Incremental and Compositional Probabilistic Analysis of Programs under Uncertainty

Fouad ben Nasr Omri Safa Omri Ralf Reussner

Institute for Program Structures and Data Organization, Software Design and Quality, Faculty of Informatics
Karlsruhe Institute of Technology, Germany

fouad.omri@kit.edu safa.omri@student.kit.edu ralf.reussner@kit.edu

Uncertainty is a common aspect of modern software systems. The growing complexity of interaction with third-party components, the heterogeneity of the behavior of users and the possible external and environmental disturbances (e.g., operating with sensor errors, robotic manipulators, etc.) are introducing an uncertainty about the input values of a program. Such uncertainty is captured by specifying probability distributions over the program inputs. Assessing the reliability of such programs takes the form of computing the probability of assertions over the program variables. It is important to analyze how uncertainty can affect the reliability of such programs. Generally, reasoning about uncertain software systems is challenging due to using real-valued variables and having an imprecision in the inputs.

This paper proposes an approach for the static analysis of programs that are executed with uncertain inputs. We adapt symbolic execution to perform a probabilistic reasoning about the behavior of a program. Given an assertion over program variables, we extract the program paths which execute the assertion. We present an incremental and compositional approach to reason probabilistically about the program paths. The approach computes the probability of executing a program path for a given scope and a given uncertainty on the inputs values. Each program path is represented by a path condition and each path condition is a conjunction of branching constraints over the program variables. The probability of a path condition is computed by (i) quantifying the solution space of the branching constraints and (ii) specifying a probabilistic model to reason on it based on a given uncertainty. We propose to split a path condition into disjoint sets of branching constraints whose solution space can be determined independently from each other. Additionally, when the program contains looping constructs, instead of setting a static bound on the exploration depth of symbolic execution, we introduce a probabilistic bound which guides the symbolic execution incrementally. We use model counting to quantify the solution space of branching constraints defined over data structures. For path conditions containing nonlinear branching constraints, we developed a solution space quantification approach which combines interval branch-and-prune algorithms with adaptive stratified Monte Carlo integration. In order to compute the probability of a path condition, we define a probabilistic model where (i) the model is the path condition, (ii) the random variables are the variables of the path condition, (iii) an event is an assignment of values to the variables such that the path condition is satisfied and (iv) the uncertainty is defined as a full joint probability distribution over the input values.

We demonstrate promising results on a set of benchmarks from medicine and robotics domains. The preliminary results show the efficiency of our approach compared to recent research approaches. We also present an empirical evaluation applying our analysis approach on challenging container implementations for bug finding and coverage analysis. The paper and the experiments are publicly available [1].

[1] *https://sdqweb.ipd.kit.edu/wiki/Probabilistic_program_analysis_validation*

Submitted to:
VPT 2015