Entropy and temporal specifications

Eugene Asarin[1], Michel Bockelet[2], Aldric Degorre [1],
Cătălin Dima[2] and Chunyan Mu[3]

[1]LIAFA – Université de Paris-Diderot
[2]LACL – Université de Paris-Est Créteil
[3]University of Birmingham

EQINOCS final workshop, May 9th, 2016

On qualitative and quantitative model-checking

Qualiltative model-checking

Given a system S and a property ϕ decide if $S \models \phi$ (answer: YES/NO).

- S: language of (ω-) words, automaton, Kripke structure, etc.
- φ: language of (ω-) words, automaton, formula in some logic (LTL, μ-calculus), etc.
- $\models$: language inclusion, model satisfaction, etc.

On qualitative and quantitative model-checking

Qualiltative model-checking

Given a system S and a property ϕ decide if $S \vDash \phi$ (answer: YES/NO).

- S: language of (ω-) words, automaton, Kripke structure, etc.
- φ: language of (ω-) words, automaton, formula in some logic (LTL, μ-calculus), etc.
- $\vDash$: language inclusion, model satisfaction, etc.

Quantitative model-checking

Given a system S and a property ϕ, measure how much $S \vDash \phi$ (answer: a real number).

Approaches:

- probability (PRISM/UppAal people, etc.)
- "reward/penalty" models (quantitative languages, simulation distances, etc.).

On qualitative and quantitative model-checking

Qualiltative model-checking

Given a system S and a property ϕ decide if $S \models \phi$ (answer: YES/NO).

- S: language of (ω-) words, automaton, Kripke structure, etc.
- φ: language of (ω-) words, automaton, formula in some logic (LTL, μ-calculus), etc.
- $\models$: language inclusion, model satisfaction, etc.

Quantitative model-checking

Given a system S and a property ϕ, measure how much $S \models \phi$ (answer: a real number).

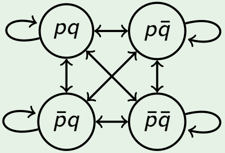
Approaches:

- probability (PRISM/UppAal people, etc.)
- "reward/penalty" models (quantitative languages, simulation distances, etc.).
- source of this work: entropy.

Why we are not happy with probability

Example

System S (state-labeled, note $\Sigma = 2^{\{p,q\}}$):



Specifications:

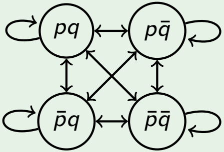
1. ϕ_1 = always p.
2. ϕ_2 = never 100 times in a row p.

In Linear Temporal Logic (LTL), $\phi_1 = \Box p$, $\phi_2 = \Box \Diamond_{<100} p$.

Why we are not happy with probability

Example

System S (state-labeled, note $\Sigma = 2^{\{p,q\}}$):



Specifications:

1. ϕ_1 = always p.
2. ϕ_2 = never 100 times in a row p.

In Linear Temporal Logic (LTL), $\phi_1 = \Box p$, $\phi_2 = \Box \Diamond_{<100}\, p$.

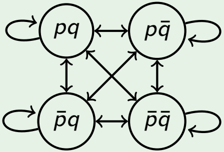
Naive analysis

- Certain effort required to satisfy ϕ_1 (never go below)
- A different (smaller?) effort required to satisfy ϕ_2 (go above at least every 100 units)

Why we are not happy with probability

Example

System S (state-labeled, note $\Sigma = 2^{\{p,q\}}$):



Specifications:

1. ϕ_1 = always p.
2. ϕ_2 = never 100 times in a row p.

In Linear Temporal Logic (LTL), $\phi_1 = \Box p$, $\phi_2 = \Box \Diamond_{<100}\, p$.

Naive analysis

- Certain effort required to satisfy ϕ_1 (never go below)
- A different (smaller?) effort required to satisfy ϕ_2 (go above at least every 100 units)

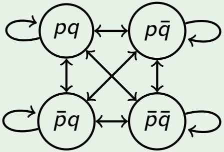
Probabilistic analysis

$\mathbb{P}(S \models \phi_1) = 0$ and $\mathbb{P}(S \models \phi_2) = 0$.

Why we are not happy with probability

Example

System S (state-labeled, note $\Sigma = 2^{\{p,q\}}$):



Specifications:

1. ϕ_1 = always p.
2. ϕ_2 = never 100 times in a row p.

In Linear Temporal Logic (LTL), $\phi_1 = \Box p$, $\phi_2 = \Box \Diamond_{<100} p$.

