Simpler Implementation of Sketches Through Enhanced Expressiveness

Miguel Velez¹ and Armando Solar-Lezama²

¹Computer & Information Sciences, University of St. Thomas ²Electrical Engineering & Computer Science, Massachusetts Institute of Technology

The goal of software synthesis is to automatically implement a correct program from a high-level implementation. The Sketch engine is a tool that allows programmers to write a *sketch*, a partial program with some missing implementation, which the synthesizer uses to discover the missing code. The programmer uses expressions and place holders to indicate what variables and functions should be used to determine the missing code and where it should go. This allows developers to specify the final result that they desire and the computer is able to discover how to get that result. When programs increase in size, it can become very difficult for the programmer to track and specify all the variables and functions that the engine should use to deduce the missing implementation. Continuing with the aim of facilitating the development process of programmers using Sketch, we implemented some features that made the language more expressive and the engine more powerful for developers. Specifically, developers are now able to use a construct to indicate the synthesizer to use any of the variables, of a specific type, within scope, to derive the missing code. We also included support for lambda expressions, thus providing a lightweight method to create a function within another function that can be used locally or passed to a high-order function. Finally, we also added support for automatic casting of expressions in high-order function calls, which allows the programmer to pass an expression where a function parameter is expected, and the synthesizer then casts it as a lambda expression. All of these features made the Sketch engine easier to use through its enhanced expressiveness and increased the domain of applications that programmers can develop with this engine.