Probabilistic Model Checking (1)
Lecture #1 of GLOBAN Summerschool

Joost-Pieter Katoen
Software Modeling and Verification Group
affiliated to University of Twente, Formal Methods and Tools

Warsaw University, September 24, 2008
Model checking

- Automated model-based verification and debugging technique
  - model of system = Kripke structure ≈ labeled transition system
  - properties expressed in temporal logic like LTL or CTL
  - provides counterexamples in case of property refutation

- Various striking examples
  - Needham-Schroeder protocol, cache coherence, storm surge barrier, C code

- 2008: Pioneers awarded prestigious ACM Turing Award

- Today: model checking of probabilistic models
Principles of Model Checking

CHRISTEL BAIER
TU Dresden, Germany

JOOST-PiETER KATOEN
RWTH Aachen University, Germany, and
University of Twente, the Netherlands

“This book offers one of the most comprehensive introductions to logic model checking techniques available today. The authors have found a way to explain both basic concepts and foundational theory thoroughly and in crystal clear prose. Highly recommended for anyone who wants to learn about this important new field, or brush up on their knowledge of the current state of the art.”

(Gerard J. Holzmann, NASA JPL, Pasadena)
Content of this lecture

• Introduction
  – why probabilities?, history, tools + applications

• Markov chains
  – paths, measurability, reachability probabilities

• Probabilistic CTL
  – syntax, semantics, model checking, PCTL versus CTL

• Abstraction
  – bisimulation, correctness, minimization
Content of this lecture

⇒ Introduction
  – why probabilities?, history, tools + applications

• Markov chains
  – paths, measurability, reachability probabilities

• Probabilistic CTL
  – syntax, semantics, model checking, PCTL versus CTL

• Abstraction
  – bisimulation, correctness, minimization
**Probabilities help**

- **When analysing system performance and dependability**
  - to quantify arrivals, waiting times, time between failure, QoS, ...

- **When modelling uncertainty in the environment**
  - to quantify imprecisions in system inputs
  - to quantify unpredictable delays, express soft deadlines, ...

- **When building protocols for networked embedded systems**
  - randomized algorithms

- **When certain problems are undecidable deterministically**
  - reachability in communicating finite-state machines
### Probabilistic models

<table>
<thead>
<tr>
<th></th>
<th>Nondeterminism</th>
<th>Nondeterminism</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td><strong>Discrete time</strong></td>
<td>discrete-time</td>
<td>Markov decision</td>
</tr>
<tr>
<td></td>
<td>Markov chain (DTMC)</td>
<td>process (MDP)</td>
</tr>
<tr>
<td><strong>Continuous time</strong></td>
<td>CTMC</td>
<td>CTMDP</td>
</tr>
</tbody>
</table>
Breakthroughs

- **Zero-one probabilities for Markov decision processes** (Vardi 1985)
  - does an LTL formula hold with probability zero?

- **Markov decision processes** (Courcoubetis & Yannakakis 1988)
  - does the maximal probability for an LTL formula equal $p$?

- **Discrete-time Markov chains** (Hansson & Jonsson 1990)
  - does the probability of a CTL formula equal $p$?

- **Markov decision processes** (Bianco & de Alfaro 1995)
  - does the maximal probability for a CTL formula equal $p$?

- **Continuous-time Markov chains** (Baier, Katoen & Hermanns 1999)
  - does the probability of a timed CTL formula equal $p$?
## Reachability probabilities

<table>
<thead>
<tr>
<th></th>
<th>Nondeterminism no</th>
<th>Nondeterminism yes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reachability</strong></td>
<td>linear equation system DTMC</td>
<td>linear programming MDP</td>
</tr>
<tr>
<td><strong>Timed reachability</strong></td>
<td>transient analysis (+ uniformization) CTMC</td>
<td>greedy backward reachability uniform CTMDP</td>
</tr>
</tbody>
</table>
What is probabilistic model checking?

requirements → inaccuracy → system

Formalizing → property specification → system model

Model Checking → satisfied → insufficient memory

\[ P \leq 0.01(\diamond \text{deadlock}) \]

\[ P \leq 0.01 \]

Model Checking

the probability

up to \(10^7\) states

state 1 0.678
state 2 0.9797
state 3 0.1523
state 4 0.2123

© JPK
Characteristics

- **What is inside?**
  - temporal logics and model checking
  - numerical and optimisation techniques from performance and OR

- **What can be checked?**
  - time-bounded reachability, long-run averages, safety and liveness

- **What is its usage?**
  - powerful tools: PRISM (4,000 downloads), MRMC, Petri net tools, Probmela
  - applications: distributed systems, biology, avionics, ...
Probability elsewhere

