Asymptotic behaviour in temporal logic

Eugene Asarin[1], Michel Bockelet[2], **Aldric Degorre**[1], Cătălin Dima[2] and Chunyan Mu[1]

[1]LIAFA – Université de Paris-Diderot
[2]LACL – Université de Paris-Est Créteil

EQINOCS Meeting
10/01/2014 at UPEM

What?

- Temporal logics are a major specification formalism in verification and synthesis.
- A formula specifies a language, the entropy of which can be studied.
- Here, we study entropy of some temporal logic with parametrized time bounds.

Why?

Why parametrized time bounds:

- Real life appliances may implement time-unbounded properties as time-bounded behaviors.
- Actual observers/monitors do not have inifinite patience.
- Can we still observe the desired behaviors, despite the above, at least for big enough time bounds?

Why?

Why parametrized time bounds:

- Real life appliances may implement time-unbounded properties as time-bounded behaviors.
- Actual observers/monitors do not have inifinite patience.
- Can we still observe the desired behaviors, despite the above, at least for big enough time bounds?

Why study entropy in this context:

As usual: rough assessment of the quality of the approximations made above.
(Probabilities are too precise: a typical safety property has probability 0.)

Reminder: LTL

[Pnueli Focs'77]

Temporal logic over boolean variables $p \in \mathrm{AP}$, with following syntax:

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

(and usual syntactic sugar: $\top, \bot, \implies, \Box, \Diamond, \ldots$)
Models: infinite words in $\left(2^{\mathrm{AP}}\right)^{\omega}$.

Example

A model of $\Box(p \implies \bigcirc q)$:

p	0	1	1	0	0...
q	1	0	1	1	0...

EQINOCS' nails and hammer

... or why this talk is not about LTL(1)

Our problem:

- "How many" behaviors satisfy a formula?
- I.e., for infinite behaviors, how many prefixes?

EQINOCS' nails and hammer

... or why this talk is not about LTL(1)

Our problem:

- "How many" behaviors satisfy a formula?
- I.e., for infinite behaviors, how many prefixes?

Our tool: entropy $\mathcal{H}$. For an ω-language L:

$$\mathcal{H}(L) = \limsup_{n\to\infty} \frac{1}{n} \log \#\mathtt{pref}(L, n)$$

EQINOCS' nails and hammer

... or why this talk is not about LTL(1)

Our problem:

- "How many" behaviors satisfy a formula?
- I.e., for infinite behaviors, how many prefixes?

Our tool: entropy $\mathcal{H}$. For an ω-language L:

$$\mathcal{H}(L) = \limsup_{n\to\infty} \frac{1}{n} \log \#\mathtt{pref}(L, n)$$

Example

- $\mathcal{H}((a+b)^\omega) = \log 2 = 1$;
- $\mathcal{H}(\llbracket \Box\Diamond p \rrbracket) = \log 2^{|\mathrm{AP}|} = |\mathrm{AP}|$ (no constraint most of the time);
- $\mathcal{H}(\llbracket \Diamond\Box p \rrbracket) = |\mathrm{AP}|$ (for any prefix, it is always possible to append p).

Entroy of LTL: either too hard... or too sad

... or why this talk is not about LTL(2)

- Unfortunately, except for a few easy and obvious cases $H([\![\varphi]\!])$ is hard to guess.

Example

One easy case, "liveness" formulas: $H([\![\Diamond\psi]\!]) = |\mathrm{AP}|$, where $[\![\psi]\!] \neq \emptyset$.

- Nonetheless, ω-regular languages $\implies$ $\exists$ translation to (Generalized Büchi) Automata [Couvreur].
- The usual (but sad!) approach $H = \log \rho(M)$ works well (M: adjacency matrix of the determinization of some subautomaton).

