrightarrow{u_1} p_1$;
- for any word $w \in \mathcal{L}(M)$ there exists a state $p_2 \sim_T s$ and a run $p_1 \xrightarrow{w} p_2$ (on the same word);
- some infinite word ζ is accepted from any $p_2 \sim_T s$.

Thus any word of the form $uw\zeta$ is accepted by $\mathcal{B}$ with valuation $\mathbf{v}$, i.e., the language inclusion holds: $u\mathcal{L}(\mathcal{M})\zeta \subset \mathcal{L}^\omega(\mathcal{B},\mathbf{v})$, which implies for prefixes: $|\mathtt{pref}_{n+k}\mathcal{L}^\omega(\mathcal{B},\mathbf{v})| \geq |\mathtt{pref}_n\mathcal{L}(\mathcal{M})|$, where k stands for the length of u. The required inequality $\mathcal{H}(\mathcal{L}(\mathcal{B},\mathbf{v})) \geq \mathcal{H}(\mathcal{L}(\mathcal{M}))$ is immediate. □

Lemma 8 (upper bound)**.** *For any* $\eta > 0$ *for* $\mathbf{v}$ *large enough, it holds that* $\mathcal{H}(\mathcal{L}(\mathcal{B},\mathbf{v})) \leq \mathcal{H} + 3\eta$.

Proof. This upper bound is more involved. Let $\alpha = 2|\mathtt{Ctr}|, \beta = |\Sigma|$.

Denote $\mathcal{L}_m(\mathcal{E}_2) = \mathcal{L}_m$. By definition of entropy for $\mathcal{E}_2$, it holds that $\limsup_{m\to\infty} \log|\mathcal{L}_m|/m = \mathcal{H}$, thus for some constant γ for all m it holds that $|\mathcal{L}_m| \leq \gamma \cdot 2^{m(\mathcal{H}+\eta)}$. We fix some $\eta > 0$, and chose T such that

$$\alpha\log(\beta\gamma)/T < \eta \text{ and } \frac{\alpha\log(T\alpha)}{T} + \frac{T-\alpha}{T}\log\frac{T}{T-\alpha} \leq \eta,$$

it is possible since expressions in left-hand sides tend to 0 as $T \to \infty$.

We define a language of finite words LD consisting of traces of all the low density finite runs of $\mathcal{E}_1$ (starting at its initial state).

Lemma 9. *It holds that* $\mathtt{pref}_n(\mathcal{L}(\mathcal{B},\mathbf{v})) \subset LD$.

Proof. Take any finite word $w \in \mathtt{pref}_n(\mathcal{L}_{\mathcal{B},\mathbf{v}})$, it is a prefix of some infinite word ζ, accepted by some path π in $Tr(\mathcal{B},\mathbf{v})$; thus by Lemma 6 there exists an accepting low density path σ in $\mathcal{E}$ such that $\pi \sim \sigma$. Since σ is accepting, it necessarily stays within $\mathcal{E}_1$. Take now its prefix σ_n of length n. By construction it is labelled by the word w. □

It remains to estimate the entropy of LD.

Lemma 10. *The upper bound holds:* $\mathcal{H}(LD) \leq \mathcal{H} + 3\eta$.

Proof. We denote the set of words in LD of length n accepted by runs with k slow transitions by LD^k_n. By definition $LD = \bigcup_n \bigcup_{k\leq\alpha n/T} LD^k_n$. Every word in LD^k_n can be factorized as $w = u_0a_1u_1a_2\cdots u_{k-1}a_ku_k$ with words u_i traces of runs without slow transitions and letters $a_i \in \Sigma$ corresponding to slow transitions. Thus, all the words u_i are recognized by the automaton $\mathcal{E}_2$. Denote $|u_i| = n_i$, thus

$$LD^k_n \subset \bigcup_{n_0+\cdots+n_k=n-k} \mathcal{L}_{n_0}\Sigma\mathcal{L}_{n_1}\Sigma\cdots\mathcal{L}_{n_{k-1}}\Sigma\mathcal{L}_{n_k}.$$

