Towards Fully-Compositional Simulation-Based Analysis of Autonomous Systems[*]

Beyazit Yalcinkaya[†], Daniel J. Fremont[‡], Sanjit A. Seshia[†]

[†]University of California, Berkeley, CA, USA
[‡]University of California, Santa Cruz, CA, USA
beyazit@berkeley.edu, dfremont@ucsc.edu, sseshia@berkeley.edu

Abstract

Scalable methods for verifying the safe behavior of learning-enabled autonomous systems, especially those operating in safety-critical settings, have become a crucial concern. The inherent black-box nature of these systems, rooted in the complexity of the learned models, prevents us from using classical model-based verification techniques. To this end, simulation-based analysis has become commonplace for assessing the correctness of AI-based autonomous systems. The challenge is then to make the simulation-based analysis techniques more scalable. A critical piece of this problem is to leverage the compositional nature of simulations to reduce computation. In this paper, we focus on this challenge, highlighting the importance of compositional simulation-based analysis and discussing future work in this domain.

Introduction

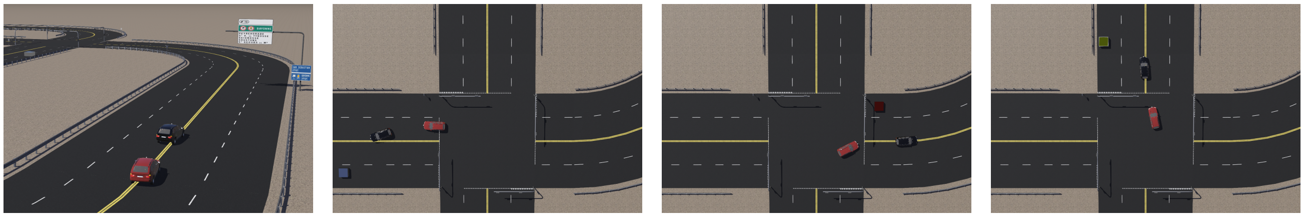
Artificial intelligence (AI) and machine learning (ML) are increasingly being integrated into autonomous systems, handling tasks spanning perception, prediction, planning, and control. Yet, the correctness, and especially the safety, of systems incorporating AI/ML-based components is still a major concern. Formal methods can play a crucial role in guaranteeing the correctness of these systems (Seshia, Sadigh, and Sastry 2022). Due to the high complexity of AI/ML components, contemporary verification and testing methods often treat them as *black-box*, departing from classic model-based approaches. To this end, simulation-based formal analysis has become a common practice for reasoning about the behaviors of autonomous systems. In this setting, the correctness of the system is evaluated against a formal specification defining the desired behavior of the system by examining its behaviors in multiple simulation runs.

High-confidence analysis results necessitate the execution of *a large number of long-running simulations*. Due to the complex environment models and the high dimensionality of feature spaces, simulation-based analysis continues to be computationally expensive. Nonetheless, in numerous instances, especially in autonomous driving, simulation models, also known as *scenarios*, compose several smaller scenarios. To improve the efficiency and therefore the scalability of simulation-based analysis, it is important to leverage the inherent composition of scenarios by converting a monolithic analysis process into several smaller ones and combining the results of these smaller scenarios.

In our previous work (Yalcinkaya et al. 2023), we introduced a compositional approach to the simulation-based analysis of autonomous systems. It is a general framework that can be applied to different tasks, e.g., *falsification* (Dreossi, Donzé, and Seshia 2019) and *statistical verification* (Legay, Delahaye, and Bensalem 2010). We define a formal structure over the scenarios used in simulations. This formalization allows for the definition of a scenario using a hierarchical composition of other scenarios. Using this hierarchical formal definition, the analysis method then decomposes a given scenario into several smaller sub-scenarios and performs the analysis at the sub-scenario level. The outcomes from each sub-analysis problem are then combined to derive a solution for the larger problem. The method assumes Markovian (memoryless) specifications. This assumption is motivated by the fact that most specifications encountered in AI-based autonomous systems are Markovian, such as the absence of collisions, adherence to traffic lights, etc. In our section on future work, we briefly discuss how to extend this approach to non-Markovian specifications.

In this paper, we present a summary of our previous work (Yalcinkaya et al. 2023). We first present a motivating example and briefly explain how the method works. We then present experiment results highlighting the efficacy of our compositional approach compared to its monolithic counterpart. We finally conclude by emphasizing the scalability of the compositional analysis and discussing future work.

[*]This work is partially supported by NSF grants 1545126 (VeHICaL, including an NSF-TiH grant) and 1837132, by DARPA contracts FA8750-18-C-0101 (AA), FA8750-20-C-0156 (SDCPS), and FA8750-23-C-0080 (ANSR), by Berkeley Deep Drive, by C3DTI, and by Toyota under the iCyPhy center.
AAAI 2024 Spring Symposium on User-Aligned Assessment of Adaptive AI Systems. Stanford University, Stanford, CA, USA.