- **In performance modelling**  
  - models: typically continuous-time Markov chains  
  - emphasis on steady-state and transient measures  
  (Erlang, 1907)

- **In stochastic control theory and operations research**  
  - models: typically discrete-time Markov decision models  
  - emphasis on finding optimal policies for average measures  
  (Bellman, 1957)

- **Our focus: model checking Markov chains**
  - temporal logic $\Rightarrow$ unambiguous and precise *measure-specification*
  - model-checking techniques $\Rightarrow$ no expert algorithmic knowledge needed
  - complex (new) measures are concisely specified and *automatically verified*
  - exchanging techniques with the other two areas
Illustrating examples

- **Security: Crowds protocol**
  - analysis of probability of anonymity

- **IEEE 1394 Firewire protocol**
  - proof that biased delay is optimal

- **Systems biology**
  - probability that enzymes are absent within the deadline

- **Software in next generation of satellites**
  - mission time probability (ESA project)
A synchronous leader election protocol

[Itai & Rodeh, 1990]

- A round-based protocol in a synchronous ring of $N > 2$ nodes
  - the nodes proceed in a lock-step fashion
  - each slot = 1 message is read + 1 state change + 1 message is sent
  $\Rightarrow$ this synchronous computation yields a Markov chain

- Each round starts by each node choosing a uniform id $\in \{1, \ldots, K\}$

- Nodes pass their selected id around the ring

- If there is a unique id, the node with the maximum unique id is leader

- If not, start another round and try again ...
Leader election

probabilistically choose an id from $[1 \ldots K]$
Leader election

send your selected id to your neighbour
Leader election

pass the received id, and check uniqueness own id
Leader election

pass the received id, and check uniqueness own id
Leader election

pass the received id, and check uniqueness own id
End of 1st round

no unique leader has been elected
Start a new round

new round and new chances!
Properties of leader election

- Almost surely eventually a leader will be elected:
  \[ P_{\geq 1}(\Diamond \text{ leader elected}) \]

- With probability \( \geq \frac{4}{5} \), eventually a leader is elected:
  \[ P_{\geq 0.8}(\Diamond \text{ leader elected}) \]

- \( \ldots \) within \( k \) steps:
  \[ P_{\geq 0.8}(\Diamond \leq^k \text{ leader elected}) \]
Probability to elect a leader within $L$ rounds

\[ P \leq q \leq (N+1) \cdot L \quad \text{leader elected} \quad \text{(Itai & Rodeh's algorithm)} \]
Content of this lecture

• Introduction
  – why probabilities?, history, tools + applications

⇒ Markov chains
  – paths, measurability, reachability probabilities

• Probabilistic CTL
  – syntax, semantics, model checking, PCTL versus CTL

• Abstraction
  – bisimulation, correctness, minimization
Discrete-time Markov chains

A DTMC $\mathcal{M}$ is a tuple $(S, P, \nu_{\text{init}}, AP, L)$ with:

- $S$ is a countable nonempty set of states
- $P : S \times S \rightarrow [0, 1]$, transition probability function s.t. $\sum_{s'} P(s, s') = 1$
  - $P(s, s')$ is the probability to jump from $s$ to $s'$ in one step
- $\nu_{\text{init}} : S \rightarrow [0, 1]$, the initial distribution with $\sum_{s \in S} \nu_{\text{init}}(s) = 1$
  - $\nu_{\text{init}}(s)$ is the probability that system starts in state $s$
  - state $s$ for which $\nu_{\text{init}}(s) > 0$ is an initial state
- $L : S \rightarrow 2^{AP}$, the labelling function

$\Rightarrow$ a DTMC is a transition system with probabilistic transitions
Craps
Craps

- Roll two dice and bet on outcome

- Come-out roll ("pass line" wager):
  - outcome 7 or 11: win
  - outcome 2, 3, or 12: loss ("craps")
  - any other outcome: roll again (outcome is "point")

- Repeat until 7 or the "point" is thrown:
  - outcome 7: loss ("seven-out")
  - outcome the point: win
  - any other outcome: roll again
A DTMC model of Craps

- **Come-out roll:**
  - 7 or 11: win
  - 2, 3, or 12: loss
  - else: roll again

- **Next roll(s):**
  - 7: loss
  - point: win
  - else: roll again
Paths

- **State graph** of DTMC $\mathcal{M}$
  - vertices are states of $\mathcal{M}$, and $(s, s')$ is an edge iff $P(s, s') > 0$