PLTL

[Alur, Etessami, LaTorre, Peled ICALP'99]

- PLTL: LTL with parameters.
- 2 new parametrized modalities: $\mathcal{U}_t$ and $\mathcal{R}_t$ (or equivalently $\Box_t$ and $\Diamond_t$).
- Model of a PLTL formula: parameter value + behavior.
- Classical problem: what parameter values make the formula satisfiable?

PLTL

[Alur, Etessami, LaTorre, Peled ICALP'99]

- PLTL: LTL with parameters.
- 2 new parametrized modalities: $\mathcal{U}_t$ and $\mathcal{R}_t$ (or equivalently $\Box_t$ and $\Diamond_t$).
- Model of a PLTL formula: parameter value + behavior.
- Classical problem: what parameter values make the formula satisfiable?

Our problem:

- For a given parameter value, compute $\mathcal{H}$?

PLTL

[Alur, Etessami, LaTorre, Peled ICALP'99]

- PLTL: LTL with parameters.
- 2 new parametrized modalities: $\mathcal{U}_t$ and $\mathcal{R}_t$ (or equivalently $\Box_t$ and $\Diamond_t$).
- Model of a PLTL formula: parameter value + behavior.
- Classical problem: what parameter values make the formula satisfiable?

Our problem:

- ~~For a given parameter value, compute $\mathcal{H}$~~ $\rightarrow$ no! (it's LTL)
- Look at $\mathcal{H}$ when parameter values go to ∞ and compare with LTL $\rightarrow$ yes, let's do this!

Outline

PLTL syntax

A PLTL formula φ in positive normal form is as follows:

$$\begin{aligned}\varphi ::= p \mid \neg p \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 & \qquad \text{propositional logic} \\ \mid \bigcirc \varphi_1 \mid \varphi_1 \mathcal{U} \varphi_2 \mid \varphi_1 \mathcal{R} \varphi_2 & \qquad \text{time modalities} \\ \mid \varphi_1 \mathcal{U}_t \varphi_2 \mid \varphi_1 \mathcal{R}_t \varphi_2 & \qquad \text{parametrized time modalities}\end{aligned}$$

($p \in \mathrm{AP}$: propositional variable; $t \in \mathbf{t}$: formal parameter)

Expected syntatic sugar: $\Box_t \varphi \equiv \bot \mathcal{R}_t \varphi$, $\Diamond_t \varphi \equiv \top \mathcal{U}_t \varphi$.

The following fragments are defined :

- $\mathrm{PLTL}_\Diamond$: PLTL without $\mathcal{R}_t$, “positive fragment”.
- $\mathrm{PLTL}_\Box$: PLTL without $\mathcal{U}_t$, “negative fragment”.

From PLTL, back to LTL

From a PLTL formula φ we derive the following LTL formulas:

- $\varphi[v]$, where $\mathbf{v} \in \mathbb{N}^{\mathbf{t}}$ is a parameter valuation: by substituting $[t \leftarrow \mathbf{v}(t)]$ in every $\mathcal{U}_t$ and $\mathcal{R}_t$ and developping;

Example

$(p\,\mathcal{U}_t q)[t \leftarrow 2] = p\,\mathcal{U}_2 q = q \vee (p \wedge \bigcirc q) \vee (p \wedge \bigcirc(p \wedge \bigcirc q))$

- φ_∞: by replacing each $\mathcal{U}_t$ by $\mathcal{U}$ and $\mathcal{R}_t$ by $\mathcal{R}$ in φ.

Example

$(p\,\mathcal{U}_t q)_\infty = p\,\mathcal{U} q$

PLTL semantics (1)

Models: For a word $w \in \left(2^{\mathsf{AP}}\right)^{\omega}$, a parameter valuation $\mathbf{v} \in \mathbb{N}^{\mathbf{t}}$ and a PLTL formula φ, we say $w, \mathbf{v} \models \varphi$ if and only if $w \models \varphi[v]$.

