An Introduction to Probabilistic Model Checking

Pedro R. D’Argenio
Dependable Systems Group – FaMAF
Universidad Nacional de Córdoba
CONICET

http://dsg.famaf.unc.edu.ar

11º ERPEM, Rio Cuarto, Dec-2015
Overview

- Motivation
- Reachability analysis on deterministic models
- Reachability analysis on non-deterministic models
- LTL
- The process of probabilistic model checking
- Quick and partial overview of the state of the art
Why verification?

Ariane 5:
64 bits fp vs 16 bits int

Pentium:
FDIV

Mars Climate Orbiter:
Métrico vs Imperial

Therac-25:
Race condition

Northeast blackout in 2003:
Race condition

Heartbleed:
Integridad/Confidencialidad
Model Checking

Properties are either true or false

\[ G \left( send(msg) \Rightarrow F \text{ rcv}(msg) \right) \]

Non-deterministic behavior
Limitations of this approach

- Many algorithms proposed *(better) solutions using randomization.*
- E.g.
  - Leader election protocol in IEEE 1394 “Firewire”
  - Binary exponential backoff on IEEE 802.3 “Ethernet”
Limitations of this approach

E.g.: IEEE 1394 Leader election protocol
Limitations of this approach

E.g.: IEEE 1394 Leader election protocol
Limitations of this approach

E.g.: IEEE 1394 Leader election protocol
Limitations of this approach

E.g.: IEEE 1394 Leader election protocol

Root contention!
Limitations of this approach

E.g.: IEEE 1394 Leader election protocol

It is solved by “flipping coins”
Limitations of this approach

Many times, correction cannot be established in a usual bivalued (modal) logic.

Nevertheless, the validity of a property can be quantified through a probability value.

E.g.

- Bounded Retransmission Protocol en Philips RC6
- Binary Exponential Backoff Algorithm en IEEE 802.3 “Ethernet”
Limitations of this approach

Suppose that a file is transmitted using the ABP or a sliding window protocol

\[ G (\text{send}(\text{msg}) \Rightarrow \text{F } \text{rcv}(\text{msg}) ) \]

but this is under the assumption that an infinite number of retrials is allowed.
Limitations of this approach

Suppose that a file is transmitted using the ABP or a sliding window protocol

\[ G \ ( \ send(msg) \Rightarrow F \ rcv(msg) ) \]

What if only a bounded number of retransmissions is allowed? (e.g. BRP)
Limitations of this approach

Properties are either true or false

$G ( \text{send(msg)} \Rightarrow F \text{rcv(msg)} )$

Non-deterministic behavior
Limitations of this approach

\[ G \left( \text{send(msg)} \implies F \text{rcv(msg)} \right) \]

- Non-deterministic behavior
- Probabilistic behavior should also be considered
- The truth value should be probabilistically quantified
Fully probabilistic systems (Markov Chain)

\( (S, P, s_0, L) \)

- \( S \) is the set of states with initial state \( s_0 \)
- \( P : S \times S \rightarrow [0, 1] \) is the probabilistic transition function, s.t. \( \forall s \in S, \sum_{s' \in S} P(s, s') = 1 \), and
- \( L : S \rightarrow \mathcal{P}(AP) \) labelling function, where \( AP \) is the set of atomic propositions.

\[ S = \{s_0, s_1, s_2, s_3\} \]

\[ P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{10} & \frac{9}{10} \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \]

\[ L(s_0) = \{\text{start}\} \]
\[ L(s_1) = \{\text{try}\} \]
\[ L(s_2) = \{\text{lost}\} \]
\[ L(s_3) = \{\text{delivered}\} \]
Models contain probabilistic information (e.g. a decision made by tossing a coin, the probability of losing a message).

The validity of a temporal formula (e.g. LTL) is quantified with a probability value in [0,1] (instead of a boolean).

\[
\text{Prob}( F \circ ) = \ 0.5 \times 0.4 + 0.5 \times 0.2 + 0.5 \times 0.7 = 0.65
\]
Example 10.3. Simulating a Die by a Fair Coin

Consider simulating the behavior of a standard six-sided die by a fair coin, as originally proposed by Knuth and Yao [242], see the Markov chain depicted in Figure 10.2.

The computation starts in the initial state $s_0$, i.e., we have $\pi_{\text{init}}(s_0) = 1$ and $\pi_{\text{init}}(s) = 0$ for all states $s \neq s_0$. The states 1, 2, 3, 4, 5, and 6 at the bottom stand for the possible die outcomes. Each inner node stands for tossing a fair coin. If the outcome is heads, the left branch determines the next state; if the outcome is tails, the right branch determines the next state.