Naive analysis

- Certain effort required to satisfy ϕ_1 (never go below)
- A different (smaller?) effort required to satisfy ϕ_2 (go above at least every 100 units)

Probabilistic analysis

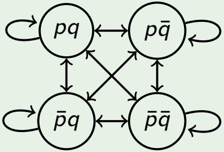
$\mathbb{P}(S \models \phi_1) = 0$ and $\mathbb{P}(S \models \phi_2) = 0$.

Mismatch between the two analyses

Our approach — entropy

Example

System S:



Specifications:

1. ϕ_1 = always p.
2. ϕ_2 = never 100 times in a row p.

In Linear Temporal Logic (LTL), $\phi_1 = \Box p$, $\phi_2 = \Box \Diamond_{<100}\, p$.

Entropy analysis

We associate a number (entropy) $\mathcal{H}$ to everything,

- Entropy of the system: $\mathcal{H}(S) = 2$.
- Entropy of runs satisfying ϕ_1 is $\mathcal{H}(S \cap \phi_1) = 1 < 2$
- Entropy of runs satisfying ϕ_2 is $\mathcal{H}(S \cap \phi_2) > 1.99$ (close to 2).

Matches the intuition!

What is entropy

Entropy of a finite word language (Chomsky, Miller)

For a language $L \subset \Sigma^*$, with $L_n = L \cap \Sigma^n$

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

Entropy of an ω-language (Staiger)

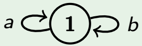
$$\mathcal{H}(L) = \mathcal{H}(\mathtt{pref}(L)) = \limsup_{n\to\infty} \frac{1}{n} \log \#\mathtt{pref}(L, n)$$

What does it mean

- Growth rate of the language: $\#L_n \approx 2^{\mathcal{H}n}$
- "average log(number of choices for a symbol)"
- Quantity of information (in bits/symbol) in words of L
- Related to compression, Kolmogorov complexity, topological entropy, Hausdorff dimension etc.

Entropy — examples

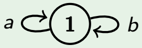
Example



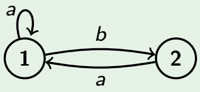
$$\mathcal{H}(\mathcal{L}(A)) = \log 2 = 1$$

Entropy — examples

Example



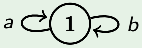
$$\mathcal{H}(\mathcal{L}(A)) = \log 2 = 1$$



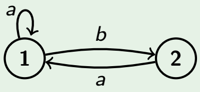
$$\mathcal{H}(\mathcal{L}(A)) = \log \frac{1+\sqrt{5}}{2}$$

Entropy — examples

Example



$$\mathcal{H}(\mathcal{L}(A)) = \log 2 = 1$$



$$\mathcal{H}(\mathcal{L}(A)) = \log \frac{1+\sqrt{5}}{2}$$

- $\mathcal{H}(\Sigma^\omega) = \log|\Sigma|$;
- Infinitely many times p:
  $\mathcal{H}([\![\Box \Diamond p]\!]) = \log|\Sigma|$ (no constraint most of the time);
- Eventually only p:
  $\mathcal{H}([\![\Diamond \Box p]\!]) = \log|\Sigma|$ (for any prefix, it is always possible to append p).

Entropy model-checking

The setting

- A system S — automaton/Kripke structure
- A specification ϕ — LTL formula

Entropy model-checking

The setting

- A system S — automaton/Kripke structure
- A specification ϕ — LTL formula

The metrics

With ω-languages L_S and L_ϕ consider the numbers:

- Entropy of the system $\mathcal{H}_S = H(L_S)$.
- Entropy of its good runs $\mathcal{H}_G = \mathcal{H}(L_S \cap L_\phi)$ and default $d = \mathcal{H}_S - \mathcal{H}_G$.
- Maybe entropy of bad runs $\mathcal{H}_B = \mathcal{H}(L_S \smallsetminus L_\phi)$.

Entropy model-checking

The setting

- A system S — automaton/Kripke structure
- A specification ϕ — LTL formula

The metrics

With ω-languages L_S and L_ϕ consider the numbers:

- Entropy of the system $\mathcal{H}_S = H(L_S)$.
- Entropy of its good runs $\mathcal{H}_G = \mathcal{H}(L_S \cap L_\phi)$ and default $d = \mathcal{H}_S - \mathcal{H}_G$.
- Maybe entropy of bad runs $\mathcal{H}_B = \mathcal{H}(L_S \setminus L_\phi)$.

An interpretation(???)

- d : how difficult is it to steer S into ϕ

Entropy model-checking

The setting

- A system S — automaton/Kripke structure
- A specification ϕ — LTL formula

The metrics

With ω-languages L_S and L_ϕ consider the numbers:

- Entropy of the system $\mathcal{H}_S = H(L_S)$.
- Entropy of its good runs $\mathcal{H}_G = \mathcal{H}(L_S \cap L_\phi)$ and default $d = \mathcal{H}_S - \mathcal{H}_G$.
- Maybe entropy of bad runs $\mathcal{H}_B = \mathcal{H}(L_S \setminus L_\phi)$.