- **Paths** in $\mathcal{M}$ are maximal (i.e., infinite) paths in its state graph
  - $Paths(\mathcal{M})$ and $Paths_{\text{fin}}(\mathcal{M})$ denote the set of (finite) paths in $\mathcal{M}$

- $Post(s) = \{ s' \in S \mid P(s, s') > 0 \}$ and $Pre(s) = \{ s' \in S \mid P(s', s) > 0 \}$
  - $Post^*(s)$ is the set of states reachable from $s$ via a finite path fragment
  - $Pre^*(s) = \{ s' \in S \mid s \in Post^*(s') \}$
Probability measure on DTMCs

- Events are *infinite paths* in the DTMC $\mathcal{M}$, i.e., $\Omega = \text{Paths}(\mathcal{M})$

- $\sigma$-algebra on $\mathcal{M}$ is generated by *cylinder sets* of finite paths $\hat{\pi}$:
  \[
  Cyl(\hat{\pi}) = \{ \pi \in \text{Paths}(\mathcal{M}) \mid \hat{\pi} \text{ is a prefix of } \pi \}
  \]
  - cylinder sets serve as basis events of the smallest $\sigma$-algebra on $\text{Paths}(\mathcal{M})$

- $\text{Pr}$ is the *probability measure* on the $\sigma$-algebra on $\text{Paths}(\mathcal{M})$:
  \[
  \text{Pr}(Cyl(s_0 \ldots s_n)) = \iota_{\text{init}}(s_0) \cdot P(s_0 \ldots s_n)
  \]
  - where $P(s_0 s_1 \ldots s_n) = \prod_{0 \leq i < n} P(s_i, s_{i+1})$ and $P(s_0) = 1$
Reachability probabilities

- What is the probability to reach a set of states \( B \subseteq S \) in DTMC \( \mathcal{M} \)?

- Which event does \( \Diamond B \) mean formally?
  - the union of all cylinders \( \text{Cyl}(s_0 \ldots s_n) \) where
  - \( s_0 \ldots s_n \) is an initial path fragment in \( \mathcal{M} \) with \( s_0, \ldots, s_{n-1} \notin B \) and \( s_n \in B \)

\[
\Pr(\Diamond B) = \sum_{s_0 \ldots s_n \in \text{Paths}_{\text{fin}}(\mathcal{M}) \cap (S \setminus B)^* B} \Pr(\text{Cyl}(s_0 \ldots s_n))
\]

\[
= \sum_{s_0 \ldots s_n \in \text{Paths}_{\text{fin}}(\mathcal{M}) \cap (S \setminus B)^* B} \lambda_{\text{init}}(s_0) \cdot P(s_0 \ldots s_n)
\]
Reachability probabilities in finite DTMCs

• Let \( \Pr(s \models \diamond B) = \Pr_s(\diamond B) = \Pr_s\{\pi \in \text{Paths}(s) \mid \pi \models \diamond B\} \)
  - where \( \Pr_s \) is the probability measure in \( \mathcal{M} \) with single initial state \( s \)

• Let variable \( x_s = \Pr(s \models \diamond B) \) for any state \( s \)
  - if \( B \) is not reachable from \( s \) then \( x_s = 0 \)
  - if \( s \in B \) then \( x_s = 1 \)

• For any state \( s \in \text{Pre}^*(B) \setminus B \):

\[
x_s = \sum_{t \in S \setminus B} P(s, t) \cdot x_t + \sum_{u \in B} P(s, u)
\]

- \( \sum_{t \in S \setminus B} P(s, t) \cdot x_t \) reach \( B \) via \( t \)
- \( \sum_{u \in B} P(s, u) \) reach \( B \) in one step
Remark: expansion law

- Recall in CTL: $\exists(C \cup B)$ is the least solution of expansion law:

$$\exists(C \cup B) \equiv B \lor (C \land \exists \land \exists(C \cup B))$$

- That is: the set $X = \text{Sat}(\exists(C \cup B))$ is the smallest set such that:

$$B \cup \{ s \in C \setminus B \mid \text{Post}(s) \cap X \neq \emptyset \} \subseteq X$$

- Previous slide “replaces” $s \in X$ by values $x_s$ in $[0, 1]$
  - if $s \in B$ then $x_s = 1$ (compare: $s \in B$ implies $s \in X$)
  - if $s \in S \setminus (C \cup B)$ then $x_s = 0$ (compare: $s \notin C \cup B$ implies $s \notin X$)

- If $s \in C \setminus B$ then $x_s = \sum_{t \in C \setminus B} P(s, t) \cdot x_t + \sum_{t \in B} P(s, t)$
  - compare: $s \in C \setminus B$ and $\text{Post}(s) \cap X \neq \emptyset$ implies $s \in X$
Linear equation system

