Master 2 Internship: Parametererized Model Checking of Probabilistic Systems with Abstractions

Internship proposal proposed in common with:

- Nathalie Bertrand and Ocan Sankur: Inria - Irisa, Rennes – Team SUMO http://www.irisa.fr/sumo/
  Mail:nathalie.bertrand@inria.fr, ocan.sankur@irisa.fr
- Benjamin Monmege and Pierre-Alain Reynier: LIS (ex-LIF), Aix-Marseille Université, Marseille – Team MOVE
  http://www.lif.univ-mrs.fr/recherche/equipes/2
  Mail: benjamin.monmege@univ-amu.fr, pierre-alain.reynier@lif.univ-mrs.fr

The internship will be based in one of the two places (Rennes or Marseille) following the applicant's choice, with the possibility to go in the other place for one or several short period(s).

Topic: We are interested in model checking of probabilistic systems using automata-based formalisms. In this approach, one models systems using Markov chains or Markov decision processes, and uses algorithms and tools which can compute the minimal and maximal probability of the specification to be specified.

Many systems of interest are made of several copies of identical processes, e.g. a server and n identical clients. We call such systems *parameterized*, where the parameter refers to the number of copies of the replicated process. Parameterized verification is concerned with the devepment of algorithms to check the properties for *all* instances of the parameter. Several algorithms have been developed for non-probabilistic systems. In this internship, we will develop algorithms based on state-space abstraction to tackle the parameterized verification of probabilistic systems. The goal will be to compute approximations of the probability of the specification for all instances, or as a function of the number of instances. For instance, we would like to be able to tell that the probability of the server to respond to any client request within 10s is at least 90% if the number of clients is $n \in [1, 50]$, and lies within $[80\%, 90\%]$ if $n \in [51, 100]$, and in $[50\%, 80\%]$ for all $n > 100$.

The idea of using abstractions for parameterized systems is illustrated in the following figure. Given an unknown number of clients, one creates an abstract model with a single client process, and an "environment" process which is an over-approximation of an arbitrary number of clients. The resulting model is smaller, and if the environment is defined properly, it still captures important properties one can prove on the server and the single client.

The candidate is expected to read the literature on parameterized verification [2] and abstraction techniques for probabilistic systems [1] and come up with specialized abstractions that capture desired properties on parameterized probabilistic systems. We will then be interested in *automatic abstraction refinement*, where one starts by building and analyzing a coarse abstraction of the given system. If the computed interval on the probability of the specification is not precise enough, one automatically refines the abstraction, and restarts the analysis.

This work will require new theoretical developments since the abstraction and refinement procedures for such systems have not been studied before. Moreover, the candidate is expected to implement the procedure and run experiments on benchmarks to evaluate the impact of the algorithm in practice.

Keywords: parameterized verification, quantitative analysis, abstraction refinement, Markov chains, Markov decision processes, stochastic games

References

[1] C. Dehnert, D. Gebler, M. Volpato, and D. N. Jansen. On abstraction of probabilistic systems. In A. Remke and M. Stoelinga, editors, *Stochastic Model Checking. Rigorous Dependability Analysis Using Model Checking Techniques for Stochastic Systems: International Autumn School, ROCKS 2012, Vahrn, Italy, October 22-26, 2012, Advanced Lectures*, pages 87–116, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[2] A. John, I. Konnov, U. Schmid, H. Veith, and J. Widder. Parameterized model checking of fault-tolerant distributed algorithms by abstraction. In *Formal Methods in Computer-Aided Design (FMCAD), 2013*, pages 201–209. IEEE, 2013.