Part 3: PCTL Model Checking

Mieke Massink
CNR-ISTI, Via Moruzzi 1, Pisa

Probabilistic Model Checking AA. 2020-2021

Model Checking PCTL

Def. Model Checking

Verify whether a state $s$ of a DTMC $D$ satisfies PCTL-formula $\Phi$

Very similar to CTL model checking:

- Calculate in a recursive way the set $\text{Sat}(\Phi)$ of states in $D$ satisfying $\Phi$
- Check if state $s$ is part of $\text{Sat}(\Phi)$

Recursive definition of $\text{Sat}(\Phi)$:

$\text{Sat}(tt) = S$
$\text{Sat}(ff) = \emptyset$
$\text{Sat}(a) = \{ s | a \in L(s) \}$
$\text{Sat}(\neg \Phi) = S \setminus \text{Sat}(\Phi)$
$\text{Sat}(\Phi_1 \lor \Phi_2) = \text{Sat}(\Phi_1) \cup \text{Sat}(\Phi_2)$
$\text{Sat}(P \subseteq p(\Phi)) = \{ s \in S | \text{prob}(s, \Phi) \subseteq p \}$

Model Checking PCTL

For $\phi$ we have three variants:

\begin{align*}
\phi &= X \Phi & \text{next operator} \\
\phi &= \Phi \leq_k \psi & \text{bounded until operator} \\
\phi &= \Phi \leq \psi & \text{unbounded until operator}
\end{align*}

We define $\text{prob}(s, \phi) = \mathbb{P}\{ \sigma \in \text{Paths}(s) | \sigma \models pt \phi \}$

This way we shall write:

$s \in \text{Sat}(P \subseteq p(\Phi)) \iff \text{prob}(s, \Phi) \subseteq p$

Next: $P \subseteq p(\Phi)$

\[ \text{prob}(s, X \Phi) = \sum_{s' \in \text{Sat}(\Phi)} P(s, s') \]

where $P$ is the stochastic matrix.
Model Checking PCTL

Algorithm

\[
prob(s, X \Phi) = \text{prob}(X \Phi) = P \Phi
\]

where \( P \) is the stochastic matrix and \( \Phi \) the vector of states with 1 at the place of the state that satisfies formula \( \Phi \) and 0 in the other places. Note the multiplication on the right of the matrix: this calculates the probability to reach the states indicated by 1 in \( \Phi \) in one step.

Model Checking PCTL

Example

\[
\begin{align*}
prob(s, \{s_1 \cap s_2 \}) &= \text{prob}(s_1 \cap s_2) = \text{prob}(s_1) + \text{prob}(s_2) \\
&= \frac{1}{4} + \frac{1}{4} = \frac{1}{2}
\end{align*}
\]
Model Checking PCTL

Solution: DTMC with four states

\[
\begin{align*}
\text{Sat}(\text{by}) &= \{s\} , \text{Sat}(\neg \text{by}) \cup \{0,2,3\} , \text{Sat}(\text{suu}) \cup \{s\} \\
\text{Sat}(\neg \text{by} \lor \text{suu}) &= \text{Sat}(\neg \text{by}) \lor \text{Sat}(\text{suu}) \cup \{0,2,3\} \\
\begin{bmatrix}
0 & 0 & 0 & 1 \\
0 & 0.99 & 0.99 & 0 \\
0.99 & 0 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} \equiv (\neg \text{by} \lor \text{suu}) &= \begin{bmatrix} 1 \end{bmatrix} \\
\text{Prob}(X \equiv) &= P \equiv = [0, 0.99, 1, 1] \\
\text{Sat}(\text{by} \lor [\text{suu} \lor (\neg \text{by} \lor \text{suu})]) &= \{1, 2, 3\}
\end{align*}
\]

Model Checking PCTL

Model checking algorithm for bounded until

Given DTMC \( D \) and path formula \( \phi = \Phi U^{\leq k} \psi \)

one can observe that:

- \( \phi \) is satisfied if within \( k \) steps a \( \Psi \)-state is reached along a \( \Phi \)-path
- \( \phi \) is violated if one reaches a state that satisfies \( \neg(\Phi \lor \psi) \) along a \( \Phi \)-path within \( k \) steps

