Entropy and temporal specifications

Eugene Asarin\textsuperscript{1}, Michel Bockelet\textsuperscript{2}, Aldric Degorre \textsuperscript{1}, Cătălin Dima\textsuperscript{2} and Chunyan Mu\textsuperscript{3}

\textsuperscript{1}LIAFA – Université de Paris-Diderot  
\textsuperscript{2}LACL – Université de Paris-Est Créteil  
\textsuperscript{3}University of Birmingham

EQINOCS final workshop, May 9th, 2016
1 Entropy and quantitative model-checking
- Quantitative model-checking in very few slides
- Entropy used as a measure
- Some experiments

2 Entropy and asymptotics
- Parametric linear temporal logic (PLTL)
- Convergence problems for PLTL formulas

3 Main result and techniques
- Discrete timed automata with parameters (GTBAC)
- Producing entropy in GTBAC
- Translating from PLTL to GTBAC

4 Computing limit entropies
- “Positive” case
- “Negative” case

5 Conclusions
On qualitative and quantitative model-checking

Qualitative model-checking

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

- $S$: language of ($\omega$-) words, automaton, Kripke structure, etc.
- $\varphi$: language of ($\omega$-) words, automaton, formula in some logic (LTL, $\mu$-calculus), etc.
- $\models$: language inclusion, model satisfaction, etc.
On qualitative and quantitative model-checking

Qualitative model-checking

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

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

Quantitative model-checking

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

Approaches:

- probability (PRISM/UppAal people, etc.)
- “reward/penalty” models (quantitative languages, simulation distances, etc.).
On qualitative and quantitative model-checking

**Qualitative model-checking**
Given a system $S$ and a property $\phi$ decide if $S \models \phi$ (answer: YES/NO).
- $S$: language of ($\omega$-) words, automaton, Kripke structure, etc.
- $\phi$: language of ($\omega$-) words, automaton, formula in some logic (LTL, $\mu$-calculus), etc.
- $\models$: language inclusion, model satisfaction, etc.

**Quantitative model-checking**
Given a system $S$ and a property $\phi$, 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. $\phi_1 = \text{always } p$.
2. $\phi_2 = \text{never 100 times in a row } p$.

In Linear Temporal Logic (LTL), $\phi_1 = \Box p$, $\phi_2 = \Box \lozenge_{<100} p$. 
Why we are not happy with probability

Example

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

Specifications:
1. $\phi_1 = \text{always } p$.
2. $\phi_2 = \text{never 100 times in a row } p$.

In Linear Temporal Logic (LTL), $\phi_1 = \square p$, $\phi_2 = \square \diamond <100 p$.

Naive analysis

- Certain effort required to satisfy $\phi_1$ (never go below)
- A different (smaller?) effort required to satisfy $\phi_2$ (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. $\phi_1 = \text{always } p$.
2. $\phi_2 = \text{never 100 times in a row } p$.

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

Naive analysis

- Certain effort required to satisfy $\phi_1$ (never go below)
- A different (smaller?) effort required to satisfy $\phi_2$ (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. $\phi_1 = \text{always } p$.
2. $\phi_2 = \text{never 100 times in a row } p$.

In Linear Temporal Logic (LTL), $\phi_1 = \square p$, $\phi_2 = \square \Diamond <100 \ p$.

Naive analysis

- Certain effort required to satisfy $\phi_1$ (never go below)
- A different (smaller?) effort required to satisfy $\phi_2$ (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. $\phi_1 = \text{always } p$.
2. $\phi_2 = \text{never 100 times in a row } p$.

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 $\phi_1$ is $\mathcal{H}(S \cap \phi_1) = 1 < 2$
- Entropy of runs satisfying $\phi_2$ 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 $L_n = L \cap \Sigma^n$

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

Entropy of an $\omega$-language (Staiger)

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

What does it mean

- Growth rate of the language: $\#L_n \approx 2^{Hn}$
- “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

\[ H(L(A)) = \log_2 1 = 1 \]
Entropy — examples

Example

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

\[ H(\mathcal{L}(A)) = \log \frac{1 + \sqrt{5}}{2} \]
Entropy — examples

Example

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

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

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

The setting

- A system $S$ — automaton/Kripke structure
- A specification $\phi$ — LTL formula
Entropy model-checking

The setting

- A system $S$ — automaton/Kripke structure
- A specification $\phi$ — LTL formula

The metrics

With $\omega$-languages $L_S$ and $L_\phi$ consider the numbers:

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

The setting

- A system $S$ — automaton/Kripke structure
- A specification $\phi$ — LTL formula

The metrics

With $\omega$-languages $L_S$ and $L_\phi$ consider the numbers:

- Entropy of the system $H_S = H(L_S)$.
- Entropy of its good runs $H_G = H(L_S \cap L_\phi)$ and default $d = H_S - H_G$.
- Maybe entropy of bad runs $H_B = H(L_S \setminus L_\phi)$.