Example

Two models of $\varphi_1 = (\Box p)\mathcal{R}_t q$:

- $\begin{array}{|c|ccccc} p & 0 & 1 & 1 & 1 & 1\ldots \\ \hline q & 1 & 1 & 0 & 0 & 0\ldots \end{array}$, $[t \leftarrow 3]$
- $\begin{array}{|c|ccccc} p & 1 & 0 & 0 & 0 & 1\ldots \\ \hline q & 1 & 1 & 0 & 0 & 0\ldots \end{array}$, $[t \leftarrow 2]$

PLTL semantics (2)

The language of PLTL formula φ with parameters valuation $\mathbf{v} \in \mathbb{N}^{\mathbf{t}}$ is $[\![\varphi]\!]_{\mathbf{v}} = [\![\varphi[\mathbf{v}]]\!] = \{w \mid w, \mathbf{v} \models \varphi\}$.

Example

Regular expression for $[\![\Box\Diamond_s\Box_t p]\!]_{s\leftarrow 2, t\leftarrow 3}$[1]:

$$\left(\left(\varepsilon + \mathtt{true} + \mathtt{true}^2\right) \cdot \bar{p}^3\right)^\omega$$

[1] Reminder: the alphabet is 2^{AP} = set of all propositional variables valuations. $\bar{p}$ and $\mathtt{true}$ are just convenient notations its subsets.

Limit entropy problem for PLTL

- A natural question: does the following indentity hold?

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

Limit entropy problem for PLTL

- A natural question: does the following indentity hold?

$$\lim_{\mathbf{v}} \mathcal{H}(\llbracket \varphi \rrbracket_{\mathbf{v}}) = \mathcal{H}(\varphi_\infty)$$

- Obviously, not always: consider $\varphi = p\mathcal{U}_t \Box (p \wedge q)$.
  For all $v \in \mathbb{N}$, $\mathcal{H}\left(\llbracket \varphi \rrbracket_{t \leftarrow v}\right) = |AP| - 2$ but $\mathcal{H}(\llbracket \varphi_\infty \rrbracket) = |AP| - 1$.

Limit entropy problem for PLTL

- A natural question: does the following indentity hold?

$$\lim_{\mathbf{v}} \mathcal{H}(\llbracket \varphi \rrbracket_{\mathbf{v}}) = \mathcal{H}(\varphi_\infty)$$

- Obviously, not always: consider $\varphi = p\mathcal{U}_t\Box(p \wedge q)$.
  For all $v \in \mathbb{N}$, $\mathcal{H}\left(\llbracket \varphi \rrbracket_{t \leftarrow v}\right) = |\mathsf{AP}| - 2$ but $\mathcal{H}(\llbracket \varphi_\infty \rrbracket) = |\mathsf{AP}| - 1$.
- Objection: the "true limit" of $p\mathcal{U}_t\Box(p \wedge q)$ is $\Box p$!

Limit entropy problem for PLTL

- A natural question: does the following indentity hold?

$$\lim_{\mathbf{v}} \mathcal{H}(\llbracket\varphi\rrbracket_{\mathbf{v}}) = \mathcal{H}(\varphi_\infty)$$

- Obviously, not always: consider $\varphi = p\mathcal{U}_t\Box(p \wedge q)$.
  For all $v \in \mathbb{N}$, $\mathcal{H}\left(\llbracket\varphi\rrbracket_{t\leftarrow v}\right) = |\mathrm{AP}| - 2$ but $\mathcal{H}(\llbracket\varphi_\infty\rrbracket) = |\mathrm{AP}| - 1$.
- Objection: the "true limit" of $p\mathcal{U}_t\Box(p \wedge q)$ is $\Box p$!
- Then what about $\psi = \Box\Diamond\Box_t p$?
  (limit: irregular language)

Limit entropy problem for PLTL

- A natural question: does the following indentity hold?