(a) `subScenario1` (b) `subScenario2L` (c) `subScenario2R` (d) `subScenario2S`

Figure 1: Snapshots of the sub-scenarios of the motivating example

Compositional Simulation-Based Analysis

Motivating Example. Consider an autonomous driving task with two cars, namely `Leader` and `Follower`, both equipped with AI-based components to drive and achieve their goals. The former navigates through the city and, upon reaching an intersection, randomly chooses to turn left, right, or to proceed straight with uniform probability. The latter, on the other hand, follows `Leader` while maintaining a safe distance. We want to analyze to see whether this system can keep a safe distance between the cars under various environment conditions. To this end, on top of the simulation environment, we provide a scenario defining the environment conditions we want to use for the simulations. The scenario defines an initial straight road segment with a curve at the end leading to an intersection. After the intersection, there are three road segments for turning left, right, or proceeding straight. Each of these road segments includes an adversarial object to trick the computer vision components of the cars into violating the system-level specification, which defines the safe distance between the cars, expressed in Metric Temporal Logic (MTL) (Koymans 1990) as $\Box(\texttt{distance} \geq 5 \land \texttt{distance} \leq 15)$. The scenario defines distributions for the initial positions of the cars and distributions for the location and the color of the adversarial objects. We use Scenic (Fremont et al. 2023), a probabilistic scenario-description language, to define this scenario.

Method. The given scenario is composed of sub-scenarios. The first one (`subScenario1` given in Figure 1(a)) is the initial straight road segment leading to the intersection. The next scenario is at the intersection and is composed sequentially with the previous one. The intersection scenario consists of three sub-scenarios indicating different possibilities that might happen, i.e., turning left, right, or proceeding straight (`subScenario2L`, `subScenario2R`, `subScenario2S` given in Figures 1(b) to 1(d), respectively). Our method automatically decomposes the given Scenic scenario into its sub-scenarios to analyze each one separately. For a scenario to be decomposed in this manner, we define an abstract syntax for writing scenarios formally in (Yalcinkaya et al. 2023), and Scenic programs inherently follow this syntax.

The procedure analyzes all sub-scenarios separately in the order indicated by the hierarchical structure of the sub-scenarios given by the monolithic scenario and combines the analysis results at the end. Specifically, in this example, we start with the analysis of `subScenario1`, and we save the final state distribution at the end of this scenario. The computed output distribution is then used as the initial state distributions of the following sub-scenarios, i.e., `subScenario2L`, `subScenario2R`, `subScenario2S`. Since there are no other scenarios following these ones, once the analyses of these sub-scenarios are completed, we terminate. The crucial insight behind this method is that the monolithic analysis of this system redundantly executes `subScenario1` many times even though that stage of the scenario does not yield any interesting behavior. On the other hand, the compositional approach analyzes `subScenario1` only once and uses the obtained output state distribution as the input distribution of other sub-scenarios, and therefore it avoids redundant computation.

Experiment Results

We instantiate our high-level algorithm for falsification and statistical verification, and we compare our compositional method with the monolithic approach. To demonstrate the performance gain obtained by our method we use the number of simulator steps taken for the analysis before termination as this is a platform-agnostic metric for evaluation. In falsification, our compositional approach takes ∼2× less simulator steps before finding a counterexample. In statistical verification, our method needs ∼4× fewer steps compared to the monolithic baseline. We also compare the specification satisfaction probabilities computed at the end of statistical verification. The experiment results show that our method computes this probability with only a ∼2% error with respect to the estimate of the monolithic baseline.

Conclusion & Future Work

We presented our framework for the compositional simulation-based analysis of AI systems, which decomposes a given simulation-based analysis task into several smaller ones to reduce computation. We believe compositional approaches for the analysis of AI systems are crucial for more scalable analyses in pursuit of safe AI systems. The next step is to extend the given method to the general case of non-Markovian specifications. This extension requires developing efficient ways to store the state of the specification and other environment parameters along with the output state distributions collected at the end of each sub-scenario.

References

Dreossi, T.; Donzé, A.; and Seshia, S. A. 2019. Compositional falsification of cyber-physical systems with machine learning components. *Journal of Automated Reasoning*, 63: 1031–1053.

Fremont, D. J.; Kim, E.; Dreossi, T.; Ghosh, S.; Yue, X.; Sangiovanni-Vincentelli, A. L.; and Seshia, S. A. 2023. Scenic: A language for scenario specification and data generation. *Machine Learning*, 112(10): 3805–3849.

Koymans, R. 1990. Specifying real-time properties with metric temporal logic. *Real-time systems*, 2(4): 255–299.

Legay, A.; Delahaye, B.; and Bensalem, S. 2010. Statistical model checking: An overview. In *International conference on runtime verification*, 122–135. Springer.

Seshia, S. A.; Sadigh, D.; and Sastry, S. S. 2022. Toward Verified Artificial Intelligence. *Communications of the ACM*, 65(7): 46–55.

Yalcinkaya, B.; Torfah, H.; Fremont, D. J.; and Seshia, S. A. 2023. Compositional Simulation-Based Analysis of AI-Based Autonomous Systems for Markovian Specifications. In *International Conference on Runtime Verification*, 191–212. Springer.