- These equations can be rewritten into the following form:

\[ \mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{b} \]

  - where vector \( \mathbf{x} = (x_s)_{s \in \tilde{S}} \) with \( \tilde{S} = \text{Pre}^*(\mathbf{B}) \setminus \mathbf{B} \)
  - \( \mathbf{A} = \left( \mathbf{P}(s, t) \right)_{s, t \in \tilde{S}} \), the transition probabilities in \( \tilde{S} \)
  - \( \mathbf{b} = \left( b_s \right)_{s \in \tilde{S}} \) contains the probabilities to reach \( \mathbf{B} \) within one step

- **Linear equation system**: \((\mathbf{I} - \mathbf{A})\mathbf{x} = \mathbf{b}\)
  
  - note: more than one solution may exist if \( \mathbf{I} - \mathbf{A} \) has no inverse (i.e., is singular)
  
  \( \Rightarrow \) characterize the desired probability as least fixed point
Unique solution

Let $\mathcal{M}$ be a finite DTMC with state space $S$ partitioned into:

- $S_{=} = \text{Sat}(-\exists (C \cup B))$
- $S_{\neq} = \text{Sat}((\neg \exists (C \cup B))$ a subset of $\{s \in S \mid \Pr(s \models C \cup B) = 1\}$ that contains $B$
- $S_{\neq} = S \setminus (S_{=} \cup S_{\neq})$

The vector $\left( \Pr(s \models C \cup B) \right)_{s \in S_{\neq}}$ is the unique solution of the linear equation system:

$$\mathbf{x} = A \mathbf{x} + \mathbf{b} \quad \text{where} \quad A = \left( P(s, t) \right)_{s, t \in S_{\neq}} \quad \text{and} \quad b = \left( P(s, S_{=} \cup S_{\neq}) \right)_{s \in S_{\neq}}$$
Computing reachability probabilities

- The probabilities of the events $C \cup \leq^n B$ can be obtained iteratively:

  $$x^{(0)} = 0 \quad \text{and} \quad x^{(i+1)} = Ax^{(i)} + b \text{ for } 0 \leq i < n$$

- where $A = \left( P(s, t) \right)_{s, t \in C \setminus B}$ and $b = \left( P(s, B) \right)_{s \in C \setminus B}$

- Then: $x^{(n)}(s) = \Pr(s \models C \cup \leq^B B) \text{ for } s \in C \setminus B$
Example: Craps game

- $\Pr(start | C \cup \leq_n B)$
- $S_{=0} = \{8, 9, 10, lost\}$
- $S_{=1} = \{won\}$
- $S？ = \{start, 4, 5, 6\}$
Example: Craps game

• $\text{start} < 4 < 5 < 6$

• $A = \frac{1}{36} \begin{pmatrix} 0 & 3 & 4 & 5 \\ 0 & 27 & 0 & 0 \\ 0 & 0 & 26 & 0 \\ 0 & 0 & 0 & 25 \end{pmatrix}$

• $b = \frac{1}{36} \begin{pmatrix} 8 \\ 3 \\ 4 \\ 5 \end{pmatrix}$

$x^{(0)} = 0$ and $x^{(i+1)} = Ax^{(i)} + b$ for $0 \leq i < n$. 
Example: Craps game

\[
\mathbf{x}^{(2)} = \frac{1}{36} \left( \begin{array}{cccc}
0 & 3 & 4 & 5 \\
0 & 27 & 0 & 0 \\
0 & 0 & 26 & 0 \\
0 & 0 & 0 & 25 \\
\end{array} \right) \cdot \frac{1}{36} \left( \begin{array}{c}
8 \\
3 \\
4 \\
5 \\
\end{array} \right) + \frac{1}{36} \left( \begin{array}{c}
8 \\
3 \\
4 \\
5 \\
\end{array} \right) = \left( \frac{1}{36} \right)^2 \left( \begin{array}{c}
338 \\
189 \\
248 \\
305 \\
\end{array} \right)
\]
Content of this lecture

• Introduction
  – why probabilities?, history, tools + applications

• Markov chains
  – paths, measurability, reachability probabilities

⇒ Probabilistic CTL
  – syntax, semantics, model checking, PCTL versus CTL

• Abstraction
  – bisimulation, correctness, minimization
PCTL Syntax

- For $a \in AP$, $J \subseteq [0, 1]$ an interval with rational bounds, and natural $n$:

$$\Phi ::= \text{true} \mid a \mid \Phi \land \Phi \mid \neg \Phi \mid \mathbb{P}_J(\varphi)$$

$$\varphi ::= \bigcirc \Phi \mid \Phi_1 \cup \Phi_2 \mid \Phi_1 \cup^{\leq n} \Phi_2$$