If the coin-tossing experiment in state $s_0$ yields heads, the system moves to state $s_1$, $s_2$, $s_3$. Tossing the coin again leads with equal probability to either state $s_2$, $s_3$ (from which the die-outcomes 2 or 3 are possible with equal probability) or to state $s_4$, $s_5$. From the latter state, a coin flipping yields with probability $\frac{1}{2}$ the outcome 1, or with probability $\frac{1}{2}$ a return to state $s_1$, $s_2$, $s_3$. The behavior for outcome tails in the initial state is symmetric. We will establish later that, in fact, this Markov chain indeed adequately models a die, i.e., the outcomes are equally likely.

Example 10.4. The Craps Gambling Game

The game craps is based on betting on the outcome of the roll of two dice. The outcome of the first roll—the "come-out" roll—determines whether there is a need for any further rolls. On outcome 7 or 11, the game is over and the player wins. The outcomes 2, 3, or 12, however, are "craps"; the player loses. On any other outcome, the dice are rolled again, but the outcome of the come-out roll is remembered (the "point"). If the next roll yields Probability of a property

\[ P(F2) \]
Example 10.3. Simulating a Die by a Fair Coin

Consider simulating the behavior of a standard six-sided die by a fair coin, as originally proposed by Knuth and Yao [242], see the Markov chain depicted in Figure 10.2.

The computation starts in the initial state \( s_0 \), i.e., \( \pi(0) = 1 \) and \( \pi(s) = 0 \) for all states \( s \neq s_0 \). The states 1, 2, 3, 4, 5, and 6 at the bottom stand for the possible die outcomes. Each inner node stands for tossing a fair coin. If the outcome is heads, the left branch determines the next state; if the outcome is tails, the right branch determines the next state.

If the coin-tossing experiment in state \( s_0 \) yields heads, the system moves to state \( s_1, s_2, s_3 \). Tossing the coin again leads with equal probability to either state \( s_2, s_3 \) (from which the die-outcomes 2 or 3 are possible with equal probability) or to state \( s'_1, s_2, s_3 \). From the latter state, a coin flipping yields with probability \( \frac{1}{2} \) the outcome 1, or with probability \( \frac{1}{2} \) a return to state \( s_1, s_2, s_3 \). The behavior for outcome tails in the initial state is symmetric. We will establish later that, in fact, this Markov chain indeed adequately models a die, i.e., the outcomes are equally likely.

Example 10.4. The Craps Gambling Game

The game craps is based on betting on the outcome of the roll of two dice. The outcome of the first roll—the "come-out" roll—determines whether there is a need for any further rolls. On outcome 7 or 11, the game is over and the player wins. The outcomes 2, 3, or 12, however, are "craps"; the player loses. On any other outcome, the dice are rolled again, but the outcome of the come-out roll is remembered (the "point"). If the next roll yields the point, the player wins; if it yields 7, the player loses.
Example 10.3. Simulating a Die by a Fair Coin

Consider simulating the behavior of a standard six-sided die by a fair coin, as originally proposed by Knuth and Yao [242], see the Markov chain depicted in Figure 10.2.

The computation starts in the initial state $s_0$, i.e., we have $\pi(s_0) = 1$ and $\pi(s) = 0$ for all states $s \neq s_0$. The states $1, 2, 3, 4, 5, 6$ at the bottom stand for the possible die outcomes. Each inner node stands for tossing a fair coin. If the outcome is heads, the left branch determines the next state; if the outcome is tails, the right branch determines the next state.

If the coin-tossing experiment in state $s_0$ yields heads, the system moves to state $s_1, s_2, s_3$. Tossing the coin again leads with equal probability to either state $s_2, s_3$ (from which the die-outcomes 2 or 3 are possible with equal probability) or to state $s_1'. s_2, s_3'. From the later state, a coin flipping yields with probability $\frac{1}{2}$ the outcome 1, or with probability $\frac{1}{2}$ a return to state $s_1, s_2, s_3$.

The behavior for outcome tails in the initial state is symmetric. We will establish later that, in fact, this Markov chain indeed adequately models a die, i.e., the outcomes are equally likely.