An interpretation(???)

- d : how difficult is it to steer S into ϕ
- $d = 0$: entropy too rough, try probability

Computation bottleneck

Basic algorithm

- Build a Büchi automaton for the property ϕ.
- Build automata for $L_S \cap L_\phi$ and $L_S \setminus L_\phi$.
- Determinize.
- Compute the entropies.

Computation bottleneck

Basic algorithm

- Build a Büchi automaton for the property ϕ.
- Build automata for $L_S \cap L_\phi$ and $L_S \setminus L_\phi$.
- Determinize.
- Compute the entropies.

Enhancements

- Use advanced translation from LTL to (generalized, deterministic) Büchi.
- Decompose in strongly connected components.

Similarly to probabilistic model-checking, requires matrix algebra over large matrices (size potentially $\sim Exp(\text{number of variables})$).

Basic properties

- $0 \leq \mathcal{H}_G, \mathcal{H}_B \leq \mathcal{H}_S \leq \log|\Sigma|$
- $\mathbb{P}(\phi) > 0 \Rightarrow \mathcal{H}_G = \mathcal{H}_S$
- $\mathcal{H}(\phi_1 \vee \phi_2) = \max(\mathcal{H}(\phi_1), \mathcal{H}(\phi_2))$
- $\mathcal{H}(\Diamond\phi) = \log|\Sigma|$ (or 0 if empty).
- $H_G < H_S \Leftrightarrow L_\phi$ nowhere dense in L_S)

Some additionnal remarks

Reminder

Every ϕ can be represented as $\sigma \wedge \lambda$ (safety and liveness)

- Safety: avoid some bad states.
- Liveness: something good happens infinitely often.

For entropy, only safety matters

$$\mathcal{H}(L_S \cap L_\phi) = \mathcal{H}(L_S \cap L_\sigma)$$

Back to our initial example

Recall:

1. ϕ_1 = always p.
2. ϕ_2 = never 100 times in a row p.

In Linear Temporal Logic (LTL), $\phi_1 = \Box p$, $\phi_2 = \Box \Diamond_{<100}\, p$.

Entropy analysis

- Entropy of runs satisfying ϕ_1 is $\mathcal{H}(S \cap \phi_1) = 1 < 2$
- Entropy of runs satisfying ϕ_2 is $\mathcal{H}(S \cap \phi_2) > 1.99$ (close to 2).

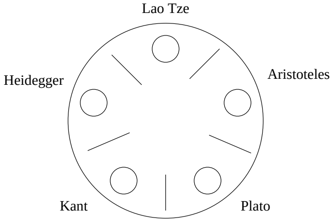
Other relevant examples?

A case study

Problem

n dining philosophers, simplified

- *n philosophers sit around a round table.*
- *Single bowl of spaghetti in the middle.*
- *n chopsticks, each placed between two philosophers.*
- *To eat, each philosophers needs two chopsticks.*
- *Race conditions on chopsticks, deadlocks possible if anarchy.*



A case study: n dining philosophers, simplified

Languages considered

- $\mathcal{L}_S$: all the runs.
- $\mathcal{L}_S \setminus \mathcal{L}_D$: runs w/o deadlock
- $\mathcal{L}_S \cap \mathcal{L}_{NS}$: no philosopher ever starves.
- $\mathcal{L}_S \cap \mathcal{L}_{Et}$: philosopher 1 eats at least every t time units.

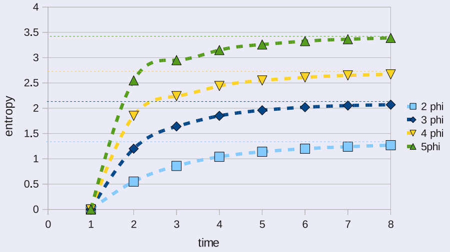
A case study: *n* dining philosophers, simplified

Languages considered

- $\mathcal{L}_S$: all the runs.
- $\mathcal{L}_S \setminus \mathcal{L}_D$: runs w/o deadlock
- $\mathcal{L}_S \cap \mathcal{L}_{NS}$: no philosopher ever starves.
- $\mathcal{L}_S \cap \mathcal{L}_{Et}$: philosopher 1 eats at least every t time units.

Entropy analysis

The first three entropies coincide, the fourth one depends on t and converges.



Dining philosophers lesson

- $\Box \Diamond e$ = no philosopher ever starves.
- $\Box \Diamond_{\leq t} e$ = philosopher 1 eats at least every t time units.