- $s_0s_1s_2\ldots \models \Phi \cup^{\leq n} \Psi$ if $\Phi$ holds until $\Psi$ holds within $n$ steps

- $s \models \mathbb{P}_J(\varphi)$ if probability that paths starting in $s$ fulfill $\varphi$ lies in $J$

abbreviate $\mathbb{P}_{[0,0.5]}(\varphi)$ by $\mathbb{P}_{\leq 0.5}(\varphi)$ and $\mathbb{P}_{[0,1]}(\varphi)$ by $\mathbb{P}_{>0}(\varphi)$ and so on
Derived operators

\[ \diamond \Phi = \text{true} \cup \Phi \]

\[ \diamond \leq_n \Phi = \text{true} \cup \leq_n \Phi \]

\[ \mathbb{P}_{\leq p}(\Box \Phi) = \mathbb{P}_{1-p}(\diamond \neg \Phi) \]

\[ \mathbb{P}_{p,q}(\square \leq_n \Phi) = \mathbb{P}_{1-q,1-p}(\diamond \leq_n \neg \Phi) \]

operators like weak until \( W \) or release \( R \) can be derived analogously
Example properties

- With probability $\geq 0.92$, a goal state is reached via legal ones:

  $$\mathbb{P} \geq 0.92 \left( \neg \text{illegal } U \text{ goal} \right)$$

- ... in maximally 137 steps:

  $$\mathbb{P} \geq 0.92 \left( \neg \text{illegal } U \leq 137 \text{ goal} \right)$$

- ... once there, remain there almost surely for the next 31 steps:

  $$\mathbb{P} \geq 0.92 \left( \neg \text{illegal } U \leq 137 \mathbb{P} = 1 (\square_{[0,31]} \text{ goal}) \right)$$

© JPK
PCTL semantics (1)

\( M, s \models \Phi \) if and only if formula \( \Phi \) holds in state \( s \) of DTMC \( M \)

Relation \( \models \) is defined by:

- \( s \models a \) iff \( a \in L(s) \)
- \( s \models \neg \Phi \) iff not \( (s \models \Phi) \)
- \( s \models \Phi \lor \Psi \) iff \( (s \models \Phi) \) or \( (s \models \Psi) \)
- \( s \models \mathbb{P}_J(\varphi) \) iff \( \Pr(s \models \varphi) \in J \)

where \( \Pr(s \models \varphi) = \Pr_s\{\pi \in \text{Paths}(s) \mid \pi \models \varphi\} \)
PCTL semantics (2)

A path in $\mathcal{M}$ is an infinite sequence $s_0 \ s_1 \ s_2 \ldots$ with $P(s_i, s_{i+1}) > 0$

Semantics of path-formulas is defined as in CTL:

$$\pi \models \Box \Phi \quad \text{iff} \quad s_1 \models \Phi$$

$$\pi \models \Phi \cup \Psi \quad \text{iff} \quad \exists n \geq 0.( s_n \models \Psi \land \forall 0 \leq i < n. s_i \models \Phi )$$

$$\pi \models \Phi \cup \leq n \Psi \quad \text{iff} \quad \exists k \geq 0. ( k \leq n \land s_k \models \Psi \land \forall 0 \leq i < k. s_i \models \Phi )$$
Measurability

For any PCTL path formula $\varphi$ and state $s$ of DTMC $\mathcal{M}$, the set $\{ \pi \in \text{Paths}(s) \mid \pi \models \varphi \}$ is measurable.
PCTL model checking

• Given a finite DTMC $\mathcal{M}$ and PCTL formula $\Phi$, how to check $\mathcal{M} \models \Phi$?