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

- Obviously, not always: consider $\varphi = p\mathcal{U}_t\Box(p \wedge q)$.
  For all $v \in \mathbb{N}$, $\mathcal{H}\left([\![\varphi]\!]_{t\leftarrow v}\right) = |\mathrm{AP}| - 2$ but $\mathcal{H}([\![\varphi_\infty]\!]) = |\mathrm{AP}| - 1$.
- Objection: the "true limit" of $p\mathcal{U}_t\Box(p \wedge q)$ is $\Box p$!
- Then what about $\psi = \Box\Diamond\Box_t p$?
  (limit: irregular language)
- Worse: $\Box_s p \wedge \Diamond_t \neg p$ does not converge, even in $\mathcal{H}$.

Our actual result

Theorem (Main)

Given a formula φ in $\mathrm{PLTL}_\Diamond$ *or* $\mathrm{PLTL}_\Box$,

- *the limit* $\lim_{\mathbf{v}} \mathcal{H}\left(\llbracket\varphi\rrbracket_{\mathbf{v}}\right)$ *always exists and is computable as logarithm of an algebraic real number;*
- *consequently, it is decidable whether* $\lim_{\mathbf{v}} \mathcal{H}\left(\llbracket\varphi\rrbracket_{\mathbf{v}}\right) = \mathcal{H}\left(\llbracket\varphi_\infty\rrbracket\right)$.

Our actual result

Theorem (Main)

Given a formula φ in $\mathrm{PLTL}_\Diamond$ *or* $\mathrm{PLTL}_\Box$*,*

- *the limit* $\lim_{\mathbf{v}} \mathcal{H}\left(\llbracket\varphi\rrbracket_{\mathbf{v}}\right)$ *always exists and is computable as logarithm of an algebraic real number;*
- *consequently, it is decidable whether* $\lim_{\mathbf{v}} \mathcal{H}\left(\llbracket\varphi\rrbracket_{\mathbf{v}}\right) = \mathcal{H}\left(\llbracket\varphi_\infty\rrbracket\right)$.

Method:

1. build parametrized automaton for φ;
2. find its "useful part" (independent of parameters value);
3. determinize it, compute its spectral radius, conclude.

"Generalised Büchi automata with parameters and counters" (BüAPC)

Definition (BüAPC with parameter set $\mathbf{t}$)

Tuple $\mathcal{A} = (Q, \Sigma, \Delta, \mathtt{Ctr}, Q_0, \mathtt{Acc})$, where

- Q, Σ, $Q_0 \subseteq Q$: as usual;
- $\mathtt{Ctr}$: finite set of time counters;
- $\Delta \subseteq Q \times \Sigma \times G_{\mathtt{Ctr},\mathbf{t}} \times 2^{\mathtt{Ctr}} \times Q$: transition relation;
- $\mathtt{Acc} \subseteq 2^{\Delta}$: finite set of colours (gen. Büchi conditions).

Transitions $q \xrightarrow{a,g,X} q' \in \Delta$: $g \in G_{\mathtt{Ctr},\mathbf{t}}$ is a guard, $a \in \Sigma$ is the action and $X \subseteq \mathtt{Ctr}$ is the reset component.
Guards: conjunctions $\bigwedge_i c_i \bowtie_i t_i$ ($c_i \in \mathtt{Ctr}$, $t_i \in \mathbf{t}$).

BüAPC semantics

For a BüAPC $\mathcal{B}$ and a valutation $\mathbf{v} \in \mathbb{N}^{\mathbf{t}}$, $Tr(\mathcal{B}, \mathbf{v})$ is a counter transition system:

- each transition increments all non-reset counters;
- a transition is firable when counters values satisfy its guard;
- a run is accepting when its starts in Q_0 and visits every colour infinitely often (Generalised Büchi condition).

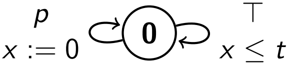

Figure: An automaton recognizing the language of formula $\Box\Diamond_t p$.

PLTL to BüAPC

Two subclasses of BüAPC

- BüAPC+ : parameters used as upper bounds only.
- BüAPC− : parameters used as lower bounds only.

