Dealing with Uncertainty in Verification of Nondeterministic Systems

Yamilet R. Serrano Llerena
School of Computing
National University of Singapore
Republic of Singapore
yserrano@comp.nus.edu.sg

ABSTRACT

Uncertainty complicates the formal verification of nondeterministic systems. Unpredictable changes and alterations in their environments can lead an invalid verification results and the decrease of confidence degree of these systems. However, current literature provides little account of addressing the uncertainty in formal verification. To address this problem, the goal of this research is to provide a method based on perturbation analysis for probabilistic model checking of nondeterministic systems which are modelled as Markov Decision Processes. And to apply our expected contributions to ubiquitous systems due to inherent presence of environment uncertainty and their resource limitations.

Categories and Subject Descriptors

D.2.4 [**Software Engineering**]: Software/Program Verification—*model checking, formal methods*; F.3.1 [**Logic and Meaning of Programs**]: Specifying and Verifying and Reasoning about Programs—*specification techniques*; G.3 [**Mathematics of Computing**]: Probability and Statistics—*Markov processes*

General Terms

Verification, Theory

Keywords

Uncertainty, Probabilistic Model Checking, Perturbation Analysis, Markov Decision Processes

1. INTRODUCTION

Nowadays, it is inevitable to notice that the uncertainty plays an important role in software engineering. Many of real-world systems with which we interact are subject to unpredictable changes and alterations in their environment that we cannot predict in advance, or even failure rates of some system components. As a result, these issues severely impact their functionality over time and they may produce variance in the quantification of the confidence degree that a system holds.

Generally, real systems such as wireless communication protocols, decision-making problems, adaptive algorithms, concurrency programs, biology systems, mobile devices, etc. have in common the presence of unpredictable behaviour that can only be accurately modelled by considering their stochastic characteristics [15, 20] . Particularly, they can be modelled as Markov Decision Processes, a variant of Markov Chains which exhibits a combination of probabilistic and nondeterministic behaviour [8]. Furthermore, they can be analysed by a probabilistic model checker which inputs a model and a specification and checks if the specification is satisfied by the model; additionally, it can output the probability or expected cost of it being satisfied behaviour [20].

However, the estimation of those probabilities in nondeterministic systems is far from accurate due to external factors and uncertainty parameters present in the model, and they can lead to a misleading or even invalid verification results [19]. Consequently, several studies have shown that all these possible alterations and risks are rarely considered explicitly in software engineering decisions and the state-of-the-art does not provide principled approaches to deal with them [7, 9, 18] .

Recent research works have addressed the problem of perturbed probabilistic models in the quantitative verification through of the perturbation analysis. These studies focus on compute perturbation bounds for probabilistic model checking of Parametric Discrete Time Markov Chains. It is important to note however, that this approach cannot be applied directly to Markov Decision Processes due to the presence of nondeterministic behaviour [21, 22].

Theoretically, this work aims to address the problem of uncertainty in nondeterministic systems. Firstly, the idea is to extend the perturbation analysis for the formal verification of Markov Decision Processes and to determine the discrepancy between a probabilistic and the real system represented by the model. Secondly, we plan to develop a prototype implementation for the verification of perturbed real systems. Practically, we aim to apply our approach to areas of software engineering such as ubiquitous computing where it is important to analyse the reliability properties since the presence of environmental uncertainty and resource limitations. Our contribution will help to design more efficient and robust real systems.

FSE'14, November 16–21, 2014, Hong Kong, China
ACM 978-1-4503-3056-5/14/11
http://dx.doi.org/10.1145/2635868.2666598

2. BACKGROUND AND RELATED WORK

In this section, we present brief literature review of the works related to the proposed research and identify the research gaps.

Firstly, we study the literature related to probabilistic model checking and suitable probabilistic model for nondeterministic systems. Probabilistic model checking is a formal verification for calculating the likelihood of the occurrence of certain events during the execution of a system [20]. For instance, a probabilistic model checker takes as input a probabilistic model (generally variants of Markov chains) represented as a state transition system which encodes the probability of making a transition between states; and a specification typically represented by some temporal logic, for example Probabilistic Computational Tree Logic (PCTL) [8]. It is then possible to verify whether or not each property is satisfied; and provide of quantitative measurements. One of the most widely known and used probabilistic model checker is PRISM [16].

Regarding to the probabilistic models used in probabilistic model checking, Markov Decision Processes (MDPs) are commonly used for modelling systems that exhibit a combination of probabilistic and nondeterministic behaviour [8]. Each transition of an MDP consists of a nondeterministic choice of actions. However, the semantics of an MDP depend on a deterministic scheduler which resolves the nondeterministic choices and it originates an induced Discrete Time Markov Chain (DTMC). Consequently, the probability mass of a set of execution sequences also depends on the chosen scheduler [1]. Reachability properties and full PCTL properties are desired specifications for the model checking MDPs. In order to perform the model checking of these properties, in the literature, there exists different techniques, for example: linear programming, value iteration and policy iteration [2, 5, 8].