Example 10.4. The Craps Gambling Game

The game craps is based on betting on the outcome of the roll of two dice. The outcome of the first roll—the "come-out" roll—determines whether there is a need for any further rolls. On outcome 7 or 11, the game is over and the player wins. The outcomes 2, 3, or 12, however, are "craps"; the player loses. On any other outcome, the dice are rolled again, but the outcome of the come-out roll is remembered (the "point"). If the next roll yields the point, the player wins; otherwise, the dice are rolled again.

Probability of a property

\[
P(s_0 s_1 s_4 2) + P(s_0 s_1 s_3 s_1 s_4 2) + \cdots
\]

\[
\frac{1}{8} + \frac{1}{32} + \frac{1}{128} + \frac{1}{512} + \cdots
\]
Example 10.3. Simulating a Die by a Fair Coin

Consider simulating the behavior of a standard six-sided die by a fair coin, as originally proposed by Knuth and Yao [242], see the Markov chain depicted in Figure 10.2.

The computation starts in the initial state $s_0$, i.e., we have $\pi_{\text{init}}(s_0) = 1$ and $\pi_{\text{init}}(s) = 0$ for all states $s \neq s_0$. The states 1, 2, 3, 4, 5, and 6 at the bottom stand for the possible die outcomes. Each inner node stands for tossing a fair coin. If the outcome is heads, the left branch determines the next state; if the outcome is tails, the right branch determines the next state.

If the coin-tossing experiment in state $s_0$ yields heads, the system moves to state $s_1, s_2, s_3$. Tossing the coin again leads with equal probability to either state $s_2, s_3$ (from which the die-outcomes 2 or 3 are possible with equal probability) or to state $s_4'$. From the latter state, a coin flipping yields with probability $\frac{1}{2}$ the outcome 1, or with probability $\frac{1}{2}$ a return to state $s_1, s_2, s_3$. The behavior for outcome tails in the initial state is symmetric. We will establish later that, in fact, this Markov chain indeed adequately models a die, i.e., the outcomes are equally likely.

Example 10.4. The Craps Gambling Game

The game craps is based on betting on the outcome of the roll of two dice. The outcome of the first roll—the "come-out" roll—determines whether there is a need for any further rolls. On outcome 7 or 11, the game is over and the player wins. The outcomes 2, 3, or 12, however, are "craps"; the player loses. On any other outcome, the dice are rolled again, but the outcome of the come-out roll is remembered (the "point"). If the next roll yields the point, the player wins. Otherwise, the player loses. The probability of winning depends on the point.

Probability of a property

$$P_{s_0}(F_{2}) = \sum_{n>0} P(s_0 s_1 (s_3 s_1)^n s_4 2) = \sum_{n>0} \frac{1}{2^{2n+1}} = \frac{1}{6}$$
Probabilistic Model Checking in fully probabilistic models

Example 10.3. Simulating a Die by a Fair Coin

Consider simulating the behavior of a standard six-sided die by a fair coin, as originally proposed by Knuth and Yao [242], see the Markov chain depicted in Figure 10.2.

The computation starts in the initial state \( s_0 \), i.e., we have \( \text{init}(s_0) = 1 \) and \( \text{init}(s) = 0 \) for all states \( s \neq s_0 \). The states 1, 2, 3, 4, 5, and 6 at the bottom stand for the possible die outcomes. Each inner node stands for tossing a fair coin. If the outcome is heads, the left branch determines the next state; if the outcome is tails, the right branch determines the next state.

If the coin-tossing experiment in state \( s_0 \) yields heads, the system moves to state \( s_1, s_2, s_3 \). Tossing the coin again leads with equal probability to either state \( s_2, s_3 \) (from which the die-outcomes 2 or 3 are possible with equal probability) or to state \( s'_1, s_2, s_3 \). From the latter state, a coin flipping yields with probability \( \frac{1}{2} \) the outcome 1, or with probability \( \frac{1}{2} \) a return to state \( s_1, s_2, s_3 \). The behavior for outcome tails in the initial state is symmetric. We will establish later that, in fact, this Markov chain indeed adequately models a die, i.e., the outcomes are equally likely.

Example 10.4. The Craps Gambling Game

The game craps is based on betting on the outcome of the roll of two dice. The outcome of the first roll—the "come-out" roll—determines whether there is a need for any further rolls. On outcome 7 or 11, the game is over and the player wins. The outcomes 2, 3, or 12, however, are "craps"; the player loses. On any other outcome, the dice are rolled again, but the outcome of the come-out roll is remembered (the "point"). If the next roll yields 7 or 11, the player wins; otherwise, the point is rolled again.