An interpretation(???)

- $d$ : how difficult is it to steer $S$ into $\phi$
Entropy model-checking

The setting

- A system $S$ — automaton/Kripke structure
- A specification $\phi$ — LTL formula

The metrics

With $\omega$-languages $L_S$ and $L_\phi$ consider the numbers:

- Entropy of the system $H_S = H(L_S)$.
- Entropy of its good runs $H_G = H(L_S \cap L_\phi)$ and default $d = H_S - H_G$.
- Maybe entropy of bad runs $H_B = H(L_S \setminus L_\phi)$.

An interpretation(???)

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

Basic algorithm

- Build a Büchi automaton for the property $\phi$.
- 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 $\phi$.
- 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 $\sim \text{Exp(number of variables)}$).
Basic properties

- \( 0 \leq H_G, H_B \leq H_S \leq \log |\Sigma| \)
- \( P(\phi) > 0 \Rightarrow H_G = H_S \)
- \( H(\phi_1 \lor \phi_2) = \max(H(\phi_1), H(\phi_2)) \)
- \( H(\diamond \phi) = \log |\Sigma| \) (or 0 if empty).
- \( H_G < H_S \Leftrightarrow L_\phi \) nowhere dense in \( L_S \)
Some additional remarks

Reminder

Every $\phi$ can be represented as $\sigma \land \lambda$ (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. $\phi_1 = \text{always } p$.
2. $\phi_2 = \text{never 100 times in a row } p$.

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

**Entropy analysis**

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

Other relevant examples?
A case study

Problem

\textit{n dining philosophers, simplified}

- \textit{n philosophers sit around a round table.}
- \textit{Single bowl of spaghetti in the middle.}
- \textit{n chopsticks, each placed between two philosophers.}
- \textit{To eat, each philosophers needs two chopsticks.}
- \textit{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

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

$$\mathcal{H}(\square \Diamond_{\leq t} e) \rightarrow \mathcal{H}(\square \Diamond e) \text{ as } t \rightarrow \infty.$$ 

**Problem**

*Asymptotics in LTL* Let $\phi_t$ be an LTL formula with parameter (time bound) $t$, let $\phi_\infty$ its unbounded version. Is it true that $\mathcal{H}(\phi_t) \rightarrow \mathcal{H}(\phi_\infty)$ for $t \rightarrow \infty$?
Dining philosophers lesson

- □ ◻ e = no philosopher ever starves.
- □ ◻<sub>t</sub> e = philosopher 1 eats at least every <i>t</i> time units.

\[ \mathcal{H}(\square \Diamond <sub>t</sub> e) \rightarrow \mathcal{H}(\square \Diamond e) \text{ as } t \rightarrow \infty. \]

**Problem**

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

**The answer**

Sometimes. More details next.
LTL

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

$$\varphi ::= p \mid \neg p \mid \Diamond \varphi \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \varphi U \varphi \mid \varphi R \varphi$$

and standard “syntactic sugar”:

$$\Diamond \varphi = T U \varphi \quad \Box \varphi = \bot R \varphi \quad \text{(or } \neg \Diamond \neg \varphi \text{)}$$

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

**Example**

<table>
<thead>
<tr>
<th>$p$</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>... (only 0s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Diamond p$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
</tbody>
</table>
**PLTL**

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

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

$$\varphi ::= p \mid \neg p \mid \Diamond \varphi \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \varphi U \varphi \mid \varphi R \varphi \mid \varphi U_t \varphi \mid \varphi R_t \varphi$$

- **Distinct parameters for distinct subformulas.**
- **Standard “syntactic sugar”:**
  $$\Diamond_t \varphi = T U_t \varphi \quad \Box_t \varphi = \bot R_t \varphi$$
  (or “$\neg \Diamond_t \neg \varphi$”)
PLTL semantics in a nutshell

- \( \varphi U_t \psi \): \( \psi \) must become true before \( t \) seconds and \( \varphi \) remain true until then;
- \( \varphi R_t \psi \): \( \psi \) must remain true until \( t \) seconds elapse or \( \varphi \) becomes true;

and hence, in particular,

- \( \Diamond_t \varphi \): \( \varphi \) becomes true before \( t \) seconds;
- \( \Box_t \varphi \): \( \varphi \) remains true for \( t \) seconds.