With respect to the presence of unpredictable changes in the environment of the models, the literature shows a deep concern about the uncertainty in software engineering. Flyvbherg *et al.* [7] argued that real-world systems require us to consider uncertainty as a first-class concern in the design, implementation, and deployment of those systems. Furthermore, Letier *et al.* [18] proposed a systematic method allowing software architects to describe uncertainty and to calculate the consequences of uncertainty through Monte-Carlo simulation. On the other hand, Filieri *et al.* [6] focus on reliability properties in adaptive systems, using DTMCs and probabilistic model checking. In this direction, we can also find researches oriented to the computation of the asymptotic bounds for probabilistic verification, given in terms of Parametric Discrete Time Markov Chains (PDTMCs)[21, 22].

In the case of Markov Decision Processes, there have been several studies that address the uncertainty through of Parametric Markov Decision Processes (PMDPs), in which transition probabilities are not fixed, but depend on a set of parameters [17]. In this context, Moritz *et al.* [10, 11] provide an approach to solve the PCTL synthesis problem for PMDPs with reachability reward properties. Likewise, several researches try to deal with uncertain Markov decision problems using optimisation algorithms, uncertain transition matrices and adaptive mechanism that aim to adjust the system and reach the robustness of the properties in the model [3, 4, 19].

3. RESEARCH QUESTIONS

The goal of this research is to address the problem of the inherent presence of uncertainty in software systems that evince non-deterministic behaviour, and its impact in the preservation of their qualitative and quantitative properties. In detail, we plan to address the following research questions:

RQ1 Can we measure the effect of uncertainty in the formal verification of nondeterministic systems?

RQ2 Can we determine the discrepancy between a probabilistic model and the real system represented by the model?

RQ3 Can we apply an enhanced verification approach to verify properties related to the way that uncertainty manifests itself in ubiquitous computing?

4. APPROACH AND CHALLENGES

In this section, we describe our research approach in order to achieve the goals mentioned above.

4.1 Perturbation Analysis in Model Checking MDPs

To address RQ1 and RQ2 we are going to develop and evaluate a tool based on perturbation analysis for nondeterministic systems. These systems will be modelled as parametric Markov decision processes because they are useful when the exact probability of a transition is not known, or when it is know but not considered relevant [20] and they are equipped with the norm of total variance to measure the perturbations of their abstract parameters.

Since Perturbation Analysis has been used successfully for Discrete Time Markov Chains (DTMCs) [21, 22] and Markov Decision Processes (MDPs) can be seen as a generalisation of DTMC, we plan to extend this approach for the model checking of MDPs. Currently, perturbation analysis provides *perturbation bounds* based on the definition of condition numbers and quadratic bounds for probabilistic model checking of perturbed stochastic models. In particular, condition numbers are intuitive and informative format to capture the sensitivity of the parameters under uncertain events or alterations. And, quadratic numbers provide suitable bound predictions under perturbed parameters. In short, these perturbations bounds predict the maximal perturbation distance that might occur to the verification results with respect to the perturbation quantities in the stochastic models. This approach has been evaluated using case studies on variant of well know system models [22].

Because of the presence of nondeterminism in MDPs, one must presume the existence of a scheduler that resolves the nondeterministic choice in each state. Thus, given some probabilistic property, the value computed for its probability of satisfaction can vary depending on which scheduler is used to resolve the nondeterministic choices. For this reason, model checking of MDPs involves determining the minimum and maximum probability of satisfaction over all possible schedulers. And thus in order to compute these minimum and maximum probabilities, algorithms for model checking MDPs must efficiently determine which scheduler(s) produce the minimum probability and which scheduler(s) produce the maximum probability. These schedulers are usually called *optimal schedulers.*

Thus, we propose to compute the minimum and maximum probabilities using a conventional technique in probabilistic model checking such as the *value iteration method*, which computes a good enough approximation of the values and offers better scalability than linear programming [8]. Due to the fact that schedulers induce parametric DTMCs. Using this finite set of parametric DTMCs, we propose to compute asymptotic bounds by assuming that the given perturbation is sufficiently small. The challenge of our approach is also to consider all the possible perturbation bounds that could exist in the model because it may be the case that the schedulers and bounds computed by the value iteration method might not reveal the maximum influence of the uncertainty in the model.

Furthermore, we will extend our approach for reachability, PCTL properties and performance properties, such as expected reward. Finally, the expected contribution of our approach is to provide useful information for taking strategies in the design and refinement of the model; and to help to the correctness of the real system.

