Entropy model checking*(presentation report)

Eugene Asarin†, Michel Blockelet‡, Aldric Degorre†, Cătălin Dima‡, and Chunyan Mu§,†
†LIAFA, University Paris Diderot & CNRS, France
‡LACL, University Paris-Est Créteil Val-de-Marne, France
§School of Computer Science, University of Birmingham, UK

We present an ongoing work on a new, entropy-based, paradigm in quantitative model-checking.

1 Introduction

Classical model-checking answers whether the behaviour(s) of a given system S satisfy a property P. Quantitative model-checking, see e.g. [4, 3], tries instead to give a quantitative measure characterizing to which extent S satisfies P. One of the approaches is to quantify how many behaviours of S satisfy (or violate) P, and the most popular way of such a quantification is to compute probabilities. In many situations probabilistic verification is highly relevant, but it has one important limitation: for many interesting properties, the probability (on infinite behaviours) is either 0 or 1 – and thus no quantitative analysis is possible.

Consider indeed the following simple but very typical example. The system S has 4 states labelled pq, $p\bar{q}$,$\bar{p}q$ and $\bar{p}\bar{q}$ (we denote this set of 4 states by Σ) where p,q are atomic predicates. It can make transitions from each state to each state. The property P is just $\Box q$, that is only two out of four states (pq and $\bar{p}q$, we denote this subset of states by Γ) are allowed. Thus the set of executions is $L_S = \Sigma^\omega$ and the set of good executions, satisfying the property is $L_P = \Gamma^\omega$. For any reasonable probability measure $\mathbb{P}$ on L_s (for example corresponding to a Markov chain with all non-zero transition probabilities), the chance for an infinite run to satisfy P, that is $\mathbb{P}(L_P)$, is 0.

However, in many situations it seems counter-intuitive to say that it is almost impossible to satisfy P. Indeed, it suffices never to go to the right half of the automaton. This can be seen just as a certain restriction of the set of runs, and could be measured. Indeed, let us count the finite behaviours of the system S (i.e. prefixes of words in L_S): for a given length n there are 4^n such behaviours. Among them 2^n satisfy the specification P. Comparing these two asymptotics (2^n out of 4^n) provides a quantitative answer to the question "How much should we restrict behaviours of S to satisfy P".

In this presentation we introduce an alternative approach to the quantitative "model-measuring", generalizing and formalizing the previous example and based on the notion of entropy.

In the presentation we will recall the notion of the entropy of an ω-language, explain how to compute the entropy, describe its application to model-checking, formulate its basic properties in model-checking context, and apply the approach to a simplified version of dining philosophers problem.

2 Entropy

Given a finite automaton $\mathscr{A}$ we denote its language by $\mathscr{L}(\mathscr{A})$. For a language $\mathscr{L} \subseteq \Sigma^*$ or $\mathscr{L} \subseteq \Sigma^\omega$, we define $\mathscr{L}_n = \mathscr{L} \cap \Sigma^n$, and $\mathtt{pref}(\mathscr{L})$ denotes the set of (finite) prefixes of (ω-) words in $\mathscr{L}$, while

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

To appear in EPTCS.

$\texttt{pref}_n(\mathscr{L})$ denotes the set of prefixes of length n of words in $\mathscr{L}$.

The *entropy* of a language (of finite words) $\mathscr{L} \subseteq \Sigma^*$ ([2]) is defined as:

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

(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.

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

Then:

Theorem 1 ([2]) *For any finite deterministic trimmed automaton $\mathscr{A}$,*

$$\mathscr{H}(\mathscr{L}(\mathscr{A})) = \log\rho(M(\mathscr{A})),$$

where $\rho(M)$ stands for the spectral radius the matrix M (i.e. maximal modulus of its eigenvalues).

Note that $\mathscr{H}(\mathscr{A})$ can be found as maximum of $\mathscr{H}(S)$ over all strongly connected components S of $\mathscr{A}$.

The entropy of an ω-language $\mathscr{L} \subset \Sigma^\omega$ is defined by $\mathscr{H}(\mathscr{L}) = \mathscr{H}(\texttt{pref}(\mathscr{L}))$ [5]. In particular, if $\mathscr{L} = \Sigma^\omega$ and $|\Sigma| = k$ then $\mathscr{H}(L) = \log k$.

Whenever $\mathscr{L}$ is an ω-regular language recognized by a Büchi automaton $\mathscr{A}$, its entropy can be computed as follows: compute the (finite) automaton $\mathscr{A}^{fin}$ recognizing $\texttt{pref}(\mathscr{L})$, determinize it and compute the entropy as the logarithm of a spectral radius using Thm 1. It is easy to see that $\mathscr{H}(\mathscr{L}) = H(cl(\mathscr{L}))$, where cl stands for topological closure.

3 Entropy in model-checking context

Consider a system S presented as a Kripke structure and a property P presented as an LTL formula. We naturally associate to them two ω-regular languages: L_S is a (topologically closed) language of all the behaviours of the system ; and L_P of all the infinite words satisfying the property. The entropy-based model-checking consists in comparing a couple of real numbers:

- $\mathscr{H}(\mathscr{L}_S)$ which characterizes the quantity of behaviours of the system.
- $\mathscr{H}(\mathscr{L}_S \cap \mathscr{L}_P)$ which characterizes the quantity of behaviours, of the system, satisfying the property. The defect $\mathscr{H}(\mathscr{L}_S) - \mathscr{H}(\mathscr{L}_S \cap \mathscr{L}_P)$ quantifies how difficult is to steer the system into satisfying the property.
- In case when the defect is zero, the entropy $\mathscr{H}(\mathscr{L}_S \setminus \mathscr{L}_P)$ characterizes the quantity of behaviours of the systems violating the property. We prefer it to be small, at least much smaller than $\mathscr{H}(\mathscr{L}_S)$.