$\mathcal{H}(\Box \Diamond_{\leq t} e) \to \mathcal{H}(\Box \Diamond e)$ as $t \to \infty$.

Problem

Asymptotics in LTL Let ϕ_t be an LTL formula with parameter (time bound) t, let ϕ_∞ its unbounded version. Is it true that $\mathcal{H}(\phi_t) \to \mathcal{H}(\phi_\infty)$ for $t \to \infty$?

Dining philosophers lesson

- $\Box \Diamond e$ = no philosopher ever starves.
- $\Box \Diamond_{\leq t} e$ = philosopher 1 eats at least every t time units.

$\mathcal{H}(\Box \Diamond_{\leq t} e) \to \mathcal{H}(\Box \Diamond e)$ as $t \to \infty$.

Problem

Asymptotics in LTL Let ϕ_t be an LTL formula with parameter (time bound) t, let ϕ_∞ its unbounded version. Is it true that $\mathcal{H}(\phi_t) \to \mathcal{H}(\phi_\infty)$ for $t \to \infty$?

The answer

Sometimes. More details next.

LTL

Linear Temporal logic over boolean variables $p \in \mathrm{AP}$:

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

and standard "syntactic sugar":

$$\Diamond\varphi = \top\mathcal{U}\varphi \qquad \Box\varphi = \bot\mathcal{R}\varphi \text{ (or "}\neg\Diamond\neg\varphi\text{")}$$

Models: infinite words in $\left(2^{\mathrm{AP}}\right)^{\omega}$.

Example

p	0	1	1	0	0	...(only 0s)
$\Diamond p$	1	1	1	0	0	...

PLTL

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

(Parametric) Linear Temporal logic over boolean variables $p \in AP$ and parameters $t \in Param$:

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

- Distinct parameters for distinct subformulas.
- Standard "syntactic sugar":

$$\Diamond_t\varphi = \top\mathcal{U}_t\varphi \qquad \Box_t\varphi = \bot\mathcal{R}_t\varphi \text{ (or "}\neg\Diamond_t\neg\varphi\text{")}$$

PLTL semantics in a nutshell

- $\varphi \mathcal{U}_t \psi$: ψ must become true before t seconds and φ remain true until then;
- $\varphi \mathcal{R}_t \psi$: ψ must remain true until t seconds elapse or φ becomes true;

and hence, in particular,

- $\Diamond_t \varphi$: φ becomes true before t seconds;
- $\Box_t \varphi$: φ remains true for t seconds.

Example

p	0	1	1	1	0	0	0	0	1 . . . (only 0s)
$[\![\Diamond_t p]\!]_{t \leftarrow 2}$	1	1	1	1	0	0	1	1	0 . . .
$[\![\Box_t p]\!]_{t \leftarrow 2}$	0	1	0	0	0	0	0	0	0 . . .

Temporal formulas: unbounded vs. parametric

- Unbounded formula: $\varphi_\infty = \Box \Diamond p$, i.e. "infinitely often p".
- Its parametric variant: $\varphi_t = \Box \Diamond_t p$, i.e. less than t seconds between two ps.
- In theory we like unbounded formulas.
- Concrete applications often "prefer" parametric specifications.
- Is φ_t close to φ_∞ for t sufficiently big?

Temporal formulas: unbounded vs. parametric

- Unbounded formula: $\varphi_\infty = \Box \Diamond p$, i.e. "infinitely often p".
- Its parametric variant: $\varphi_t = \Box \Diamond_t p$, i.e. less than t seconds between two ps.
- In theory we like unbounded formulas.
- Concrete applications often "prefer" parametric specifications.
- Is φ_t close to φ_∞ for t sufficiently big?

Problem

Give an interpretation to $\lim_t \Box \Diamond_t p = \Box \Diamond p$.

Notations

- $w \in (2^{\mathrm{AP}})^\omega, \mathbf{v} \in \mathbb{N}^{Param}$ then $w, \mathbf{v} \models \varphi$ whenever $w \models \varphi[t \leftarrow \mathbf{v}]$
- $[\![\varphi]\!]_{\mathbf{v}} = \{w \in (2^{\mathrm{AP}})^\omega \mid w, \mathbf{v} \models \varphi\}$.
- φ_∞ = the formula in which all bounded operators are replaced with their unbounded analogs.

$$\left(\Diamond \Box_t p\right)_\infty = \Diamond \Box p$$

Our problem, reformulated

How "close" is φ_t to φ_∞ for big t's?

Interpreting $\lim_t \Box \Diamond_t p = \Box \Diamond p$

Set-theoretic interpretation?

