

Can Reactive Synthesis and Syntax-Guided Synthesis Be Friends?

MohammadAmin Raeisi

Summary

This paper presents a program synthesis technique that combines reactive synthesis and SyGuS, allowing us to automatically generate non-trivial reactive programs, such as a Linux kernel scheduler. In reactive synthesis, we are interested in finding a reactive system, such as a hardware circuit, that implements a given temporal specification. If such a system exists, we say that it realizes the specification. Reactive synthesis focuses on the potentially infinite interaction between the system and its environment and constructs intricate control strategies that ensure that the system reacts appropriately to any possible move by the environment. While reactive synthesis is focused on generating reactive systems, the goal of SyGuS synthesis techniques is to find a data-transforming function between input and output data. A SyGuS solver takes as input a specification, written in some logical formalism, such as a set of input/output examples. The goal is to find a function describing the relation given by the specification.

The main steps of this system are as follows: it first identifies data transformation tasks in the specification, and then, using SyGuS, generates functions that implement these data transformations. Next, using reactive synthesis integrates these functions into a control structure that satisfies the reactive requirements. As the interface between SyGuS and reactive synthesis, it uses TSL, a variant of temporal logic that is sufficiently powerful to express the requirements on both data transformations and reactive behavior.

Strengths

- The pseudo-code of the presented algorithms makes the content clear to a good extent.
- The diagrams and table at the end of the paper show the results of different benchmarks well.
- It is good that the limitations of the tool are mentioned in their GitHub documents and the limitations of the grammar are mentioned in the paper.
- The diagram on page 5 shows the Overview of the Synthesis Procedure well.

Weaknesses

- It is better to give a brief explanation of Reactive synthesis in the introduction section.
- The introduction section is very long.
- It does not explain the concept of square and circle used in the introduction example.
- The text of the article is so complicated that you have to read it several times to understand each part.
- The temporal atom collection outlined in section 4.1 of the paper is substituted by an approximation and The refinement loop given in section 4.4 is not fully implemented in their tool.
- It is better to explain the steps of the algorithm by providing one or more examples. Failure to provide examples has made the contents more difficult to understand.