• Check whether state $s$ in a DTMC satisfies a PCTL formula:
  – compute \textit{recursively} the set $\text{Sat}(\Phi)$ of states that satisfy $\Phi$
  – check whether state $s$ belongs to $\text{Sat}(\Phi)$
  \Rightarrow \text{bottom-up traversal} of the parse tree of $\Phi$ (like for CTL)

• For the propositional fragment: as for CTL

• \textbf{How to compute $\text{Sat}(\Phi)$ for the probabilistic operators?}
PCTL model checking

- Alternative formulation: $s \models \mathbb{P}^J(\Diamond \Phi)$ if and only if $\Pr(s \models \Diamond \Phi) \in J$

- Next: $\Pr(s \models \Diamond \Phi)$ equals $\sum_{s' \in \text{Sat}(\Phi)} P(s, s')$

- Matrix-vector multiplication:

\[
(\Pr(s \models \Diamond \Phi))_{s \in S} = P \cdot \iota_\Phi
\]

where $\iota_\Phi$ is the characteristic vector of $\text{Sat}(\Phi)$, i.e., $\iota_\Phi(s) = 1$ iff $s \in \text{Sat}(\Phi)$
Checking probabilistic reachability

- $s \models \mathbb{P}_J(\Phi \cup^h \Psi)$ if and only if $\Pr(s \models \Phi \cup^h \Psi) \in J$

- $\Pr(s \models \Phi \cup^h \Psi)$ is the least solution of: (Hansson & Jonsson, 1990)
  
  - 1 if $s \models \Psi$
  
  - for $h > 0$ and $s \models \Phi \land \neg \Psi$:
    $$\sum_{s' \in S} P(s, s') \cdot \Pr(s' \models \Phi \cup^{h-1} \Psi)$$

  - 0 otherwise

- Standard reachability for $\mathbb{P}^>0(\Phi \cup^h \Psi)$ and $\mathbb{P}^\geq1(\Phi \cup^h \Psi)$
  
  - for efficiency reasons (avoiding solving system of linear equations)
Reduction to transient analysis

- Make all $\Psi$- and all $\neg (\Phi \lor \Psi)$-states absorbing in $\mathcal{M}$

- Check $\Diamond^{=h} \Psi$ in the obtained DTMC $\mathcal{M}'$

- This is a standard transient analysis in $\mathcal{M}'$:

$$\sum_{s' \models \Psi} \Pr_{s} \{ \pi \in \text{Paths}(s) \mid \sigma[h] = s' \}$$

- compute by $(P')^h \cdot \iota_{\Psi}$ where $\iota_{\Psi}$ is the characteristic vector of $\text{Sat}(\Psi)$

$\Rightarrow$ Matrix-vector multiplication
Time complexity

For finite DTMC $\mathcal{M}$ and PCTL formula $\Phi$, $\mathcal{M} \models \Phi$ can be solved in time

$$O\left(poly(|\mathcal{M}|) \cdot n_{\text{max}} \cdot |\Phi|\right)$$

where $n_{\text{max}} = \max\{ n \mid \Psi_1 \cup^{\leq n} \Psi_2 \text{ occurs in } \Phi \}$ with $\max \emptyset = 1$
Verification times

command-line tool MRMC ran on a Pentium 4, 2.66 GHz, 1 GB RAM laptop
The qualitative fragment of PCTL

- For $a \in AP$:

$$
\Phi ::= \text{true} \mid a \mid \Phi \land \Phi \mid \neg \Phi \mid P_{>0}(\varphi) \mid P_{=}1(\varphi)
$$

$$
\varphi ::= \Box \Phi \mid \Phi_1 U \Phi_2
$$

- The probability bounds $= 0$ and $< 1$ can be derived:

$$
P_{=}0(\varphi) \equiv \neg P_{>0}(\varphi) \quad \text{and} \quad P_{<1}(\varphi) \equiv \neg P_{=}1(\varphi)
$$

- No bounded until, and only $> 0$, $= 0$, $> 1$ and $= 1$ intervals

so: $P_{=}1(\Diamond P_{>0}(\Box a))$ and $P_{<1}(P_{>0}(\Diamond a) U b)$ are qualitative PCTL formulas
Qualitative PCTL = CTL?

- PCTL-formula $\Phi$ is *equivalent* to CTL-formula $\Psi$:
  - $\Phi \equiv \Psi$ if and only if $Sat_M(\Phi) = Sat_{TS(M)}(\Psi)$ for each DTMC $M$

- $\exists \varphi$ requires $\varphi$ on *some* paths, $P > 0(\varphi)$ with *positive* probability
  - $P > 0(\bigcirc a) \equiv \exists \bigcirc a$ and $P > 0(\Diamond a) \equiv \exists \Diamond a$ and $P > 0(a \cup b) \equiv \exists a \cup b$

- $\forall \varphi$ requires $\varphi$ to hold for *all* paths, $P = 1(\varphi)$ for *almost* all
  - $P = 1(\bigcirc a) \equiv \forall \bigcirc a$ and $P = 1(\Box a) \equiv \forall \Box a$

- **But:** $P > 0(\varphi) \equiv \exists \varphi$ and $P = 1(\varphi) \equiv \forall \varphi$ do not hold in general!
Qualitative PCTL versus CTL