- $[\![\Box \Diamond_t p]\!]_{\mathbf{v}}$ is monotonic (increasing wrt $\mathbf{v} \in \mathbb{N}$) .
- Its limit exists and is

$$\bigcup_{\mathbf{v}\in\mathbb{N}} [\![\Box \Diamond_t p]\!]_{\mathbf{v}}$$

Interpreting $\lim_t \Box \Diamond_t p = \Box \Diamond p$

Set-theoretic interpretation?

- $[\![\Box \Diamond_t p]\!]_{\mathbf{v}}$ is monotonic (increasing wrt $\mathbf{v} \in \mathbb{N}$) .
- Its limit exists and is

$$\bigcup_{\mathbf{v} \in \mathbb{N}} [\![\Box \Diamond_t p]\!]_{\mathbf{v}}$$

- ... but it is not an ω-regular language:

$$\bigcup_{\mathbf{v} \in \mathbb{N}} [\![\Box \Diamond_t p]\!]_{\mathbf{v}} =$$ "words having (uniformly) upper-bounded subsequences of $\neg p$"

- So $\bigcup_{\mathbf{v} \in \mathbb{N}} [\![\Box \Diamond_t p]\!]_{\mathbf{v}} \neq [\![\Box \Diamond p]\!]$.

Interpreting $\lim_t \Box \Diamond_t p = \Box \Diamond p$

Topological interpretation?

- Work with (topological) closures:

$$cl\Big(\bigcup_{t\in\mathbb{N}} [\![\Box \Diamond_t p]\!]\Big) = cl([\![\Box \Diamond p]\!]) = [\![\mathtt{true}]\!]$$

- But also:

$$cl\Big(\bigcap_{t\in\mathbb{N}} [\![\Diamond \Box_t p]\!]\Big) = cl([\![\Diamond \Box p]\!]) = [\![\mathtt{true}]\!]?$$

- Also not clear how to generalize to formulas with nested bounded operators (even if the operators have the same "polarity").

Interpreting $\lim_t \Box \Diamond_t p = \Box \Diamond p$

Probabilistic interpretations?

Incompatibility with "convergence" of formulas

Take any Markov chain $\mathcal{M}$ with positive probabilities and p true in some state and false in some other.

- Then $Pr(\mathcal{M}, \mathbf{v} \models \Box \Diamond_t p) = 0$ for all $\mathbf{v} \in \mathbb{N}$;
- but meanwhile $Pr(\mathcal{M} \models \Box \Diamond p) = 1$.

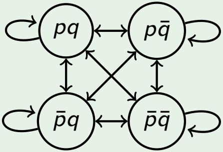
Too coarse metric

Many interesting probabilities are actually either 0 or 1.

Interpreting $\lim_t \square \diamond_t p = \square \diamond p$

Probabilistic interpretations?

Example

System S: 

Specifications: $\phi = \square p$, or more involved ψ = never 100 times in a row $\bar{p} = \square \diamond_{<100} p$.

Our proposal for interpreting $\lim_t \Box \Diamond_t p = \Box \Diamond p$

Interpretation as entropy

Convergence in entropy

$$\lim_{\mathbf{v}\to\infty} \mathcal{H}([\![\Box \Diamond_t p]\!]_{\mathbf{v}}) = \lim_{\mathbf{v}\to\infty} \left(|AP| - 2^{-\mathbf{v}}\right) = |AP| = \mathcal{H}([\![\Box \Diamond p]\!])$$

$$\lim_{\mathbf{v}\to\infty} \mathcal{H}([\![\Diamond \Box_t p]\!]_{\mathbf{v}}) = \lim_{\mathbf{v}\to\infty} |AP| = |AP| = \mathcal{H}([\![\Diamond \Box p]\!])$$

Our proposal for interpreting $\lim_t \Box \Diamond_t p = \Box \Diamond p$

Interpretation as entropy

Convergence in entropy

$$\lim_{\mathbf{v}\to\infty} \mathcal{H}([\![\Box \Diamond_t p]\!]_{\mathbf{v}}) = \lim_{\mathbf{v}\to\infty} \left(|AP| - 2^{-\mathbf{v}}\right) = |AP| = \mathcal{H}([\![\Box \Diamond p]\!])$$

$$\lim_{\mathbf{v}\to\infty} \mathcal{H}([\![\Diamond \Box_t p]\!]_{\mathbf{v}}) = \lim_{\mathbf{v}\to\infty} |AP| = |AP| = \mathcal{H}([\![\Diamond \Box p]\!])$$

But also for all $\mathbf{v}$,

$$\mathcal{H}([\![\Diamond_t \Box p]\!]_{\mathbf{v}}) = 1 \neq 2 = \mathcal{H}([\![\Diamond \Box p]\!])$$

Goal

We want to decide whether $\lim_{\mathbf{v}} \mathcal{H}([\![\phi_t]\!]_{\mathbf{v}}) = \mathcal{H}([\![\phi_\infty]\!])$.