Example

<table>
<thead>
<tr>
<th>( p )</th>
<th>0 1 1 1 0 0 0 0 1 . . . (only 0s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( [\Diamond_t p]_{t \leftarrow 2} )</td>
<td>1 1 1 1 0 0 1 1 0 . . .</td>
</tr>
<tr>
<td>( [\Box_t p]_{t \leftarrow 2} )</td>
<td>0 1 0 0 0 0 0 0 0 . . .</td>
</tr>
</tbody>
</table>
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 $p$s.
- In theory we like unbounded formulas.
- Concrete applications often “prefer” parametric specifications.
- Is $\varphi_t$ close to $\varphi_\infty$ for $t$ sufficiently big?
Temporal formulas: unbounded vs. parametric

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

Problem

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

- $w \in (2^{AP})^\omega$, $v \in \mathbb{N}^{Param}$ then $w, v \models \phi$ whenever $w \models \phi[t \leftarrow v]$.
- $[\phi]_v = \{w \in (2^{AP})^\omega \mid w, v \models \phi\}$.
- $\phi_\infty$ = the formula in which all bounded operators are replaced with their unbounded analogs.

$$(\lozenge \square_t p)_\infty = \lozenge \square p$$

Our problem, reformulated

How “close” is $\phi_t$ to $\phi_\infty$ for big $t$’s?
Interpreting $\lim_t \Box_t \Diamond \ p = \Box \Diamond \ p$

Set-theoretic interpretation?

- $[\Box_t \Diamond \ p]_v$ is monotonic (increasing wrt $v \in \mathbb{N}$).
- Its limit exists and is

$$\bigcup_{v \in \mathbb{N}} [\Box_t \Diamond \ p]_v$$
Interpreting $\lim_t \lozenge_t p = \square \lozenge p$

Set-theoretic interpretation?

- $[\square \lozenge_t p]_v$ is monotonic (increasing wrt $v \in \mathbb{N}$).
- Its limit exists and is $\bigcup_{v \in \mathbb{N}} [\square \lozenge_t p]_v$

... but it is not an $\omega$-regular language:

- $\bigcup_{v \in \mathbb{N}} [\square \lozenge_t p]_v$ = “words having (uniformly) upper-bounded subsequences of $\neg p$”

- So $\bigcup_{v \in \mathbb{N}} [\square \lozenge_t p]_v \neq [\square \lozenge p]$. 
Interpreting $\lim_t \Box \Diamond_t p = \Box \Diamond p$

Topological interpretation?

- Work with (topological) closures:
  \[ cl\left( \bigcup_{t \in \mathbb{N}} [\Box \Diamond_t p] \right) = cl([\Box \Diamond p]) = [true] \]

- But also:
  \[ cl\left( \bigcap_{t \in \mathbb{N}} [\Diamond \Box_t p] \right) = cl([\Diamond \Box p]) = [true]? \]

- Also not clear how to generalize to formulas with nested bounded operators (even if the operators have the same “polarity”).
Interpreting $\lim_t \Box_t \Diamond p = \Box \Diamond p$

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

---

**Too coarse metric**

Many interesting probabilities are actually either 0 or 1.
Interpreting $\lim_t \Box \Diamond_t p = \Box \Diamond p$

Probabilistic interpretations?