In the example of the introduction, $\mathscr{H}(L_S) = 2$ and $\mathscr{H}(\mathscr{L}_S \cap \mathscr{L}_P) = 1$, the defect is 1. This means that in order to steer the system into the property one should cut in average $1/2$ of its transitions at each execution step.

All the quantities mentioned above can be computed via building a (generalized) Büchi automaton for the language, and applying the algorithm sketched in the previous section. In the presentation we will discuss basic properties of these three quantities, their relation to topology, to probability and to fairness.

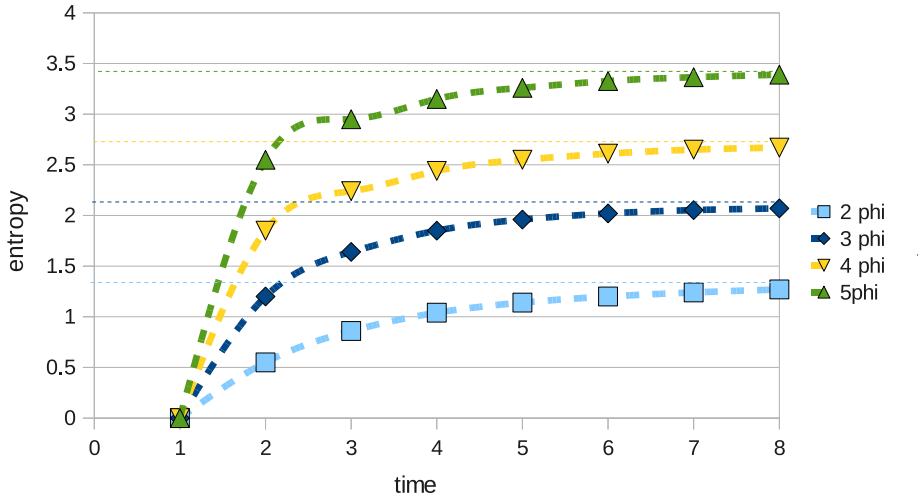
Any ω-regular property P can be represented as a conjunction of P_s which is closed (safety property), and P_l which is dense (livenesse property). The entropy $\mathscr{H}(\mathscr{L}_S \cap \mathscr{L}_P)$ only depends on P_S but not on P_l.

4 Dining philosophers

To illustrate the approach we consider a case study, a simplified version of dining philosophers problem. We fix the number of philosophers and consider several languages:

- $\mathscr{L}_S$ of all the behaviours of the system.
- $\mathscr{L}_S \setminus \mathscr{L}_D$ of all the behaviours which do not enter the deadlock state.
- $\mathscr{L}_S \cap \mathscr{L}_{NS}$ of all the behaviours where no philosopher ever starves.
- $\mathscr{L}_S \cap \mathscr{L}_{Et}$ of all the behaviours where philosopher 1 eats at least every t time units.

The entropies of the three first languages coincide, since (informally) entropy is too rough to analyse liveness. It can be said that few effort is necessary to avoid deadlock and starvation. The things become more interesting with the fourth language when we want to feed every t units. If t is small, this is difficult, when t grows, it becomes easier and easier. As follows from our results presented in [1], for $t \to \infty$ the entropy converges: $\mathscr{H}(\mathscr{L}_S \cap \mathscr{L}_{Et}) \to \mathscr{H}(\mathscr{L}_S)$. The experimental graph below represents this entropy for different numbers of philosophers and values of parameter t,



5 Conclusions

We presented an entropy-based approach to quantitative verification which applies to a class of problems where probability approach fails.

This research is still in its initial phase, and entropy-based methodology is to be developed. It would be interesting to relate entropy to other quantitative approaches. We believe that defect of entropy can be naturally interpreted in Ramadge-Wonham framework (see [6]), as well as in terms of Kolmogorov complexity. In [1] entropy approach allowed us to reason on asymptotics in LTL.

Acknowledgment The authors are thankful to Sylvain Lombardy for a motivating discussion at the origin of this work.

References

[1] E. Asarin, M. Blockelet, C. Dima, A. Degorre & C. Mu (2014): *Asymptotic behaviour in temporal logic*. Submitted.

[2] N. Chomsky & G. A. Miller (1958): *Finite state languages*. Information and Control 1(2), pp. 91 – 112, doi:10.1016/S0019-9958(58)90082-2.

[3] Thomas A. Henzinger & Jan Otop (2013): *From Model Checking to Model Measuring*. In Pedro R. D'Argenio & Hernán C. Melgratti, editors: CONCUR, Lecture Notes in Computer Science 8052, Springer, pp. 273–287. Available at `http://dx.doi.org/10.1007/978-3-642-40184-8_20`.

[4] Marta Kwiatkowska (2007): *Quantitative verification: models techniques and tools*. In: FSE, ACM SIGSOFT, pp. 449–458.

[5] Ludwig Staiger (1985): *Entropy of finite-state omega-languages*. Problems of Control and Information Theory 14(5), pp. 383–392.

[6] J.G. Thistle (1996): *Supervisory control of discrete event systems*. Mathematical and Computer Modelling 23(1112), pp. 25 – 53.