Restricting to fragments of PLTL

First, some bad news

For instance: $\Box_t p \wedge \Diamond_s \neg p$ admits no entropy limit.

So we restrict our problem to:

Fragments of PLTL [Alur et al, ICALP'99]

- $\mathrm{PLTL}_{\Diamond}$: PLTL without $\mathcal{R}_t$, "positive fragment".

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

- $\mathrm{PLTL}_{\Box}$: PLTL without $\mathcal{U}_t$, "negative fragment".

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

Our actual result

Theorem (Main)

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

- $\lim_{\mathbf{v}} \mathcal{H}\left(\llbracket\varphi\rrbracket_{\mathbf{v}}\right)$ *always exists and is computable as the 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)

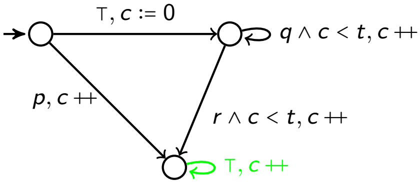
Given a formula φ *in* $\mathrm{PLTL}_\diamond$ *or* $\mathrm{PLTL}_\Box$,

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

Method for computing $\lim_{\mathbf{v}} \mathcal{H}$

1. Build a parameterized Büchi automaton for φ.
2. Find its useful part (details depend on $\mathrm{PLTL}_\diamond$ or $\mathrm{PLTL}_\Box$).
3. Determinize the “limit” automaton, compute its spectral radius, conclude.

Generalized Büchi automata with parameters and counters (BüAPC)



BüAPC≃ discrete timed automaton with parameters

- $p, q, r \in \mathrm{AP}$
- c is a counter (a discrete clock either incremented or reset at each transition)
- t is a parameter
- all transition colors (here: only green) must be visited infinitely often
- for a BüAPC $\mathcal{B}$, $\mathcal{L}(\mathcal{B}, \mathbf{v})$ is its language for $t := \mathbf{v}$

Where is entropy produced in a GTBAC?

We need to compute

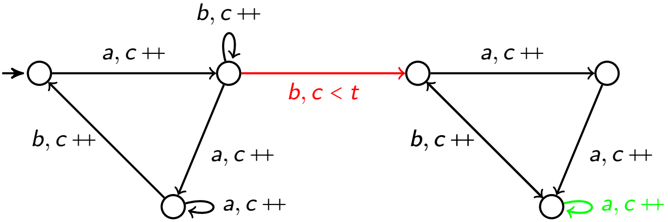
$$\lim_{v\to\infty} \mathcal{H}\big(\mathcal{L}(\mathcal{B},\mathbf{v})\big) = \lim_{v\to\infty} \limsup_{n\to\infty} \frac{1}{n} \log \#\mathcal{L}_n(\mathcal{B},\mathbf{v})$$

Where is entropy produced in a GTBAC?

We need to compute

$$\lim_{v\to\infty} \mathcal{H}\big(\mathcal{L}(\mathcal{B},\mathbf{v})\big) = \lim_{v\to\infty} \limsup_{n\to\infty} \frac{1}{n} \log \#\mathcal{L}_n(\mathcal{B},\mathbf{v})$$

One single transition with a lower guard, no resets:



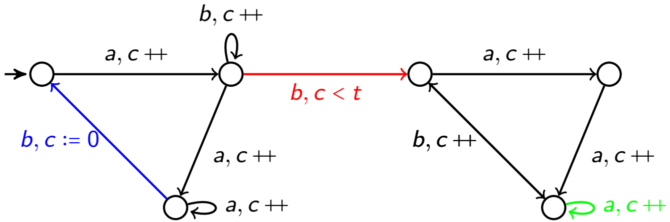
Only the right-hand side component produces entropy for any t.

Where is entropy produced in a GTBAC?

We need to compute

$$\lim_{v\to\infty} \mathcal{H}\big(\mathcal{L}(\mathcal{B},\mathbf{v})\big) = \lim_{v\to\infty} \limsup_{n\to\infty} \frac{1}{n} \log \#\mathcal{L}_n(\mathcal{B},\mathbf{v})$$

One single transition with a lower guard, some resets:



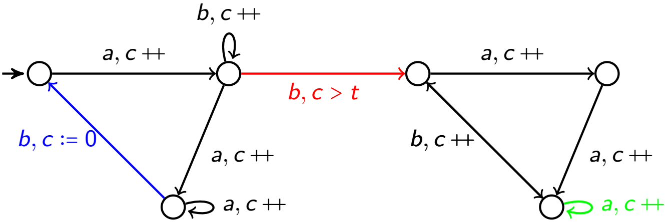
The left-hand side component produces the entropy: any run can be modified by looping through the blue reset and then taking the red transition.