Bounded Until: \( \mathcal{P} \leq k \Phi U^{\leq k} \psi \)

For \( k \geq 0 \), \( \text{prob}(s, \Phi U^{\leq k} \psi) \) satisfies the following recurrent equation:

\[
\begin{align*}
\text{prob}(s, \Phi U^{\leq k} \psi) &= k \geq 0, \text{ if } s \in \text{Sat}(\Psi) \text{ then } 1 \\
&\text{ else if not } s \in \text{Sat}(\Phi) \text{ then } 0 \\
&\text{ else } \sum_{s' \in S} P(s, s').\text{prob}(s', \Phi U^{\leq k-1} \psi) \\
&\quad k < 0, \text{ prob}(s, \Phi U^{\leq k} \psi) = 0
\end{align*}
\]

\( \text{prob}(s, \Phi U^{\leq k} \psi) \) is the minimal solution of the above equation.

Problem: it is not a very efficient algorithm.
Example: Triple Modular Redundant System

- Three processors and a voter
- Each processor runs the same program
- the voter takes the majority as output
- Every component may fail
- There is a single repair service

Further assumptions:
- If the voter fails, the system fails
- After repair of the voter the system is "as new"
- The state of the system is a pair (#process, #voter)

Example: Triple Modular Redundant System

Specification of the system behaviour as a DTMC:

Example: Triple Modular Redundant System

Verify the PCTL formula:

\[ P_{\geq 0.4}((up3 \lor up2) U^{\leq 3} down) \]

"The probability that the system works with at least two processors and then goes down within 3 time units is greater than 0.4"

1) Transform the DTMC (steps 1) and 2)):

become absorbing

Example: Triple Modular Redundant System

1) Transform the DTMC (steps 1) and 2)):

become absorbing

2) Compute transient probability for \( k = 3 \)
Model checking Unbounded Until

The algorithm has three steps:

1. Extend the set $S_f$ to a set $S_f^+$ that contains all states from which one cannot reach a state in $S_s$ with positive probability.
2. Extend the set $S_s$ to a set $S_s^+$ with states in $S_i$ for which the probability to reach a state in $S_s$, without passing by a state in $S_f$ is equal to 1.
3. Solve the set of linear equations defined as:

$$\text{prob}(s, \Phi U \Psi) = \begin{cases} 1 & \text{if } s \in S_s^+ \text{ then } 1 \\ 0 & \text{else if } s \in S_s^+ \text{ then } 0 \\ \sum_{s' \in S} P(s, s').\text{prob}(s', \Phi U \Psi) & \text{else} \end{cases}$$

The latter equations can be solved by Gaussian elimination.
Model checking Unbounded Until

Algorithm to compute $S^+_i$

Both algorithms are based on the ‘shortest path’ algorithm by Dijkstra.

It has been proven that $|S_i|$ iterations are sufficient to reach a fixed point.

Example:
Model checking Unbounded Until

Special Cases

- $P_{>0}(\phi U \psi)$ is very similar to CTL form. $\exists (\phi U \psi)$
- $P_{\geq 1}(\phi U \psi)$ is very similar to CTL form. $\forall (\phi U \psi)$

It is possible to use standard CTL Model Checking for these cases because that is more efficient.

!!! BUT !!!:

Only in these special cases and under the condition that **fair** CTL model checking is used.

---

Probabilistically fair

A path $\sigma = s_0, s_1, \ldots$ is probabilistically fair if

iff

for every state $s$ that occurs infinitely often in path $\sigma$ also every successor of $s$ (in the model) occurs infinitely often.

Example:

$$P_{\geq 1}(a U b) ? True \ or \ false?$$
Probabilistic Bisimulation

A probabilistic bisimulation on DTMC $D = (S, P, L)$ is an equivalence $R$ on the state space $S$ such that:

If $(s, s') \in R$ then:

- $L(s) = L(s')$
- $P(s, C) = P(s', C)$ for all classes $C \in S/R$

where

$$P(s, C) = \sum_{s' \in C} P(s, s')$$

Probabilistic Bisimulation

PCTL equivalence corresponds to

Probabilistic Bisimulation