This union has $\binom{n}{k}$ terms; the cardinality of each of those is bounded by

$$|\Sigma|^k \cdot \prod_{i=0}^{k} \gamma 2^{n_i(\mathcal{H}+\eta)} = \gamma^k\beta^k 2^{(n-k)(\mathcal{H}+\eta)} = (\gamma\beta)^k 2^{(n-k)(\mathcal{H}+\eta)} \leq (\gamma\beta)^k 2^{n(\mathcal{H}+\eta)},$$

thus $|LD^k_n| \leq (\gamma\beta)^k 2^{n(\mathcal{H}+\eta)}\binom{n}{k}$ and

$$|LD_n| = \sum_{k\leq\frac{\alpha n}{T}} |LD^k_n| \leq \frac{\alpha n}{T}(\gamma\beta)^{\frac{\alpha n}{T}} 2^{n(\mathcal{H}+\eta)}\binom{n}{\frac{\alpha n}{T}},$$

thus

$$\mathcal{H}(LD) = \limsup_{n\to\infty}\frac{\log|LD_n|}{n} \leq \lim_{n\to\infty}\left(\frac{\log\frac{\alpha n}{T}}{n} + \frac{\alpha\log(\gamma\beta)}{T} + \mathcal{H} + \eta + \frac{\log\binom{n}{\frac{\alpha n}{T}}}{n}\right) \leq \mathcal{H} + 2\eta + \lim_{n\to\infty}\frac{\log\binom{n}{\frac{\alpha n}{T}}}{n}.$$

The last term can be computed using Stirling's formula, which after simplifications yields

$$\lim_{n\to\infty}\frac{\log\binom{n}{\alpha n/T}}{n} = \frac{\alpha\log(T\alpha)}{T} + \frac{T-\alpha}{T}\log\frac{T}{T-\alpha} \leq \eta.$$

The result is now immediate. □

Proposition 4 follows from Lemmata 7–8; and Theorem 2 from Propositions 3–4. □

6. Conclusions

In this article we studied the asymptotic behaviour of temporal formulas when time bounds tend to ∞. The key tool that allowed comparing the languages and finding cases of convergence (such as $\Box\Diamond_t p \to \Box\Diamond p$) and non-convergence (as $\Diamond_t\Box p \nrightarrow \Diamond\Box p$) was the entropy of ω-languages.

We believe that entropy will become a standard size measure in the fast developing domain of quantitative verification [5, 6]. Its strong advantage over probability is that for most relevant properties (e.g., never visit some state) the probability on infinite runs is either 0 or 1. As for the entropy, for safety properties it often takes intermediate values, and for any property it measures its safety component. We will develop this novel quantitative verification paradigm and illustrate it by examples in another paper.

References

[1] R. Alur, T. Feder, and T. A. Henzinger. The benefits of relaxing punctuality. *Journal of the ACM*, 43(1):116–146, 1996.

[2] R. Alur, K. Etessami, S. La Torre, and D. Peled. Parametric temporal logic for "model measuring". *ACM Trans. Comput. Log.*, 2(3):388–407, 2001.

[3] N. Chomsky and G. A. Miller. Finite state languages. *Information and Control*, 1(2):91 – 112, 1958. ISSN 0019-9958. .

[4] J. M. Couvreur. On-the-fly verification of linear temporal logic. In *World Congress on Formal Methods*, pages 253–271, 1999.

[5] T. A. Henzinger and J. Otop. From model checking to model measuring. In P. R. D'Argenio and H. C. Melgratti, editors, *CONCUR*, volume 8052 of *Lecture Notes in Computer Science*, pages 273–287. Springer, 2013. ISBN 978-3-642-40183-1.

[6] M. Kwiatkowska. Quantitative verif cation: models techniques and tools. In *FSE*, pages 449–458. ACM SIGSOFT, 2007.

[7] D. Lind and B. Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, 1995.

[8] L. Staiger. Entropy of f nite-state omega-languages. *Problems of Control and Information Theory*, 14(5):383–392, 1985.

[9] W. Thomas. Automata on inf nite objects. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics (B)*, pages 133–192. 1990.