Where is entropy produced in a GTBAC?

We need to compute

$$\lim_{v\to\infty} \mathcal{H}\big(\mathcal{L}(\mathcal{B}, \mathbf{v})\big) = \lim_{v\to\infty} \limsup_{n\to\infty} \frac{1}{n} \log \#\mathcal{L}_n(\mathcal{B}, \mathbf{v})$$

One single transition with an upper guard, some resets:



The left-hand side component produces entropy since any run can be modified by looping sufficiently (at most t times) in state 2.

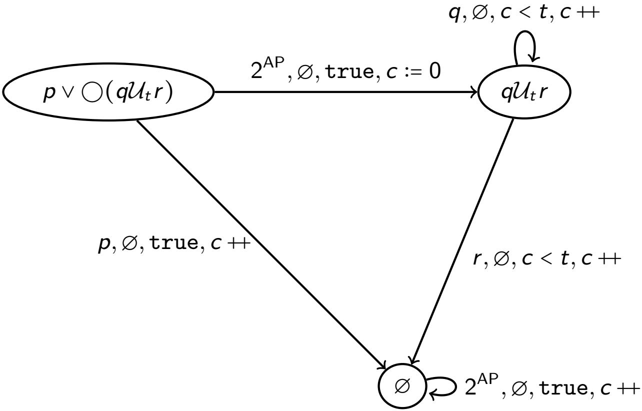
Construction sketch

(construction inspired by [Couvreur], extended with counters for $\mathcal{R}_t$ and $\mathcal{U}_t$)

- states: consistent sets of subformulas;
- "colours": obligations to satisfy an $\mathcal{U}$ (1 for each occurrence).
- counters: for satisfying $\mathcal{R}_t$ and $\mathcal{U}_t$ (1 for each occurrence):
  - counters always reset except when relevant
    (i.e. within corresponding $\mathcal{R}_t$'s or $\mathcal{U}_t$'s scope)
  - upper-bounded guards allow "staying" in the scope of a $\mathcal{U}_t$;
  - lower-bounded guards allow "escaping" the scope of a $\mathcal{R}_t$.

Example of construction

Automaton built for $p \vee \bigcirc(q\mathcal{U}_t r)$



No color because there is no $\mathcal{U}$. All infinite runs are accepting.

PLTL to BüAPC

Two subclasses of BüAPC

- BüAPC+ : all guards are upper bounds $\bigwedge_i x_i \leq t_i$
- BüAPC− : all guards are lower bounds $\bigwedge_i x_i \geq t_i$

PLTL to BüAPC

Two subclasses of BüAPC

- BüAPC+ : all guards are upper bounds $\bigwedge_i x_i \leq t_i$
- BüAPC− : all guards are lower bounds $\bigwedge_i x_i \geq t_i$

Theorem

For a PLTL *formula* φ, *we can construct a* BüAPC $\mathcal{A}$ *such that*

- *for any* $\mathbf{v} \in \mathbb{N}^{Param}$, $[\![\varphi]\!]_{\mathbf{v}} = \mathcal{L}(\mathcal{A}, \mathbf{v})$;
- *if* φ *is in* $\mathrm{PLTL}_{\Diamond}$ *then* $\mathcal{A}$ *is a* BüAPC+;
- *and if* φ *is in* $\mathrm{PLTL}_{\Box}$ *then* $\mathcal{A}$ *is a* BüAPC−.

Key result

Theorem

For any BüAPC+ *or* BüAPC−, $\mathcal{B}$, *the limit entropy* $\lim_{\mathbf{v}} \mathcal{H}(\mathcal{L}(\mathcal{B}, \mathbf{v}))$ *exists and can be computed.*

...and thus the main theorem (stated before) directly follows: limit entropy of $PLTL_{\Diamond}$ and $PLTL_{\Box}$ formulas can be computed.

BüAPC+: asymptotic analysis, a single strongly connected component

$\mathcal{B}$: BüAPC+ (guards: $x < t$), $\mathbf{v} \to \infty$

- If $\mathcal{B}$ does not reset all counters, $\mathcal{L}(\mathcal{B}, \mathbf{v}) = \varnothing$.
- Otherwise ($\mathcal{B}$ resets all counters)
  - $\mathcal{B}_\infty := \mathcal{B}$ without constraints and parameters.
  - Clearly $\mathcal{H}(\mathcal{B}, \mathbf{v}) \leq \mathcal{H}(\mathcal{B}_\infty)$, since $\mathcal{L}(\mathcal{B}, \mathbf{v}) \subseteq \mathcal{L}(\mathcal{B}_\infty)$.
  - Other direction: $\frac{|\mathbf{v}|+c}{|\mathbf{v}|}\mathcal{H}(\mathcal{B}, \mathbf{v}) > \mathcal{H}(\mathcal{B}_\infty)$ (see below the proof method).
  - Thus $\lim_{\mathbf{v}} \mathcal{H}(\mathcal{B}, \mathbf{v}) = \mathcal{H}(\mathcal{B}_\infty)$.