4.2 Uncertainty in Ubiquitous Computing

We plan to address RQ3 after to provide an efficient way to handling the uncertainty in MDP model through of Perturbation Analysis. We have considered the ubiquitous computing systems as a concrete case studies of the potential applicability of our approach. The reason of focusing on ubiquitous computing is because it is one of the perfect scenarios where the environmental uncertainty and resource limitation are clearly visible in the systems; and at same time, we expect high level of predictability and robustness [13].

Consequently, there exist many challenges in the formal verification of these systems. In addition, for modelling ubiquitous computing devices, we often need to include probabilities, time delays and resource usage in the models. Therefore, models in this area are inherent probabilistic. According to Kwiatkowska *et al.* [14], several problems in this area can be addressed, for example the unreliability of wireless communication technologies such as Bluetooth which use randomised back off schemes to minimise collisions; also, embedded devices are frequently powered by battery and components may be prone to failure.

In short, this emergent area opens many research gaps related to the study of uncertainty; and on the other hand, it is potentially one of the best scenarios for applying the contributions of our research in a near future.

5. RESEARCH PROGRESS

In this section, we describe our progress to date. For addressing RQ1 and RQ2, we have developed an early prototype implementation in Haskell Programming Language to interact with PRISM. The reason for choosing Haskell is because it is a purely-functional programming language and it is an open-source product which allows rapid development of robust and correct software [12].

Currently in our approach, a non deterministic system is modelled as a Markov Decision Process which is built in PRISM. Next, from the probabilistic model checker we export a set of states and a transition matrix which represents all the possible transitions in the model. These elements, plus the set of perturbation parameters are the input of Haskell prototype implementation.

So far, our prototype implementation is focused on reachability properties of MDPs. For example: *What is the minimum or maximum probability of reaching a set of target states T?*. In consequence, we have implemented the qualitative and quantitative reachability of the model using value iteration method. And from the optimal schedulers generated as output of the value iteration method, we compute the perturbed bound as *a condition number* which is able to provide suitable bound prediction in practice. Additionally, we have explored three different methods that trade off precision for efficiency and we have computed all the possible condition numbers in the model. In this way, we can obtain a better understanding of the uncertainty effect in the model and the impact over it.

We have evaluated our approach on several theory MDPs obtained from different textbooks and handbooks. Verification results from those models in our Haskell prototype are same in comparison to quantitative verification in PRISM. On the other hand, we have also confirmed that the perturbation bounds have been computed correctly. Likewise, we have included in our experiments different case studies based on Communication Network, Self-Stabilisation Algorithm [20] and Machine Replacement Problem [4]. They have been perturbed simulating a failure component when a message is sent, a noise in the transfer of a token and the decision of taking the repairing action, respectively.

Figure 1 shows the model of machine replacement problem. As can be seen from the figure, the problem has been modelled using 10 states and 2 nondeterministic choices per state. Particularly, this case study gives us an relevant hints about the perturbation effect in the model and the importance of considering all possible perturbation bounds. For example, considering that the target is the maximum ageing of the machine, which is 8 years in the model. This model has 32 optimal schedulers for the worst-case of reachability and 2 for the best-case. Their conditional numbers are 0.44 and 0, respectively. On the other hand, with respect to 1024 schedulers that this case study has in total, their perturbation bounds as condition number are between 0 and 6.89. In other words, it means that there exists at least one scheduler which due to the perturbation, it drives to the maximum noise in the model. And this scheduler is not the same who provides the maximum probability.

For next stages in our approach, we will improve the aforementioned methods and provide an optimal algorithm for calculating the all perturbation bounds without exploring the complete model. As well as, we plan to extend our approach for full PCTL, including reward properties. Likewise, it is pending for our future work, the application of our approach in the verification of ubiquitous systems in order to address RQ3.

6. CONCLUSIONS

Motivated by the uncertainty problem in the modelling and quantitative verification of real systems, we plan to study the impact and the sensitivity of constrained reachability, PCTL and expected reward properties of those systems modelled as Markov Decision Processes, to perturbations of their distribution parameters.

Our research seeks to provide a method based on perturbation analysis for the computation of asymptotic bounds which aim to predict the maximal perturbation distance that might occur to the verification results. Early results from

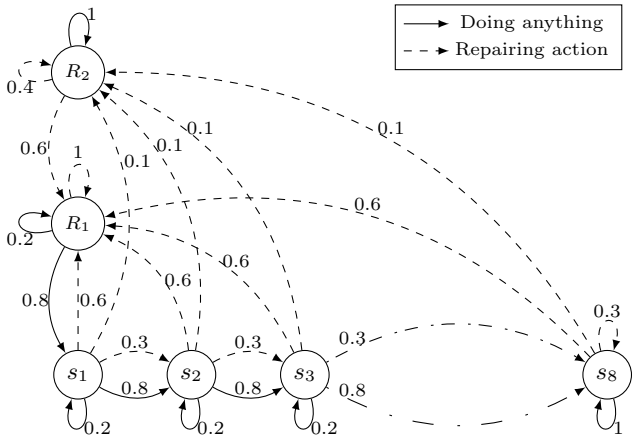

Figure 1: Machine Replacement Problem modelled as a Markov Decision Process

our initial prototype show some promise. Furthermore, we plan to evaluate our contributions to the real applications as ubiquitous systems due to they present an uncertainty environment and resource limitations.

7. ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Dr. David Rosenblum (david@comp.nus.edu.sg) and Dr. Guoxin Su (sugx@comp.nus.edu.sg), for their support and guidance throughout this research. This work was supported in part by a Research Scholarship from National University of Singapore.

8. REFERENCES

[1] Husain Aljazzar and Stefan Leue. Generation of counterexamples for model checking of Markov decision processes. In *QEST*, pages 197–206, 2009.

[2] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. The MIT Press, 2007.

[3] B. Bethke, L. F. Bertuccelli, and J. P. How. Experimental demonstration of adaptive MDP-based planning with model uncertainty. In *AIAA Guidance Navigation and Control*, Honolulu, Hawaii, 2008.

[4] Erick Delage and Shie Mannor. Percentile optimization in uncertain Markov decision processes with application to efficient exploration. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 225–232, New York, NY, USA, 2007. ACM.

[5] Kousha Etessami, Marta Z. Kwiatkowska, Moshe Y. Vardi, and Mihalis Yannakakis. Multi-objective model checking of Markov decision processes. *Logical Methods in Computer Science*, 4(4), 2008.

[6] Antonio Filieri, Carlo Ghezzi, and Giordano Tamburrelli. Run-time efficient probabilistic model checking. In *ICSE*, pages 341–350, 2011.

[7] B Flyvbjerg and A Budzier. Why your IT project may be riskier than you think. *Harvard Business Review*, 89(9):23–25, 2011.

[8] Vojtech Forejt, Marta Z. Kwiatkowska, Gethin Norman, and David Parker. Automated verification techniques for probabilistic systems. In *SFM*, pages 53–113, 2011.

[9] David Garlan. Software engineering in an uncertain world. In *FoSER*, pages 125–128, 2010.

[10] Ernst Moritz Hahn, Tingting Han, and Lijun Zhang. Synthesis for PCTL in parametric Markov decision processes. In *NASA Formal Methods*, pages 146–161, 2011.

[11] Ernst Moritz Hahn, Holger Hermanns, and Lijun Zhang. Probabilistic reachability for parametric Markov models. In *SPIN*, pages 88–106, 2009.

[12] Haskell.org. The Haskell programming language. `http://www.haskell.org/haskellwiki/Haskell`. Accessed: 2014-06-20.

[13] M. Kwiatkowska, G. Norman, and D. Parker. A framework for verification of software with time and probabilities. In K. Chatterjee and T. Henzinger, editors, *Proc. 8th International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'10)*, volume 6246 of *LNCS*, pages 25–45. Springer, 2010.

[14] Marta Z. Kwiatkowska. Advances in quantitative verification for ubiquitous computing. In *ICTAC*, pages 42–58, 2013.

[15] Marta Z. Kwiatkowska, Gethin Norman, and David Parker. Probabilistic model checking in practice: case studies with PRISM. *SIGMETRICS Performance Evaluation Review*, 32(4):16–21, 2005.

[16] Marta Z. Kwiatkowska, Gethin Norman, and David Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *CAV*, pages 585–591, 2011.

[17] Ruggero Lanotte, Andrea Maggiolo-Schettini, and Angelo Troina. Parametric probabilistic transition systems for system design and analysis. *Formal Asp. Comput.*, 19(1):93–109, 2007.

[18] Emmanuel Letier, David Stefan, and Earl T. Barr. Uncertainty, risk, and information value in software requirements and architecture. In *ICSE*, pages 883–894, 2014.

[19] Arnab Nilim and Laurent El Ghaoui. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

[20] J. Rutten, M. Kwiatkowska, G. Norman, and D. Parker. *Mathematical Techniques for Analyzing Concurrent and Probabilistic Systems, P. Panangaden and F. van Breugel (eds.)*, volume 23 of *CRM Monograph Series*. American Mathematical Society, 2004.

[21] Guoxin Su and David S. Rosenblum. Asymptotic bounds for quantitative verification of perturbed probabilistic systems. In *ICFEM*, pages 297–312, 2013.

[22] Guoxin Su and David S. Rosenblum. Perturbation analysis of stochastic systems with empirical distribution parameters. In *ICSE*, pages 311–321, 2014.