A. Some technical details

A.1 Translation from PLTL to BüAPC

We start the formalization with some necessary def nitions and notations: given a PLTL formula φ, we let $\mathsf{Sub}(\varphi)$ denote the set of subformulas of φ. Given a set of formulas $F \subseteq \mathsf{Sub}(\varphi)$, we say that F is *consistent* if it satisf es all the following properties:

- if $\psi_1 \vee \psi_2 \in F$, $\psi_1 \mathcal{U} \psi_2 \in F$ or $\psi_1 \mathcal{U}_t \psi_2 \in F$ then $\psi_1 \in F$ or $\psi_2 \in F$;
- if $\psi_1 \wedge \psi_2 \in F$ then $\psi_1 \in F$ and $\psi_2 \in F$;
- if $\psi_1 \mathcal{R} \psi_2 \in F$ or $\psi_1 \mathcal{R}_t \psi_2 \in F$ then $\psi_2 \in F$.

The automaton accepting $[\![\varphi]\!]_{\mathtt{v}}$ is

$$\mathcal{A}_\varphi = (Q_\varphi, 2^{AP}, \Delta_\varphi, \mathtt{Ctr}_\varphi, Q_0^\varphi, \mathtt{Acc}_\varphi)$$

where:

$$\begin{aligned} Q_\varphi &= \big\{F \subseteq \mathsf{Sub}(\varphi) \mid F \text{ is consistent}\big\};\\ Q_0^\varphi &= \big\{F \in Q_\varphi \mid \varphi \in F\big\};\\ \mathtt{Ctr}_\varphi &= \big\{x_{\psi_1 \mathcal{U}_t \psi_2} \mid \psi_1 \mathcal{U}_t \psi_2 \in \mathsf{Sub}(\varphi)\big\}\\ &\quad \cup \big\{x_{\psi_1 \mathcal{R}_t \psi_2} \mid \psi_1 \mathcal{R}_t \psi_2 \in \mathsf{Sub}(\varphi)\big\}. \end{aligned}$$

Furthermore, Δ_φ is composed of transitions $F \xrightarrow{P,g,X} F'$ satisfying the following requirements:

1. $P = F \cap \Pi$.
2. $g = \bigwedge \Gamma$ where $\Gamma \subseteq \big\{(x_{\psi_1 \mathcal{U}_t \psi_2} \leq t) \mid \psi_1 \mathcal{U}_t \psi_2 \in F\big\} \cup \big\{(x_{\psi_1 \mathcal{R}_t \psi_2} \geq t) \mid \psi_1 \mathcal{R}_t \psi_2 \in F\big\}$ bears the following two properties:
   (a) $(x_{\psi_1 \mathcal{U}_t \psi_2} \leq t) \in \Gamma$ for all $\psi_1 \mathcal{U}_t \psi_2 \in \mathsf{Sub}(\varphi)$;
   (b) if $\psi_1 \mathcal{R}_t \psi_2 \in F$ and $\psi_1, \psi_2 \notin F'$ then $(x_{\psi_1 \mathcal{R}_t \psi_2} \geq t) \in \Gamma$.
3. F' is consistent.
4. For each $\bigcirc \psi \in F$, $\psi \in F'$.
5. For each $\psi_1 \mathcal{U} \psi_2 \in F$, F' contains at least one of ψ_1 or ψ_2, and if $\psi_2 \notin F'$ then $\psi_1 \mathcal{U} \psi_2 \in F'$.
6. For each $\psi_1 \mathcal{U}_t \psi_2 \in F$, F' contains at least one of ψ_1 or ψ_2, and if $\psi_2 \notin F'$ then $\psi_1 \mathcal{U}_t \psi_2 \in F'$.
7. For each $\psi_1 \mathcal{R} \psi_2 \in F$, $\psi_2 \in F'$ and if $\psi_1 \notin F'$ then $\psi_1 \mathcal{R} \psi_2 \in F'$.
8. For each $\psi_1 \mathcal{R}_t \psi_2 \in F$, if $(x_{\psi_1 \mathcal{R}_t \psi_2} \geq t) \notin \Gamma$ then $\psi_2 \in F'$, and, furthermore, if also $\psi_1 \notin F'$ then $\psi_1 \mathcal{R}_t \psi_2 \in F'$.
9. $X = \{x_{\psi_1 \mathcal{U}_t \psi_2} \mid \psi_2 \in F' \text{ or } \psi_1 \mathcal{U}_t \psi_2 \notin F\} \cup \{x_{\psi_1 \mathcal{R}_t \psi_2} \mid (x_{\psi_1 \mathcal{R}_t \psi_2} \geq t) \in \Gamma \text{ or } \psi_1 \mathcal{R}_t \psi_2 \notin F\}$.