Using DFS, we can calculate whether 2 is reachable with probability 0.

\[
\begin{align*}
P_{s_2}(F \ 2) &= P_{s_5}(F \ 2) = P_{s_6}(F \ 2) = 0 \\
P_1(F \ 2) &= P_3(F \ 2) = P_4(F \ 2) = 0 \\
P_5(F \ 2) &= P_6(F \ 2) = 0 \\
P_2(F \ 2) &= 1 \\
P_{s_0}(F \ 2) &= \frac{1}{2} \ P_{s_1}(F \ 2) + \frac{1}{2} \ P_{s_2}(F \ 2) \\
P_{s_1}(F \ 2) &= \frac{1}{2} \ P_{s_3}(F \ 2) + \frac{1}{2} \ P_{s_4}(F \ 2) \\
P_{s_3}(F \ 2) &= \frac{1}{2} \ P_{s_1}(F \ 2) + \frac{1}{2} \ P_1(F \ 2) \\
P_{s_4}(F \ 2) &= \frac{1}{2} \ P_2(F \ 2) + \frac{1}{2} \ P_3(F \ 2)
\end{align*}
\]
Probabilistic Model Checking in fully probabilistic models

In general:

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

- \[ x_s = 1 \] if \( s \in Pr^>^0(B) \setminus B \)
- \[ x_s = 0 \] if \( s \notin Pr^>^0(B) \)

It is solved with standard numeric techniques (Jacobi, Gauss-Seidel)

\( B \) is the set of goal states

The set of states that reach \( B \) with some probability
The need of non-determinism

Parallel composition / Distributed components
- probabilities within a single component are easy to estimate,
- relative probabilities of events located geographically distant depend on a highly unpredictable global state.

Underspecification
- some probabilities are unknown at early stage of modeling.

Abstraction
- models are abstract representations of the system under study.

Control synthesis
- intentional underspecification to synthesize optimal decisions.
Probability of a property

To calculate probabilities in this setting, non-determinism has to be resolved.

Schedulers are functions that select the next transition according to the past execution.
Probability of a property

To calculate probabilities in this setting, non-determinism has to be resolved.

Schedulers are functions that select the next transition according to the past execution.

A scheduler constructs a fully probabilistic tree

(There are also randomized variants)
An LTL formula has associated two values:

- **The maximum** probability under all schedulers
  \[ P_{\text{max}}( F \Diamond ) = 0.96 \]

- **The minimum** probability under all schedulers
  \[ P_{\text{min}}( F \Diamond ) = 0.65 \]
Probability of a property

An LTL formula has associated two values:

- The maximum probability under all schedulers
  \[ P_{\text{max}}(F \circ ) = 0.96 \]

- The minimum probability under all schedulers
  \[ P_{\text{min}}(F \circ ) = 0.65 \]
An LTL formula has associated two values:

- The maximum probability under all schedulers
  \[ P_{\text{max}}( F \rightarrow ) = 0.96 \]
- The minimum probability under all schedulers
  \[ P_{\text{min}}( F \rightarrow ) = 0.65 \]
An LTL formula has associated two values:

- The maximum probability under all schedulers
  \[ P_{\text{max}}(F \cdot) = 0.96 \]
- The minimum probability under all schedulers
  \[ P_{\text{min}}(F \cdot) = 0.65 \]

Randomized and deterministic schedulers are equally expressive for max/min prob. of reach. properties.
Markov decision processes

The structure is as before, only that we have a family of matrices, one for each possible decision.
Markov decision processes

What is the maximum probability of obtaining the desired amount of money?
Model checking
Markov decision processes

$P^+_s$ is a shorthand for $P^\max_s(F\ F a l)$

$P^+_l = P^+_l = 0$

$P^+_u = 1$
Model checking
Markov decision processes

\[ P^+_s \text{ is a shorthand for } P^\max_s(F \text{ al}) \]

\[ P^+_l = P^+_l = 0 \]

\[ P^+_a l = 1 \]

\[ P^+_1 = 0.7P^+_1 + 0.2P^+_2 + 0.1P^+_l \]
Model checking
Markov decision processes

\[ P_s^+ \text{ is a shorthand for } P_s^{\max}(F \text{ al}) \]

\[
\begin{align*}
P_{l_a}^+ &= P_{l_c}^+ = 0 \\
P_{al}^+ &= 1 \\
P_1^+ &= 0.3P_1^+ + 0.2P_8^+ + 0.5P_{l_c}^+
\end{align*}
\]
Model checking
Markov decision processes

