Counterexamples and Probabilistic Model Checking

Counterexamples in general

Finding counterexamples is like bug-hunting. However, unlike usual software engineering practice where bugs are found by extensive testing, model-checking uses formal methods in order to find counterexamples, i.e. formally specified behavior that contradicts system’s formally expressed specification.

Counterexamples in LTL are typically finite execution paths and a good counterexample is a short path.

Counterexamples in CTL are finite trees.

Counterexamples in PCTL

Recall the BNF definition of PCTL WFF:

\[
\phi ::= \bot | \top | a | \neg \phi | \phi_1 \land \phi_2 | P^\preceq p(\psi) \\
\psi ::= \phi_1 U^{\preceq h} \phi_2 | \phi_1 W^{\preceq h} \bot
\]

where $\preceq \in \{<, >, \leq, \geq\}$, $p$ is probability, $p \in [0, 1]$ and $h$ is the length of the paths we are considering, $h \in \mathbb{N} \cup \{\infty\}$. If we are considering paths of all lengths we can omit superscript instead of writing $^{\leq \infty}$. The set of connectives in the above BNF definition is adequate, i.e. we can derive all other connectives from it. That is, other connectives can be considered as “syntactic sugar”. For instance:

\[
P^{\preceq p}(F^{\leq h} \phi) \triangleq P^{\preceq p}(U^{\leq h} \top) \\
P^{\preceq p}(G^{\leq h} \phi) \triangleq P^{\preceq p}(W^{\leq h} \bot)
\]

Some examples of PCTL WFFs:

$P^{\preceq 0.5}(a U b)$ “probability of reaching $b$ state via an $a$-path is $\leq \frac{1}{2}$.”

$P^{\preceq 0.5}(a U^{\leq 10} b)$ “probability of reaching $b$ state via an $a$-path in 10 steps is $\leq \frac{1}{2}$.” We can omit $P^{\preceq 0.5}$ and just write $a U^{\leq 10} b$.
A counterexample in PCTL is a set of paths. If the probability bounds are non-strict, i.e. of the form \( \leq p \) then the counterexamples are finite sets of paths. If probability bounds are strict, i.e. of the form \( < p \) then infinite paths might be needed as a counter example, the analysis for the Markov chain on slide 11 is in the previous lecture.

A minimal counterexample \( C \) is s.t. for any other counterexample \( C' \) we have \( |C| \leq |C'| \). A counterexample \( C \) is smallest (strongest) if it is minimal and for any other minimal counterexample \( C' \) we have \( \Pr(C) \geq \Pr(C') \).

Discrete-Time Markov Chain Example

We can write the probabilities of transitioning from one state to another in a Markov chain as a transition matrix. The transition matrix for Markov chain in the example is:

\[
A = \begin{bmatrix}
  s_0 & s_1 & s_2 & t_1 & t_2 & u \\
  s_0 & 0.6 & 0.3 & 0.1 & & \\
  s_1 & 0.667 & 0.333 & & & \\
  s_2 & 0.2 & 0.5 & 0.3 & & \\
  t_1 & 0.9 & & 0.1 & & \\
  u & 0.2 & 1 & & & \\
  & & & & & \\
\end{bmatrix}
\]

A probability of transitioning from one state to another in 2 steps is \( A^2 \) and in \( n \) steps, \( A^n \).

With some transformations, we can reduce the problem of finding a smallest counterexample to finding a shortest weighted path in a graph. We show the transformations for finding the smallest counterexample to condition \( (\phi U^{\leq 1/2} \psi) \):

**Step1.** We remove some edges from the original Markov chain to obtain a new Markov chain by making all \( \psi \)-states and \((\neg \phi \land \neg \psi)\)-states absorbing states. Absorbing states have only one outgoing edge to self (a loop edge).

**Step2.** Insert a sink state and redirect all outgoing edges of \( \psi \)-states to it.

**Step3.** Turn the Markov chain obtained after Step2. into a weighted directed graph with weights on edges between two states \( w(s, s') \triangleq \log\left(\frac{1}{\Pr(s, s')}\right)\)

Recall that \( \log(ab) = \log(a) + \log(b) \), \( \log(a/b) = \log(a) - \log(b) \). For a finite path \( \pi \triangleq s_0s_1...s_n \),

\[
w(\pi) = \log\left(\frac{1}{\Pr(s_0, s_1)}\right) + ... + \log\left(\frac{1}{\Pr(s_{n-1}, s_n)}\right) = \log\left(\frac{1}{\Pr(s_0, s_1) \cdot ... \cdot \Pr(s_{n-1}, s_n)}\right) = \log\left(\frac{1}{\Pr(\pi)}\right)
\]

and we have that \( \Pr(\pi') \geq \Pr(\pi) \iff w(\pi') \leq w(\pi) \)
A BNF definition of quantified propositional logic (QPL) formula:

\[ \phi ::= \bot | \top | x | \neg \phi_1 \land \phi_2 | \exists x \phi | \forall x \phi \]

For quantified variables we look at the assignments that satisfy formula:
\( \exists x \phi \) is true if either \( \phi[x/\top] \) is true or \( \phi[x/\bot] \) is true.
\( \forall x \phi \) is true if both \( \phi[x/\top] \) is true and \( \phi[x/\bot] \) is true.

An example QPL formula \( \exists a ((\forall b (a \land b)) \lor (\neg a \lor b)) \land (\forall b (a \lor (\neg b \land c))) \) and its parse tree:

As in first-order logic there are binding occurrences and bound occurrences

Quantifier-free QPL is the same as propositional logic.
In closed QPL formula all leaves are bound occurrences.
In prenex form all quantifiers are in the outermost position, i.e. in parse tree quantifiers are at the top.
For every QPL WFF there is one in prenex form.