SS 2017
Software Verification
Probabilistic modelling – DTMC extensions / MDP

Prof. Dr. Holger Schlingloff¹,²
Dr. Esteban Pavese¹

(1) Institut für Informatik der Humboldt Universität
(2) Fraunhofer Institut für offene Kommunikationssysteme FOKUS
Recap

- PCTL vs. CTL

- PCTL qualitative properties vs. CTL

- Recall the model checking algorithm for PCTL
One final operation...

- Probabilistic systems allow for one additional verification operation that has no parallel in either CTL or LTL.
- Recall that DTMCs:
  - have no deadlock
  - every transition is probabilistic
- ... so it makes sense to ask what the behaviour of a DTMC is at *infinite* time.
Steady-State Probabilities

- The crucial question is, given a DTMC that has been running for a long time, what is the probability of the process being in a given state?

- Some applications...
  - A load-balancing server. Which load balancing strategy will likely be selected a month from now? A year from now?
  - How stressed will be my system (i.e., is it more likely to be in a waiting state or busy state?)
Steady-State Probabilities

• Let’s define a very simple DTMC that models whether days are rainy or not

• By doing some *very advanced* meteorological analysis, we discovered that
  - if today is sunny, the likelihood that tomorrow is also sunny is of 85%
  - if today is rainy, there is a 60% chance tomorrow is also rainy

• Today is sunny. What is the chance it will also be sunny next Tuesday?
Steady-State Probabilities

- Recall our X operator in PCTL
- Which states satisfy $P_{\neg p}X (\text{sunny})$?

\[
\begin{bmatrix}
\text{sunny} & \text{rainy} \\
0.85 & 0.15 \\
0.40 & 0.60
\end{bmatrix}
\begin{bmatrix}
1 \\
0
\end{bmatrix}
= 
\begin{bmatrix}
0.85 \\
0.40
\end{bmatrix}
\]
Steady-State Probabilities

- Recall our X operator in PCTL
- What about being sunny two days from now?

\[
\begin{pmatrix}
\text{sunny} & \text{rainy} \\
0.85 & 0.15 \\
0.40 & 0.60
\end{pmatrix}
\begin{pmatrix} 1 \\ 0 \end{pmatrix}
= \begin{pmatrix}
0.7825 \\
0.58
\end{pmatrix}
\]
Steady-State probabilities

- OK. Let’s go the other way now
- Today is sunny. What’s the chance tomorrow is sunny as well?
- We know of course it is 0.85. But matrix-wise

\[
\begin{pmatrix}
1 & 0 \\
0.40 & 0.60
\end{pmatrix}
\begin{pmatrix}
0.85 & 0.15 \\
0.15 & 0.85
\end{pmatrix}
= 
\begin{pmatrix}
0.85 & 0.15
\end{pmatrix}
\]
Steady-State probabilities

• Today is sunny. What’s the chance tomorrow is sunny as well? – 0.85
• What’s the chance it is sunny two days from now?
• Similarly to the X case ...

\[
\begin{pmatrix}
1 & 0 \\
0.40 & 0.60
\end{pmatrix}
\begin{pmatrix}
0.85 & 0.15 \\
0.40 & 0.60
\end{pmatrix}^2
= \begin{pmatrix}
0.7825 & 0.2175
\end{pmatrix}
\]
Steady-State probabilities

• It is the same way we calculated the probability of satisfying a bounded until property

• Now we can calculate the probabilities of being in a certain state
  ▪ assuming some initial condition

• In general

\[ v_{init} P^{time} = v_{@time} \]
Steady-State probabilities

- Again with the same calculations:

\[ v_{@2} = (0.7825 \ 0.2175) \]
\[ v_{@5} = (0.7323 \ 0.2677) \]
\[ v_{@10} = (0.7273 \ 0.2727) \]
\[ v_{@20} = (0.7272 \ 0.2728) \]
\[ v_{@30} = (0.7272 \ 0.2728) \]
Steady-State probabilities

- It seems to converge! In particular, if today is sunny, the chance of it being a rainy day a long time from now approximates 0.2728
  - of course, actual meteorological models are much more complex
  - don’t take this as a guarantee to anything