Finally,

$$\begin{aligned} acc = \big\{A_{\psi_1 \mathcal{U} \psi_2} \mid \psi_1 \mathcal{U} \psi_2 \in \mathsf{Sub}(\varphi)\big\} \cup\\ \big\{A_{\psi_1 \mathcal{U}_t \psi_2} \mid \psi_1 \mathcal{U}_t \psi_2 \in \mathsf{Sub}(\varphi)\big\}, \end{aligned}$$

where

$$\begin{aligned} A_{\psi_1 \mathcal{U} \psi_2} &= \big\{F \xrightarrow{P,g,X} F' \mid \psi_1 \mathcal{U} \psi_2 \notin F \text{ or } \psi_2 \in F'\big\}\\ A_{\psi_1 \mathcal{U}_t \psi_2} &= \big\{F \xrightarrow{P,g,X} F' \mid \psi_1 \mathcal{U}_t \psi_2 \notin F \text{ or } \psi_2 \in F'\big\}. \end{aligned}$$

Note that when starting with a $\mathsf{PLTL}_\Diamond$ formula we obtain a BüAPC+; especially requirement 2(a) ensures that each counter in $\mathcal{A}_\varphi$ is tested on each transition. Also the hypothesis that bounded operators in φ use distinct parameters ensures the existence of the bijection between $\mathtt{Ctr}$ and $\mathbf{t}$. Similarly, starting from a $\mathsf{PLTL}_\Box$ formula we get a BüAPC−.

A.2 Proof of Lemma 6

There remained two items to prove, concerning the algorithm in Fig. 5:

(i) that the transition we choose at every iteration exists in $\mathcal{E}$;

(iii) and that slow transitions are chosen rarely enough.

For existence, assume that at some iteration we already computed σ, pref x of a legal run of $\mathcal{E}$ such that $\pi[0, |\sigma| - 1] \sim \sigma$. We consider the T last transitions of this run, i.e. the subrun $\sigma[|\sigma| - T, |\sigma| - 1]$. Two situations are possible:

- Either there is no transition of Δ_2 in this subrun. Since it is long enough, it must have a cycle of Δ_1, but the algorithm never chose to take a slow transition, which means counters tested by $g_{\pi[|\sigma|]}$ were already big since the beginning and stayed so until the end of the subrun.
- Or there are transitions of Δ_2. We consider the last one of them. Since this transition is slow, as the algorithm goes, it means it was preceded in σ by an elementary cycle of Δ_1. That cycle is a factor of $\sigma[|\sigma| - T - |\Delta_1|, |\sigma| - 1]$, thus does not reset counters tested by $g_{\pi[|\sigma|]}$ and hence these counters become big as a result of the slow transition. After this transition there is no further reset concerning these counters, so they are still big at the end of the subrun.

In both cases, since counters tested at time $|\sigma|$ are big, $\pi[|\sigma|]$ can be simulated, at least by a normal transition. If at this moment, the algorithm chooses a slow transition, it means that it was preceded, in σ, by an elementary cycle of Δ_1, and therefore this slow transition actually exists in Δ_2, by def nition.

Now we prove σ is low density. Consider any interval $[t, t+T-1]$. In $\sigma[t, t+T-1]$, a slow transition $\sigma[r]$ can be chosen only if there is $r' \in [r, r+T-1]$ such that $\pi[r']$ tests a counter c that was not big yet at time $r-1$. A counter c that becomes big as a result of such a slow transition is never reset during interval $[r, r']$, therefore it is never needed to take another slow transition in $[r, t+T-1]$ for the sole reason of making c big again. Consequently, each slow transition of $\sigma[t, t+T-1]$ pumps at least one counter that has not already been pumped by a previous slow transition of that subrun. Therefore, there can be at most $|\mathtt{Ctr}|$ slow transitions during any interval of duration T.