$P^+_s$ is a shorthand for $P^+_{s \max}(F \text{ al})$

$P^+_{l_s} = P^+_{l_c} = 0$

$P^+_{a l} = 1$

$P^+_1 = \max \left( 0.7P^+_1 + 0.2P^+_2 + 0.1P^+_l, \ 0.3P^+_1 + 0.2P^+_8 + 0.5P^+_l \right)$
Model checking
Markov decision processes

$P_s^+$ is a shorthand for $P_{max}(F\, a\, l)$

$P_{l_s}^+ = P_{l_c}^+ = 0$

$P_{al}^+ = 1$

$P_1^+ = \max (0.7P_1^+ + 0.2P_2^+ + 0.1P_{l_s}^+, 0.3P_1^+ + 0.2P_8^+ + 0.5P_{l_c}^+)$

$P_2^+ = \max (0.55P_2^+ + 0.25P_4^+ + 0.1P_1^+ + 0.1P_{l_s}^+, 0.3P_2^+ + 0.2P_{al}^+ + 0.5P_{l_c}^+)$

$P_4^+ = \max (0.55P_4^+ + 0.25P_8^+ + 0.1P_2^+ + 0.1P_{l_s}^+, 0.3P_4^+ + 0.2P_{al}^+ + 0.5P_{l_c}^+)$

$P_8^+ = \max (0.55P_8^+ + 0.25P_{al}^+ + 0.1P_4^+ + 0.1P_{l_s}^+, 0.3P_8^+ + 0.2P_{al}^+ + 0.5P_{l_c}^+)$
Model checking
Markov decision processes

In general:

\[ x_s = \max_{a \in A} \sum_{t \in S} P_a(s, t) \cdot x_t \]

\[ x_s = 1 \quad \text{if } s \in Pr^{>0}(B) \setminus B \]

\[ x_s = 0 \quad \text{if } s \notin Pr^{>0}(B) \]

\[ x_s = 1 \quad \text{if } s \in B \]

Linear optimization problem.
Solved with standard numerical analysis techniques.

B is the set of goal states.
The set of states that may reach B with some probability.
LTL reduced to reachability

LTL = propositional logic + temporal modalities:

- $G \varphi$ : “$\varphi$ holds globally”
- $F \varphi$ : “Finally $\varphi$ holds”
- $\varphi U \psi$ : “$\varphi$ holds until $\psi$ holds”

E.g.:

$$G \left( \text{send-msg } \Rightarrow \ F \text{ rcv-msg } \right)$$
LTL reduced to reachability

Every LTL formula can be translated to a Büchi Automaton that represents the accepting behaviour.
Example 10.4. The Craps Gambling Game

The game craps is based on betting on the outcome of the roll of two dice. The outcome of the first roll—the "come-out" roll—determines whether there is a need for any further rolls. On outcome 7 or 11, the game is over and the player wins. The outcomes 2, 3, or 12, however, are "craps"; the player loses. On any other outcome, the dice are rolled again, but the outcome of the come-out roll is remembered (the "point"). If the next roll yields 7, the player loses. If the next roll yields the point, then the player wins. A roll of 2, 3, or 12, or 7, is called a "natural." If any other outcome is rolled, the next roll determines the winner.

Example 10.3. Simulating a Die by a Fair Coin

Consider simulating the behavior of a standard six-sided die by a fair coin, as originally proposed by Knuth and Yao [242], see the Markov chain depicted in Figure 10.2.

\[
\begin{align*}
\mathbf{P}_S(\phi) &= \text{?} \\
\phi &= \Box \Diamond \text{crit}_1 \land \Box \Diamond \text{crit}_2
\end{align*}
\]

dtmc
module die
   s : [0..7] init 0;
d : [0..6] init 0;
[] s=0 -> 0.5 : (s'=1) + 0.5 : (s'=2);
[] s=1 -> 0.5 : (s'=3) + 0.5 : (s'=4);
[] s=2 -> 0.5 : (s'=5) + 0.5 : (s'=6);
[] s=3 -> 0.5 : (s'=1) + 0.5 : (s'=7) & (d'=1);
[] s=4 -> 0.5 : (s'=7) & (d'=2) + 0.5 : (s'=7) & (d'=3);
[] s=5 -> 0.5 : (s'=7) & (d'=4) + 0.5 : (s'=7) & (d'=5);
[] s=6 -> 0.5 : (s'=2) + 0.5 : (s'=7) & (d'=6);
[] s=7 -> (s'=7);
endmodule

Compose \( M_S \) with \( A_\phi \)

Calculate probability of reaching accepting BSCCs in \( M_S \times A_\phi \)
Example 10.4. The Craps Gambling Game proposed by Knuth and Yao [242], see the Markov chain depicted in Figure 10.2.