Theorem

For a PLTL *formula* φ *over* AP *and* $\mathbf{t}$*, we can construct a* BüAPC $\mathcal{A}$ *over alphabet* 2^{AP} *parametrized by* $\mathbf{t}$ *such that*

- *for any* $\mathbf{v} \in \mathbb{N}^{\mathbf{t}}$, $[\![\varphi]\!]_{\mathbf{v}} = \mathcal{L}(Tr(\mathcal{A}, \mathbf{v}))$;
- *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−.

Construction sketch

Construction inspired by [Couvreur]:

- states are consistent sets of subformulas;
- each “colour” represents an obligation to satisfy an $\mathcal{U}$.

We added counters and guards:

- one counter per $\mathcal{R}_t$ and $\mathcal{U}_t$
- counters always reset except when relevant (i.e. within corresponding $\mathcal{R}_t$'s or $\mathcal{U}_t$'s scope)
- upperbounded guards allow “staying” in the scope of a $\mathcal{U}_t$;
- lowerbounded guards allow “escaping” the scope of a $\mathcal{R}_t$.

Exemple of construction

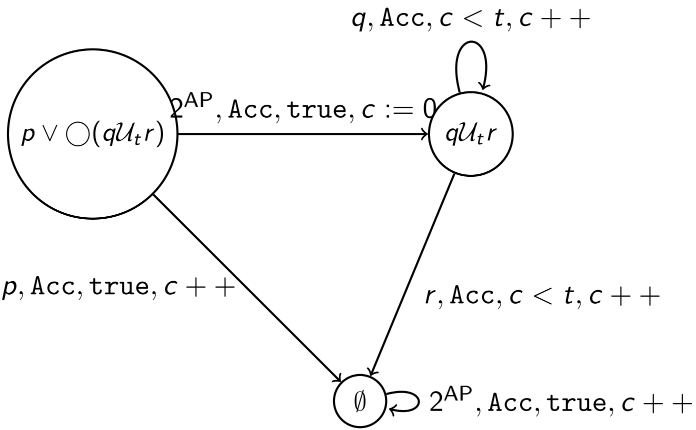

Figure: (simplified) automaton built for $p \vee \bigcirc(q\mathcal{U}_t r)$.

Here $\mathtt{Acc} = \emptyset$ because no $\mathcal{U}$ $\rightarrow$ all infinite runs are accepting.

Algorithm for BüAPC+

Data: a BüAPC+ $\mathcal{B}$
Result: $\mathcal{H} = \lim_{\mathbf{v}} \mathcal{H}(\mathcal{L}(\mathcal{B}, \mathbf{v}))$ as log of an algebraic number
SCC $\leftarrow$ `Tarjan(`$\underline{\mathcal{B}}$`);`
SCC$_G$ $\leftarrow$ set of non-trivial components resetting all counters;
SCC$_A$ $\leftarrow$ set of accepting non-trivial components;
$\mathcal{B}_1$ $\leftarrow$`trim(`$\underline{\mathcal{B}, Q_0, \mathrm{SCC}_A \cap \mathrm{SCC}_G}$`) ;` `/* useful part */`
$\mathcal{B}_2$ $\leftarrow$ `finite_automaton(`$\underline{\texttt{restrict(}\underline{B_1, \mathrm{SCC}_G}\texttt{)}}$`);`
`/* restricted to good SCC */`
return $\underline{\mathcal{H}(\mathcal{L}(\mathcal{B}_2))}$.

Algorithm 1: computing limit entropy for BüAPC+

Proposition

For a BüAPC+ $\mathcal{B}$, *the algorithm above computes* $\mathcal{H} = \lim_{\mathbf{v}} \mathcal{H}(\mathcal{L}(\mathcal{B}, \mathbf{v}))$.