**Example**

System $S$:
Specifications: $\phi = \Box p$, or more involved $\psi = \text{never 100 times in a row } \neg p = \Box \Diamond <100 p$. 
Our proposal for interpreting $\lim_{t} \square \Diamond_t p = \square \Diamond p$

**Interpretation as entropy**

### Convergence in entropy

\[
\begin{align*}
\lim_{v \to \infty} \mathcal{H}(\left[\square \Diamond_t p\right]_v) &= \lim_{v \to \infty} (|AP| - 2^{-v}) \\
&= |AP| = \mathcal{H}(\left[\square \Diamond p\right]) \\
\lim_{v \to \infty} \mathcal{H}(\left[\Diamond \square_t p\right]_v) &= \lim_{v \to \infty} |AP| \\
&= |AP| = \mathcal{H}(\left[\Diamond \square p\right])
\end{align*}
\]
Our proposal for interpreting $\lim_t □ ◊ p = □ ◊ p$

Interpretation as entropy

**Convergence in entropy**

\[
\lim_{v \to \infty} \mathcal{H}(\lbrack □ ◊ t p \rbrack_v) = \lim_{v \to \infty} (|AP| - 2^{-v}) = |AP| = \mathcal{H}(\lbrack □ ◊ p \rbrack)
\]

\[
\lim_{v \to \infty} \mathcal{H}(\lbrack ◊ □ t p \rbrack_v) = \lim_{v \to \infty} |AP| = |AP| = \mathcal{H}(\lbrack ◊ □ p \rbrack)
\]

But also for all $v$,

\[
\mathcal{H}(\lbrack ◊ t □ p \rbrack_v) = 1 \neq 2 = \mathcal{H}(\lbrack ◊ □ p \rbrack)
\]

**Goal**

We want to decide whether $\lim_v \mathcal{H}(\lbrack φ_t \rbrack_v) = \mathcal{H}(\lbrack φ_∞ \rbrack)$. 
Main result and techniques

Restricting to fragments of PLTL

First, some bad news

For instance: $\square_t p \land \diamond_s \neg p$ admits no entropy limit.

So we restrict our problem to:

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

1. $\text{PLTL}_\Diamond$: PLTL without $R_t$, “positive fragment”.
   
   $\phi ::= p \mid \neg p \mid \Box \phi \mid \phi \land \phi \mid \phi \lor \phi \mid \phi U \phi \mid \phi R \phi \mid \phi U_t \phi$

2. $\text{PLTL}_\Box$: PLTL without $U_t$, “negative fragment”.
   
   $\phi ::= p \mid \neg p \mid \Box \phi \mid \phi \land \phi \mid \phi \lor \phi \mid \phi U \phi \mid \phi R \phi \mid \phi R_t \phi$
Our actual result

Theorem (Main)

Given a formula $\varphi$ in $\text{PLTL}\Diamond$ or $\text{PLTL} \square$,

- $\lim_v H(\llbracket \varphi \rrbracket_v)$ always exists and is computable as the logarithm of an algebraic real number;
- consequently, it is decidable whether $\lim_v H(\llbracket \varphi \rrbracket_v) = H(\llbracket \varphi_\infty \rrbracket)$. 
Main result and techniques

Our actual result

Theorem (Main)

Given a formula \( \varphi \) in PLTL\( \diamond \) or PLTL\( \square \),

- \( \lim_v \mathcal{H}(\llbracket \varphi \rrbracket_v) \) always exists and is computable as the logarithm of an algebraic real number;

- consequently, it is decidable whether \( \lim_v \mathcal{H}(\llbracket \varphi \rrbracket_v) = \mathcal{H}(\llbracket \varphi_\infty \rrbracket) \).

Method for computing \( \lim_v \mathcal{H} \)

1. Build a parameterized Büchi automaton for \( \varphi \).
2. Find its useful part (details depend on PLTL\( \diamond \) or PLTL\( \square \)).
3. Determinize the “limit” automaton, compute its spectral radius, conclude.
Generalized Büchi automata with parameters and counters (BüAPC)

BüAPC \cong \text{discrete timed automaton with parameters}

- $p, q, r \in \text{AP}$
- $c$ 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 $t := \mathbf{v}$
Main result and techniques  Producing entropy in GTBAC

Where is entropy produced in a GTBAC?

We need to compute

\[
\lim_{v \to \infty} H(\mathcal{L}(B, v)) = \lim_{v \to \infty} \limsup_{n \to \infty} \frac{1}{n} \log \#\mathcal{L}_n(B, v)
\]
Where is entropy produced in a GTBAC?