Tossing the coin again leads with equal probability to either state $1$ or $2$. The behavioral outcome $1$ results if the first toss is tails. The states $1$, $2$, $3$, $4$, $5$, and $6$ at the bottom stand for the possible die-outcomes $1$, $2$, $3$, $4$, $5$, and $6$. The state $0$ is the absorbing state, and $0.5$ is the probability of ending the game, i.e., for a heads, the leftmost state is reached.

The states $1$, $3$, $4$, and $6$ are accepting BSCCs in $M_S$.

The module $S$ of accepting BSCCs in $M_S$ is defined as:

\[
\begin{align*}
 s &: [0..7] \ INIT \ 0; \\
 d &: [0..6] \ INIT \ 0; \\
 s = 0 &\rightarrow 0.5 \cdot (s' = 1) + 0.5 \cdot (s' = 2); \\
 s = 1 &\rightarrow 0.5 \cdot (s' = 3) + 0.5 \cdot (s' = 4); \\
 s = 2 &\rightarrow 0.5 \cdot (s' = 5) + 0.5 \cdot (s' = 6); \\
 s = 3 &\rightarrow 0.5 \cdot (s' = 1) + 0.5 \cdot (s' = 7) \land (d' = 1); \\
 s = 4 &\rightarrow 0.5 \cdot (s' = 7) \land (d' = 2) + 0.5 \cdot (s' = 7) \land (d' = 3); \\
 s = 5 &\rightarrow 0.5 \cdot (s' = 7) \land (d' = 4) + 0.5 \cdot (s' = 7) \land (d' = 5); \\
 s = 6 &\rightarrow 0.5 \cdot (s' = 7) \land (d' = 6). \\
\end{align*}
\]

\[P_S(\phi) = ?\] is modeled.

Calculate probability of reaching accepting BSCCs in $M_S \times A_\phi$. The correctness condition $\phi$ is:

[Diagram of $A_\phi$]
Highlights on Fundamentals of Probabilistic Model Checking

- **Vardi ’85**
  - Qualitative MC on deterministic and non-deterministic PTSs

- **Courcoubetis & Yanakakis ’88**
  - Quantitative MC on non-deterministic PTSs using LTL and lower/upper bounds

- **Hansson & Jonsson ’90**
  - Quantitative MC on deterministic PTSs introducing PCTL

- **Bianco & de Alfaro ’95**
  - Quantitative MC on non-deterministic PTSs using PCTL*

- **de Alfaro, Kwiatkowska, Norman, Parker, & Segala ’2000**
  - Symbolic quantitative MC on non-deterministic PTSs
Highlights on Fundamentals of Probabilistic Model Checking

- **Vardi ’85**
  - Qualitative MC on deterministic and non-deterministic PTSs

- **Courcoubetis & Yanakakis ’88**
  - Quantitative MC on non-deterministic PTSs using LTL and lower/upper bounds

- **Hansson & Jonsson ’90**
  - Quantitative MC on deterministic PTSs introducing PCTL

- **Bianco & de Alfaro ’95**
  - Quantitative MC on non-deterministic PTSs using PCTL*

- **de Alfaro, Kwiatkowska, Norman, Parker, & Segala ’2000**
  - Symbolic quantitative MC on non-deterministic PTSs

1st. algorithm to qualitative MC MDPs
1st. algorithm for probabilistic MC
1st. modalities with probabilities
1st. “clever” algorithm
1st. efficient tool: PRISM
... and more

- Model Checking Rewards properties
  [Andova, Hermanns & Katoen 2003]

- Model Checking CTMC & steady state properties
  [Baier, Havenkort, Hermanns & Katoen 2002]

- Model Checking CTMDP
  [Baier, Hermanns, Katoen & Havenkort 2004 / Baier, Hahn, Havenkort, Hermanns & Katoen 2013]

- Counterexample derivation
... and more

- Attacking the state explosion problem
  - Abstraction techniques
  - Partial order reduction

- and much more:
  - Controller synthesis and games
  - Partial observation & distributed schedulers
  - Statistical Model Checking
An Introduction to Probabilistic Model Checking

Pedro R. D’Argenio
Dependable Systems Group – FaMAF
Universidad Nacional de Córdoba
CONICET

http://dsg.famaf.unc.edu.ar

11º ERPREM, Rio Cuarto, Dec-2015