- There is no CTL-formula that is equivalent to $\mathbb{P}_{\geq 1}(\Diamond a)$
- There is no CTL-formula that is equivalent to $\mathbb{P}_{>0}(\Box a)$
- There is no qualitative PCTL-formula that is equivalent to $\forall \Diamond a$
- There is no qualitative PCTL-formula that is equivalent to $\exists \Box a$

$\Rightarrow$ PCTL with $\forall \varphi$ and $\exists \varphi$ is more expressive than PCTL
Content of this lecture

- **Introduction**
  - why probabilities?, history, tools + applications

- **Markov chains**
  - paths, measurability, reachability probabilities

- **Probabilistic CTL**
  - syntax, semantics, model checking, PCTL versus CTL

⇒ Abstraction
  - bisimulation, correctness, minimization
Probabilistic bisimulation: intuition

- Strong bisimulation is used to compare labeled transition systems
- Strongly bisimilar states exhibit the same step-wise behaviour
- We like to adapt bisimulation to DTMCs
- This yields a probabilistic variant of strong bisimulation
- When do two DTMC states exhibit the same step-wise behaviour?
- Key: if their transition probability for each equivalence class coincides

_for simplicity, assume a unique initial state_
Probabilistic bisimulation

Let $\mathcal{M} = (S, P, AP, L)$ be a DTMC and $R \subseteq S \times S$ an equivalence

$R$ is a probabilistic bisimulation on $S$ if for any $(s, s') \in R$:

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

where $P(s, C) = \sum_{s' \in C} P(s, s')$ [Larsen & Skou, 1989]

$s \sim s'$ if $\exists$ a probabilistic bisimulation $R$ with $(s, s') \in R$
Example
Quotient DTMC under $\sim$

$\mathcal{M}/\sim = (S', P', AP, L')$, the quotient of $\mathcal{M} = (S, P, AP, L)$ under $\sim$:

- $S' = S/\sim = \{ [s]_{\sim} \mid s \in S \}$
- $P'([s]_{\sim}, C) = P(s, C)$
- $L'([s]_{\sim}) = L(s)$

get $\mathcal{M}/\sim$ by partition-refinement in time $O(M \cdot \log N + |AP| \cdot N)$ [Derisavi et al., 2001]
A DTMC model of Craps
Minimizing Craps

initial partitioning for the atomic propositions $AP = \{ \text{loss} \}$
A first refinement

refine (“split”) with respect to the set of red states
A second refinement

refine ("split") with respect to the set of green states
Quotient DTMC
Preservation of PCTL

\[ s \sim s' \iff (\forall \Phi \in PCTL : s \models \Phi \text{ if and only if } s' \models \Phi) \]
IEEE 802.11 group communication protocol

<table>
<thead>
<tr>
<th>OD</th>
<th>original CTMC</th>
<th>lumped CTMC</th>
<th>red. factor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>states</td>
<td>transitions</td>
<td>ver. time</td>
</tr>
<tr>
<td>4</td>
<td>1125</td>
<td>5369</td>
<td>121.9</td>
</tr>
<tr>
<td>12</td>
<td>37349</td>
<td>236313</td>
<td>7180</td>
</tr>
<tr>
<td>20</td>
<td>231525</td>
<td>1590329</td>
<td>50133</td>
</tr>
<tr>
<td>28</td>
<td>804837</td>
<td>5750873</td>
<td>195086</td>
</tr>
<tr>
<td>36</td>
<td>2076773</td>
<td>15187833</td>
<td>5103900</td>
</tr>
<tr>
<td>40</td>
<td>3101445</td>
<td>22871849</td>
<td>7725041</td>
</tr>
</tbody>
</table>
Weak probabilistic bisimulation

- Let $\mathcal{M} = (S, P, AP, L)$ be a DTMC and $R \subseteq S \times S$ an equivalence

- $R$ is a weak probabilistic bisimulation on $S$ if for any $(s_1, s_2) \in R$:
  - $L(s_1) = L(s_2)$
  - $s_1$ can reach a state outside $[s_1]_R$ iff $s_2$ can do so
  - if $P(s_i, [s_i]_R) < 1$ for $i=1, 2$ then:
    \[
    \frac{P(s_1, C)}{1 - P(s_1, [s_1]_R)} = \frac{P(s_2, C)}{1 - P(s_2, [s_2]_R)}
    \]
    for all $C \in S/R, C \neq [s_1]_R$