- Is there a way then to calculate \( v_{@\infty} \)?
Steady-State probabilities

• The steady state vector $v_{@\infty}$ must satisfy the property that $v_{@\infty}P = v_{@\infty}$

\[
v_{\infty}P = v_{\infty}
\]

\[
v_{\infty}P - v_{\infty} = 0
\]

\[
v_{\infty}(P - I) = 0
\]

• Now $v_{@\infty}$ encodes the unknowns of this system
Steady-State probabilities

- Back to the example

\[
\begin{pmatrix}
    v_s & v_r
\end{pmatrix}
\begin{pmatrix}
    0.85 & 0.15 \\
    0.40 & 0.60
\end{pmatrix}
- \begin{pmatrix}
    1 & 0 \\
    0 & 1
\end{pmatrix}
= \begin{pmatrix}
    v_s & v_r
\end{pmatrix}
\]

\[
\begin{pmatrix}
    v_s & v_r
\end{pmatrix}
\begin{pmatrix}
    -0.15 & 0.15 \\
    0.40 & -0.40
\end{pmatrix}
= \begin{pmatrix}
    v_s & v_r
\end{pmatrix}
\]

- \(-0.15v_s + 0.40v_r = 0\)
- \(0.15v_s - 0.40v_r = 0\)

- hm, it’s the same equation. Infinite solutions?
Steady-State probabilities

• NO! The steady state vector must be stochastic!
  ▪ that is, the probabilities need to add up to 1

\[ v_s + v_r = 1 \]

\[ 0.15v_s - 0.40v_r = 0 \]

• solving for the vector yields

\[ v_s = 0.727272... \]

\[ v_r = 0.272727... \]
Steady-State probabilities

- So, it works! It gives the same result we had seen before

- One funny thing though...

- Our previous result was based on knowing *today was sunny*

- turns out it doesn’t matter *at all*. Given enough time, it all evens out
One final note:

- Finite DTMCs always have a steady state vector (a limiting distribution)
- But infinite ones may not (e.g. random walk)
- Conditions for its existence are known
  - but outside the scope of the lesson
  - key words are irreducibility and periodicity
Markov Decision Processes

- DTMCs are useful, but they introduce problems for modelling
- Namely, the absence of action labels and non-determinism does not allow for accurate description of synchronising systems
- Markov Decision Processes (MDPs) allow non-determinism and have a parallel composition semantics
Markov Decision Processes

- The main differences with DTMCs are that
  - MDPs have a finite action set $A$
  - The transition relation is now
    \[ R : S \times A \rightarrow \mathcal{D}(S) \]
    - that is, given a state \textit{and an action}, we get the target distribution
  - Many distributions possible from a state
  - How to choose the appropriate one?
    - Nondeterministic choice, schedulers
Markov Decision Processes

- Applications
  - as already said, compositional modelling
  - but also, modelling complete uncertainty
    - vs. probabilistic uncertainty
  - multiple distributions for a same action
    - e.g. probabilistic “modes”
  - probabilistic algorithms / models, but with explicit scheduling uncertainty
    - e.g. the OS
MDP Model Checking

- PCTL still is a valid logic to reason about MDPs
- However the semantics are slightly different
  - and the model checking algorithm reflects these differences

- Consider for example the X (next) operator
What is the probability of getting heads in the next step from state $s_1$?
MDP model checking

If $b$ is taken = 0.0
If $c$ is taken = 0.5
MDP model checking

- Non-deterministic choices need to be resolved by an adversary or scheduler.
- Given a finite path, the adversary chooses the next action resolving nondeterminism.
- Adding an adversary to an MDP yields a DTMC:
  - where all transitions are probabilistic.
  - ...but this DTMC may be infinite!
- Special case: 0-memory adversaries:
  - these induce a finite DTMC.
MDP model checking

- For each adversary choice
  - a DTMC is chosen
  - that yields a unique probability
- Therefore a property may have infinitely many different satisfaction probabilities
- But at least one adversary will provide the minimum probability of satisfaction
  - and another (or maybe the same) the maximum
- Model checking entails an optimization problem