Interactive Program Synthesis by Augmented Examples

MohammadAmin Raeisi

Summary
This paper presents an interactive synthesis method for generating regular expressions and developing an interactive synthesizer called REGAE. Regexes are a versatile text-processing utility that has found numerous applications ranging from search and replacement to input validation. Despite their popularity, regexes are considered hard to understand, compose, and error-prone, even for experienced programmers. For programmers and non-programmers, it is more natural to clarify their intent on top of the original examples they give rather than the programs synthesized by a foreign PBE system. This paper introduces an interaction model that supports two types of augmentation on user-given examples: (1) *Semantic augmentation*, (2) *Data augmentation*.

As an instance of *semantic augmentation*, REGAE allows users to mark parts of input examples that should be kept verbatim or generalized to a character class, such as digits in a regex.

As an instance of *Data augmentation*, REGAE automatically generates two types of additional inputs, those similar to the user-provided examples and corner cases and then clusters them based on how synthesized programs behave on them. Hence users do not have to mentally simulate the synthesized regexes on those new inputs. When multiple candidate regexes are selected, REGAE generates input examples exhibiting behavioural discrepancies.

Strengths

- The paper has presented the motivation and problem statement well by providing an example in the introduction.
- Their interactive method is not only for programmer users; non-programmer users also can easily use it without special programming knowledge.
- The regex DSL they chose is very suitable for synthesizing because even professional users may make mistakes in its programming.
- The Figures of the examples and pseudo-codes in the paper help to understand the algorithm well.
- It has shown that this method is better than the previous methods by presenting the diagrams.

Weaknesses

- It is better to specify the sections and subsections with numbers to understand better the relationship between the different parts of the paper.
- It would be better to put the link to the implemented tool in the paper.
- This paper has many sections which haven't said anything new and only repeated the introductory words.
- It's better to merge the section Direct program manipulation to guide synthesis to Related works section.
- The pseudo-codes provided by the paper are long and do not initially give the reader a good feeling.