Counter abstraction

We construct a symbolic Büchi Automaton:

- counter values are abstracted to either low or high;
- locations are splitted w.r.t. all possible sets of high counters: $q \to$ symbolic states $(q, C_1), (q, C_2), \ldots$ $C_i \subseteq \texttt{Ctr}$;
- transitions are
  - either normal: they mimick transitions of $\mathcal{B}$
  - or slow: $(q, C) \to (q, \texttt{Ctr} \backslash R')$, simulating the effect of an iterated cycle testing $C' \subseteq C$ and resetting R' such that $C' \cap R' = \emptyset$.

Abstracting counters on an example.

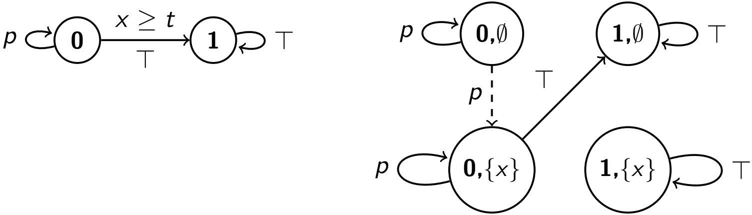

Figure: Concrete and symbolic automaton recognizing the language of the negative formula $\Box_t p$. The dashed arrow represents a slow transition.

Algorithm for BüAPC−

Data: a BüAPC− $\mathcal{B}$
Result: $\lim_{\mathbf{v}} \mathcal{H}(\mathcal{L}(\mathcal{B}, \mathbf{v}))$ as log of an algebraic number
$\mathcal{E} \leftarrow \texttt{symbolic}(\underline{\mathcal{B}})$;
`/* some transitions labelled as slow */`
$\mathcal{E}_1 \leftarrow \texttt{trim}(\underline{\mathcal{E}, Q_0 \times \emptyset, \texttt{Acc}})$;
$\mathcal{E}_2 \leftarrow \texttt{finite\_automaton}(\underline{\texttt{restrict}(\mathcal{E}_1, \text{normal transitions})})$;
`/* slow transitions removed */`
return $\underline{\mathcal{H}(\mathcal{L}(\mathcal{E}_2))}$;

Algorithm 2: computing limit entropy for BüAPC−

Proposition

For a BüAPC− $\mathcal{B}$, the algorithm above computes $\lim_{\mathbf{v}} \mathcal{H}(\mathcal{L}(\mathcal{B}, \mathbf{v}))$.

Proof sketch

$\mathcal{H}(L(\mathcal{E}_2)) \leq \mathcal{H}(L(\mathcal{B}, \mathbf{v}))$:

we prove that $Tr(\mathcal{B}, \mathbf{v})$ weakly simulates $\mathcal{E}$. On $\mathcal{E}_2$ (in particular its max-$\mathcal{H}$ SCC), the simulation is strong (same letters words).

$\mathcal{H}(L(\mathcal{B}, \mathbf{v}) \leq \mathcal{H}(L(\mathcal{E}_2)) + \eta$:

we prove that $\mathcal{E}$ simulates $Tr(\mathcal{B}, \mathbf{v})$ and can do it by using only some language of "low-density runs". Low-density runs use slow transitions ($\notin \mathcal{E}_2$), but rarely enough so that $\mathcal{H}(LD) \leq \mathcal{H}(L(\mathcal{E}_2)) + \eta$.

Summary

- We explored the notion of convergence of PLTL languages.
- We proved convergence in entropy for two subclases ($\mathrm{PLTL}_{\Diamond}$ and $\mathrm{PLTL}_{\Box}$).
- We defined a new class of automata and wrote the translation from PLTL.
- We showed how to compute entropy limits for two subclasses of BüAPC into which $\mathrm{PLTL}_{\Diamond}$ and $\mathrm{PLTL}_{\Box}$ translate.

Ongoing and related work

- entropy of ω-languages (relate to topology);
- experimental results;
- related experiments (ex: philosophers, where the parameter is the # of philsopohers, not a time bound);
- tool.

Thank you!

Questions?