- $s \approx s'$ if $\exists$ a weak probabilistic bisimulation $R$ with $(s, s') \in R$
Logical characterization

\[ s \approx s' \iff (\forall \Phi \in PCTL \setminus \Box : s \models \Phi \text{ if and only if } s' \models \Phi) \]
Probabilistic simulation

- For transition systems, state $s'$ simulates state $s$ if
  - for each successor $t$ of $s$ there is a one-step successor $t'$ of $s'$ that simulates $t$

  $\Rightarrow$ simulation of two states is defined in terms of simulation of successor states

- What are successor states in the probabilistic setting?
  - the target of a transition is in fact a probability distribution

  $\Rightarrow$ the simulation relation $\sqsubseteq$ needs to be lifted from states to distributions
Weight function $\Delta$

- $\Delta$ "distributes" a distribution $\mu$ over set $X$ to one $\mu'$ over set $Y$
  - such that the total probability assigned by $\Delta$ to $y \in Y$ equals the original probability $\mu'(y)$ on $Y$
  - and symmetrically for the total probability mass of $x \in X$ assigned by $\Delta$

- $\Delta$ is a distribution on $R \subseteq X \times Y$ such that:
  - the probability to select $(x, y)$ with $(x, y) \in R$ is one, and
  - the probability to select $(x, \cdot) \in R$ equals $\mu(x)$, and
  - the probability to select $(\cdot, y) \in R$ equals $\mu'(y)$
Weight function

• Let $R \subseteq S \times S$, and $\mu, \mu' \in Distr(S)$

• $\Delta \in Distr(S \times S)$ is a weight function for $(\mu, \mu')$ and $R$ whenever:

$$\Delta(s, s') > 0 \text{ implies } (s, s') \in R \quad \text{and}$$

$$\mu(s) = \sum_{s' \in S} \Delta(s, s') \quad \text{and} \quad \mu'(s') = \sum_{s \in S} \Delta(s, s') \text{ for any } s, s' \in S$$

• $\mu \sqsubseteq_R \mu'$ iff there exists a weight function for $(\mu, \mu')$ and $R$
Weight function example
Probabilistic simulation

- Let $\mathcal{M} = (S, P, AP, L)$ be a DTMC and $R \subseteq S \times S$

- $R$ is a \textit{probabilistic simulation} on $S$ if for all $(s, s') \in R$:
  
  $$L(s) = L(s') \quad \text{and} \quad P(s, \cdot) \sqsubseteq_R P(s', \cdot)$$

- $s \sqsubseteq_p s'$ if there exists a probabilistic simulation $R$ with $(s, s') \in R$
Probabilistic simulation example

\[ R = \{ (s_1, s_2), (s, u), (t, u), (t, v), (w_1, w_2), (w_1, w_3) \} \]

is a probabilistic simulation (cf. weight function before)
Simulation equivalence = bisimulation

For any DTMC:
probabilistic simulation equivalence
coincides with
probabilistic bisimulation

does only hold for deterministic labeled transition systems
Logical characterization

\[ s \sqsubseteq s' \iff (\forall \Phi \in \text{safePCTL} : s' \models \Phi \text{ implies } s \models \Phi) \]

The syntax of the safe fragment of PCTL is given by:

\[ \Phi ::= \text{true} \mid a \mid \neg a \mid \Phi \land \Phi \mid \Phi \lor \Phi \mid P_{\geq p}(\Phi W \Phi) \mid P_{\geq p}(\Phi W \trianglelefteq n \Phi) \]

A typical safe PCTL formula: \( P_{\geq 0.99}(\square \leq 100 \neg \text{error}) \)

© JPK
## Overview

<table>
<thead>
<tr>
<th>Logical preservation</th>
<th>strong bisimulation $\sim$</th>
<th>weak bisimulation $\approx$</th>
<th>strong simulation $\sqsubseteq$</th>
<th>weak simulation $\approx$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCTL</td>
<td>PCTL $\setminus \Box$</td>
<td>safePCTL</td>
<td>safePCTL $\setminus \Box$</td>
<td></td>
</tr>
<tr>
<td>checking equivalence</td>
<td>partition refinement $O(m \log n)$</td>
<td>partition refinement $O(n^3)$</td>
<td>parametric maximal flow problem $O(m^2 \cdot n)$</td>
<td>parametric maximal flow problem $O(m^2 \cdot n^3)$</td>
</tr>
<tr>
<td>graph minimization</td>
<td>$O(m \log n)$</td>
<td>$O(n^3)$</td>
<td>$-$</td>
<td>$-$</td>
</tr>
</tbody>
</table>

$O(m \log n) = O(n^3) = O(m^2 \cdot n)$
Thank you for the attendance

谢谢大家！