Proof method

Construct an injection $(\mathcal{L}(\mathcal{B}_\infty) \to \mathcal{L}(\mathcal{B}, \mathbf{v}))$ that inserts resetting cycles every $\sim |\mathbf{v}|$ transitions

$\Rightarrow$ constraints of $\mathcal{B}_{\mathbf{v}}$ satisfied

$\Rightarrow$ small increase of length.

BüAPC+: computing the limit entropy

General case: Only consider (reachable, co-reachable, ...) SCCs of $\mathcal{B}$ that reset all counters.

Idea of the algorithm

- Find the part of $\mathcal{B}$ that resets all counters and is usable in accepting runs (for all $\mathbf{v}$).
- Compute its entropy.

BüAPC+: computing the limit entropy

Algorithm

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

Proposition

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

BüAPC–: asymptotic analysis

$\mathcal{B}$: BüAPC– (guards: $x > t$), $\mathbf{v} \to \infty$

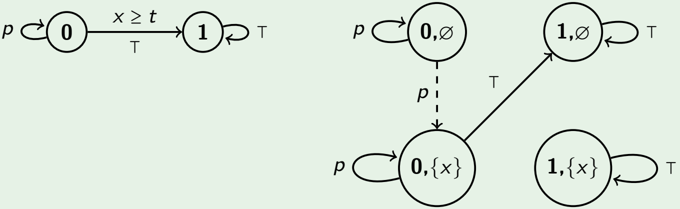
Essential object to build

Symbolic automaton $\mathcal{E}$, mimicking $\mathcal{B}$ for big $\mathbf{v}$.

Construction idea

$\mathcal{E}$ remembers which counters are big. Thus we know what transitions can be fired. $\mathcal{E}$ also has "pumping" transitions everywhere $\mathcal{B}$ had non-resetting cycles.

Example ($\mathcal{B}$ and $\mathcal{E}$ for $\Box_t p$)



Dashed arrow: a "pumping" transition.

BüAPC–: computing limit entropy

Idea of the algorithm

- Build symbolic automaton $\mathcal{E}$
- Compute the entropy of its useful part.

Algorithm

Data: a BüAPC– $\mathcal{B}$
Result: $\lim_{\mathbf{v}} \mathcal{H}(\mathcal{L}(\mathcal{B}, \mathbf{v}))$ as log of an algebraic number
$\mathcal{E} \leftarrow \mathtt{symbolic}(\mathcal{B})$;
$\mathcal{E}_1 \leftarrow \mathtt{trim}(\mathcal{E}, Q_0 \times \emptyset, \mathtt{Acc})$;
$\mathcal{E}_2 \leftarrow \mathtt{restrict}(\mathcal{E}_1$, non-pumping transitions);
return $\mathcal{H}(\mathcal{L}(\mathcal{E}_2))$;

Proposition

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

Conclusions

Problems

- How to formalize asymptotic convergence for PLTL?
- How to decide it?

Conclusions

Problems

- How to formalize asymptotic convergence for PLTL?
- How to decide it?

Results

- Comparing convergence in entropy to other convergences.
- Criteria of convergence in entropy for $PLTL_{\diamond}$ and $PLTL_{\square}$.
- Computing limits of entropies for BüAPC+ and BüAPC−.

Conclusions

Problems

- How to formalize asymptotic convergence for PLTL?
- How to decide it?

Results

- Comparing convergence in entropy to other convergences.
- Criteria of convergence in entropy for $PLTL_{\diamond}$ and $PLTL_{\Box}$.
- Computing limits of entropies for BüAPC+ and BüAPC−.

Open questions and further work

- Entropy and topology?
- Relevance in verification?
- Extensions to branching temporal logics?

Conclusions

Problems

- How to formalize asymptotic convergence for PLTL?
- How to decide it?

Results

- Comparing convergence in entropy to other convergences.
- Criteria of convergence in entropy for $PLTL_{\Diamond}$ and $PLTL_{\Box}$.
- Computing limits of entropies for BüAPC+ and BüAPC−.

Open questions and further work

- Entropy and topology?
- Relevance in verification?
- Extensions to branching temporal logics?

Thank you!