Augmented Example-based Synthesis using Relational Perturbation Properties

MohammadAmin Raeisi

Summary

This paper presents a new approach to addressing the ambiguity/generalizability problem in example-based synthesis and is implemented as *SketchAX* tool. Their framework supports multiple domains and synthesizers, can be instantiated with different user interfaces (UIs), and can be used with existing techniques based on structured DSLs, ranking functions, or user feedback loops. The key feature of SketchAX is that it places a *semantic bias* on the hypothesis space based on *relational perturbation properties*. In contrast to general relational properties that may express constraints relating to multiple programs or multiple executions of a single program, relational perturbation properties relate the perturbation/change in a program output to the perturbation/change in a program input. An example of such a property is *permutation invariance*: the program output does not change when the program input (array) elements are permuted.

Their core approach is based on two steps: (1) automatically generate an augmented set of examples by applying relational perturbation properties to the user-provided examples, and (2) use an existing example-based synthesizer to generate a program consistent with the augmented set of examples. Their solution strategy of enforcing relational properties using examples instead of formal specifications is inspired by two observations: (1) not all example-based synthesizers accept specifications over all inputs, (2) in cases where an example-based synthesizer accepts such specifications, there is typically a significant performance penalty in terms of synthesis time. Relational perturbation properties enable the synthesizer to easily generate additional examples from user-provided examples.

Strengths

- This paper has presented the problem statement well by providing an example.
- It has shown the superiority of this method over previous methods by presenting various benchmarks and diagrams.
- This paper has tried to convey the algorithm better by using pseudo-codes.
- It expresses the motivation well without adding any extra words in the introduction part.
- By defining multiple relative properties for the synthesizer, it has greatly reduced the search space.
- It is not limited to a specific domain.
- This paper has reduced user interaction with the synthesizer.

Weaknesses

- The paper is very long, and the reader gets bored reading it!
- This paper has not proposed a new idea and has only added minor changes to the previous methods.
- The pseudo-codes in the article have a complicated notation and at first glance, they do not convey the algorithm easily.
- The paper is primarily theoretical and does not focus much on implementation details.
- It is better to include the link to the tool they have implemented in the paper.