

Counterexamples in Cyber-Physical Systems Theorem Proving Falsification of Hybrid Discrete-Continuous Programs

Tessa Hall and Stefan Mitsch^[0000-0002-3194-9759]

School of Computing
DePaul University
Chicago IL 60604, USA
{thall142, smitsch}@depaul.edu

Keywords: Hybrid programs · Theorem Proving · Falsification

Abstract

Autonomous and cyber-physical systems operate increasingly often in safety-critical domains (e.g., robot navigation, aircraft collision avoidance, autonomous driving, or automated railroad operations), which makes safety guarantees a necessity when aiming for trustworthy systems. Theorem proving provides highest safety guarantees by showing the correctness of such systems with human-inspectable proofs consisting of arguments explainable in a logic system. The correctness arguments in cyber-physical systems require a logic system that can express the interaction between computation, control, and physics, which are a combination of discrete and continuous phenomena. Such *hybrid systems* models are, however, extremely subtle to get right because of their nondeterministic nature to cover infinitely many possible system executions at once, which makes for strong correctness guarantees but makes it challenging to simulate sample behavior when providing counterexamples for incorrect models. Proof attempts therefore often iterate between making progress in the proof and finding and correcting modeling mistakes.

To facilitate finding modeling mistakes, we propose an approach based on an integration of theorem proving with falsification. The two methods complement each other well: theorem proving mathematically shows absence of bugs in a correct model (i.e., a verified model provably satisfies a desired correctness property), while falsification uses optimization to find bugs in an incorrect model quickly and automatically (i.e., a falsified model violates a desired correctness property). We propose to find bugs in nondeterministic models by using falsification to steer their nondeterministic operators when attempting to find violations of a desired correctness property.