We need to compute

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

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

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

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

\[
\lim_{v \to \infty} \mathcal{H}(\mathcal{L}(B, v)) = \lim_{v \to \infty} \lim_{n \to \infty} \frac{1}{n} \log \# \mathcal{L}_n(B, v)
\]

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 $R_t$ and $U_t$)

- states: consistent sets of subformulas;
- “colours”: obligations to satisfy an $U$ (1 for each occurrence).
- counters: for satisfying $R_t$ and $U_t$ (1 for each occurrence):
  - counters always reset except when relevant
    (i.e. within corresponding $R_t$’s or $U_t$’s scope)
  - upper-bounded guards allow “staying” in the scope of a $U_t$;
  - lower-bounded guards allow “escaping” the scope of a $R_t$. 
Example of construction
Automaton built for $p \lor \Diamond (q U_t r)$

No color because there is no $\mathcal{U}$. All infinite runs are accepting.
PLTL to B"uAPC

Two subclasses of B"uAPC

- $\text{B"uAPC}^+$: all guards are upper bounds $\land_i x_i \leq t_i$
- $\text{B"uAPC}^-$: all guards are lower bounds $\land_i x_i \geq t_i$
Main result and techniques Translating from PLTL to GTBAC

PLTL to BüAPC

Two subclasses of BüAPC

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

Theorem

*For a PLTL formula $\varphi$, we can construct a BüAPC $A$ such that*

- *for any $v \in \mathbb{N}^{Param}$, $[\varphi]_v = L(A, v)$;*
- *if $\varphi$ is in PLTL$\diamond$ then $A$ is a BüAPC⁺;*
- *and if $\varphi$ is in PLTL$\square$ then $A$ is a BüAPC⁻.*
Key result

Theorem

For any $\text{BüAPC}^+$ or $\text{BüAPC}^-$, $B$, the limit entropy $\lim_v \mathcal{H}(\mathcal{L}(B, v))$ exists and can be computed.

...and thus the main theorem (stated before) directly follows: limit entropy of $\text{PLTL}^\Diamond$ and $\text{PLTL}^\Box$ formulas can be computed.
BüAPC+: asymptotic analysis, a single strongly connected component

\(B\): BüAPC+ (guards: \(x < t\), \(v \to \infty\)

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

Proof method

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

\(\Rightarrow\) constraints of \(B_v\) satisfied

\(\Rightarrow\) small increase of length.
BüAPC+: computing the limit entropy

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

Idea of the algorithm

- Find the part of $B$ that resets all counters and is usable in accepting runs (for all $v$).
- Compute its entropy.
BüAPC+: computing the limit entropy

Algorithm

Data: a BüAPC+ \( B \)

Result: \( H = \lim_\nu H(B, \nu) \) as log of an algebraic number

\( \text{SCC} \leftarrow \text{Tarjan}(B); \)

\( \text{SCC}_G \leftarrow \text{set of non-trivial components resetting all counters}; \)

\( \text{SCC}_A \leftarrow \text{set of accepting non-trivial components}; \)

\( B_1 \leftarrow \text{trim}(B, Q_0, \text{SCC}_A \cap \text{SCC}_G); \) /* find useful part */

\( B_2 \leftarrow \text{restrict}(B_1, \text{SCC}_G); \) /* keep good SCCs */

return \( H(L(B_2)). \)

Proposition

For a BüAPC+ \( B \), the algorithm above computes \( H = \lim_\nu H(B, \nu) \).
BüAPC−: asymptotic analysis

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

**Essential object to build**

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

**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 $\square_t p$)**

![Diagram of BüAPC− and BüAPC− automata]

Dashed arrow: a “pumping” transition.
BüAPC−: computing limit entropy

Idea of the algorithm

- Build symbolic automaton $E$
- Compute the entropy of its useful part.

Algorithm

**Data:** a BüAPC− $B$

**Result:** $\lim_{v} \mathcal{H}(\mathcal{L}(B, v))$ as log of an algebraic number

\[
E \leftarrow \text{symbolic}(B);
\]
\[
E_1 \leftarrow \text{trim}(E, Q_0 \times \emptyset, \text{Acc})
\]
\[
E_2 \leftarrow \text{restrict}(E_1, \text{non-pumping transitions})
\]

**return** $\mathcal{H}(\mathcal{L}(E_2))$

Proposition

*For a BüAPC− $B$, the algorithm above computes* $\lim_{v} \mathcal{H}(B, v)$. 
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\(\square\).
- Computing limits of entropies for BüAPC\(\text{+}\) and BüAPC\(\text{−}\).

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\(\bigtriangleup\) and PLTL\(\Box\).
- Computing limits of entropies for BüAPC\(+\) and BüAPC\(\leq\).

Open questions and further work

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

Thank you!