VESTA: A Statistical Model-checker and Analyzer for Probabilistic Systems

Authors:
Koushik Sen       Mahesh Viswanathan      Gul Agha
University of Illinois at Urbana-Champaign

YoungMin Kwon

Authors:
Koushik Sen       Mahesh Viswanathan      Gul Agha
University of Illinois at Urbana-Champaign
Vesta Tool

- Input: A probabilistic model $M$ given as
  - a Java class on which one can perform discrete-event simulation
  - a CTMC model in a special language (similar to that used in PRISM)
  - a Probabilistic Rewrite Theory in Maude
Vesta Tool

- **Input:** A probabilistic model $M$ given as
  - a Java class on which one can perform discrete-event simulation
  - a CTMC model in a special language (similar to that used in PRISM)
  - a Probabilistic Rewrite Theory in Maude

- **Input:** A formula $F$ in Continuous Stochastic Logic (CSL) or Probabilistic Computation Tree Logic (PCTL)
  - Vesta can model check $F$ against $M$, i.e. check if $M \models F$
Vesta Tool

- Input: A probabilistic model \( M \) given as
  - a Java class on which one can perform discrete-event simulation
  - a CTMC model in a special language (similar to that used in PRISM)
  - a Probabilistic Rewrite Theory in Maude

- Input: A formula \( F \) in Continuous Stochastic Logic (CSL) or Probabilistic Computation Tree Logic (PCTL)
  - Vesta can model check \( F \) against \( M \), i.e. check if \( M \models F \)

- Input: An expression \( E \) in Quantitative Temporal Expressions (QuaTEx)
  - Vesta can compute the expected value of \( E \)
Model Assumption

- Sample execution paths can be generated through discrete-event simulation
  - Execution paths are sequences of the form
    \[ \pi = s_0 \rightarrow^{t_0} s_1 \rightarrow^{t_1} s_2 \rightarrow^{t_2} \ldots \]
    where each \( s_i \) is a state of the model and \( t_i \in \mathbb{R}_{>0} \) is the time spent in the state \( s_i \) before moving to the state \( s_{i+1} \)

- A probability space can be defined on the execution paths of the model in such a way that the paths satisfying any path formula in our concerned logic (CSL or PCTL), is measurable
Continuous Stochastic Logic (CSL) and PCTL

- $\phi ::= \text{true} \mid a \mid \phi \land \phi \mid \neg \phi \mid P_{\triangleright p}(\psi)$
- $\psi ::= \phi \mathop{U}^{<t} \phi \mid \phi \mathop{U} \phi \mid X \phi$
  where $\leq \in \{<,>,\geq,\leq\}$
- $P_{<0.5}(\Diamond \text{ full})$
  - Probability that queue becomes full is less than 0.5
- $P_{>0.98}(\neg \text{ retransmit } \mathop{U} \text{ receive})$
  - Probability that a message is eventually received successfully without any need for retransmission is greater than 0.98
Our algorithm A takes as input
- a stochastic model M,
- a formula $\phi$ in CSL,
- error bounds $\alpha$ and $\beta$, and
- three other parameters $\delta_1$, $\delta_2$, and $p_s$.

The result of model checking is denoted by $A^{\delta_1,\delta_2,p_s}(M, \phi, \alpha, \beta)$
- can be either true or false.

Details in [Sen et al. CAV’05]
Model Checking: Main Result Summarized

**Theorem:** If the model $M$ satisfies the following conditions

- **C1:** For every subformula of the form $P_{\geq p} \psi$ in the formula $\phi$ and for every state $s$ in $M$, the probability that a path from $s$ satisfies $\psi$ must not lie in the range

  \[ \frac{(p-\delta_1-\alpha)}{1-\alpha}, \frac{(p+\delta_1)}{1-\beta} \]

- **C2:** For any subformula of the form $\phi_1 \mathbin{U} \phi_2$ and for every state $s$ in $M$, the probability that a path from $s$ satisfies $\phi_1 \mathbin{U} \phi_2$ must not lie in the range $\left(0, \frac{\delta_2}{(1-p_s)^{N-1}q^{N-1}}\right)$, where $N$ is the number of states in the model $M$ and $q$ is the smallest non-zero transition probability in $M$

Then the algorithm provides the following guarantees

- **R1:**
  - $\Pr[A^{\delta_1, \delta_2, p_s}(M, \phi, \alpha, \beta) = \text{true} \mid M \not\models \phi] \leq \alpha$
  - $\Pr[A^{\delta_1, \delta_2, p_s}(M, \phi, \alpha, \beta) = \text{false} \mid M \models \phi] \leq \beta$
What is the expected number of clients that successfully connect to S?

\[
\text{CountConnected}() = \begin{cases} 
\text{if completed()} \text{ then count()} \\
\text{else } \bigcirc (\text{CountConnected}()) \text{ fi}
\end{cases}
\]

eval \( E[\text{CountConnected}()] \)
What is the probability that a client connected to S within 10 seconds after it initiated the connection request?

\[
\text{Prob}() = \begin{cases} 
0.0 & \text{if } \text{globaltime()} > 10 \\
(\text{Prob}()) & \text{if } \text{connected()} \\
1.0 & \text{else}
\end{cases}
\]

\[
\text{eval } \mathbb{E}[\text{Prob}()]
\]
The expected value of a QuaTEx expression is statistically evaluated with respect to two parameters $\alpha$ and $\delta$ provided as input.

We approximate the expected value by the mean of $n$ samples such that the size of $(1-\alpha)100\%$ confidence interval for the expected value computed from the samples is bounded by $\delta$.

Details in [Agha et al. QAPL’05]
Conclusion

- Vesta 2.0 supports
  - statistical model checking of probabilistic systems
  - query various quantitative aspects of a probabilistic system
- The tool is available for download at

  http://osl.cs.uiuc.edu/~ksen/vesta2/