Algorithmic Improvisation for Dependable Intelligent Autonomy

Sanjit A. Seshia
University of California, Berkeley, CA, USA
http://www.eecs.berkeley.edu/~sseshia
sseshia@eecs.berkeley.edu

Abstract

Algorithmic Improvisation, also called control improvisation or controlled improvisation, is a new framework for automatically synthesizing systems with specified random but controllable behavior. In this talk, I will present the theory of algorithmic improvisation and show how it can be used in a wide variety of applications where randomness can provide variety, robustness, or unpredictability while guaranteeing safety or other properties. Applications demonstrated to date include robotic surveillance, software fuzz testing, music improvisation, human modeling, generating test cases for simulating cyber-physical systems, and generation of synthetic data sets to train and test machine learning algorithms. In this talk, I will particularly focus on applications to the design of intelligent autonomous systems, presenting work on randomized planning for robotics and a domain-specific probabilistic programming language for the design and analysis of learning-based autonomous systems.

2012 ACM Subject Classification Theory of computation → Automated reasoning; Computing methodologies → Machine learning; Computing methodologies → Artificial intelligence; Theory of computation → Formal languages and automata theory; Software and its engineering → Domain specific languages; Theory of computation → Complexity theory and logic

Keywords and phrases Formal methods, synthesis, verification, randomized algorithms, formal specification, testing, machine learning, synthetic data generation, planning

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2020.5

Category Invited Talk

Funding This work has been supported in part by the National Science Foundation (NSF) Graduate Research Fellowship Program under Grant No. DGE1106400, by NSF grants CCF-1139138, CNS-1545126 (VeHICaL), CNS-1646208, and CCF-1837132, DARPA under agreement numbers FA8750-16-C0043 and FA8750-18-C-0101, by the iCyPhy center, Berkeley Deep Drive, and by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

Acknowledgements The work on algorithmic improvisation described in this talk has been co-led by my Ph.D. student, Daniel Fremont, and is reported in his Ph.D. dissertation.

Summary

Algorithmic improvisation is a new framework for automatically synthesizing systems with specified random but controllable behavior. Such systems are known as improvisers and have applications in a variety of applications where randomness can provide variety, robustness, or unpredictability in a specified, controlled manner. This framework, also termed as *control improvisation* or *controlled improvisation*, was proposed and formalized by the author and colleagues several years ago [2, 6, 5]. Informally, an improviser is a generator of data items $d_1, d_2, d_3, \ldots$ subject to three kinds of constraints:

1. *Hard Constraints:* Each data item d_i must satisfy all these constraints.
2. *Soft Constraints:* A data item d_i must satisfy these constraints as measured by a tunable quantity, typically a probability written as $1 - \delta$ for tunable parameter δ.

3. *Randomness Constraints:* The output distribution of the improviser must satisfy specified properties, e.g., obeying a particular distribution.

The problem of synthesizing an improviser is termed as the control (or controlled) improvisation problem. The papers and thesis by Fremont et al. [6, 5, 4] lay out the foundations of the theory of control improvisation, analyzing its complexity for different variants of the problem involving various forms of constraints.

Algorithmic improvisation has been demonstrated in a variety of applications. Here are some of these applications:

- Music improvisation, generating controlled random variations of a given melody [2];
- Modeling human behavior for controlling Internet-of-Things (IoT) devices in a home automation context [1];
- Synthesizing control policies for controlling vehicles [9];
- Synthesizing randomized plans for robotic surveillance [3];
- Generating test inputs for software fuzz testing [4];
- Generating test cases for simulating cyber-physical systems [8, 7], and
- Generation of synthetic data for training and testing machine learning applications [7].

In these applications, the type of data generated by the improviser varies (music, control policies, test inputs, images, etc.) and the formalism used to encode constraints also varies, including logics, automata, and domain specific languages.

In this invited talk, I will give an overview of the theory of algorithmic improvisation, give a tour of some of the key applications with a particular focus on the design of intelligent autonomous systems, and present an outlook on the exciting future directions that remain to be explored.

References

1 Ilge Akkaya, Daniel Fremont, Rafael Valle, Alexandre Donzé, Edward A. Lee, and Sanjit A. Seshia. Control improvisation for probabilistic temporal specifications. In *Proceedings of the 1st IEEE International Conference on Internet-of-Things Design and Implementation (IoTDI)*, April 2016.

2 Alexandre Donzé, Rafael Valle, Ilge Akkaya, Sophie Libkind, Sanjit A. Seshia, and David Wessel. Machine improvisation with formal specifications. In *Proceedings of the 40th International Computer Music Conference (ICMC)*, pages 1277–1284, September 2014.

3 Daniel Fremont and Sanjit A. Seshia. Reactive control improvisation. In *30th International Conference on Computer Aided Verification (CAV)*, 2018.

4 Daniel J. Fremont. *Algorithmic Improvisation*. PhD thesis, EECS Department, University of California, Berkeley, August 2019. URL: `http://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-133.html`.

5 Daniel J. Fremont, Alexandre Donzé, and Sanjit A. Seshia. Control improvisation. *CoRR*, abs/1704.06319, 2017.

6 Daniel J. Fremont, Alexandre Donzé, Sanjit A. Seshia, and David Wessel. Control improvisation. In *35th IARCS Annual Conference on Foundation of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 45 of *LIPIcs*, pages 463–474, December 2015.

7 Daniel J. Fremont, Tommaso Dreossi, Shromona Ghosh, Xiangyu Yue, Alberto L. Sangiovanni-Vincentelli, and Sanjit A. Seshia. Scenic: A language for scenario specification and scene generation. In *Proceedings of the 40th annual ACM SIGPLAN conference on Programming Language Design and Implementation (PLDI)*, June 2019.

8 Daniel J. Fremont, Edward Kim, Yash Vardhan Pant, Sanjit A. Seshia, Atul Acharya, Xantha Bruso, Paul Wells, Steve Lemke, Qiang Lu, and Shalin Mehta. Formal scenario-based testing of autonomous vehicles: From simulation to the real world. In *23rd IEEE International Conference on Intelligent Transportation Systems (ITSC)*, September 2020.

9 Jin I. Ge and Richard M. Murray. Voluntary lane-change policy synthesis with control improvisation. In *57th IEEE Conference on Decision and Control, CDC 2018, Miami, FL, USA, December 17-19, 2018*, pages 3640–3647. IEEE, 2018.