Project C.A.S.P. - Weekly Update 9

Ryan Tedeschi, Dylan Carson, Chris Bowlen

April 3, 2017

1   Development Progress

Ryan and Dylan met Monday, Tuesday, Friday, and Sunday of the past week to work on the program. Ryan was able to complete the scaffolding for the Standalone Application, as well as the graphical portion of the Outline Module. Ryan also performed a variety of small tasks regarding the function of the core module, including the way the Core Module communicates with the submodules, and also began work on the Translate Module. Dylan has almost completed the Analyze Module, with only some logic for calculating the complexity of loops to complete.

The standalone application allows the user to select the target module, the source language, source code, and additional module arguments. The constructed command is displayed to the user, and the user has the option to show the raw command line output upon execution. When the user executes the command, a new thread is created and the core program is invoked. Upon completion, the output is parsed and a function associated with the selected submodule is executed. In the case of the Outline module, the outline data is read from the core program and a flowchart is constructed.

The Analyze module uses the parse tree of the source code to create another tree containing the analyses of functions, loops, and other code artifacts. If a function is found that was not provided in the source code, a warning is created and returned to the consumer, and the unknown function is assumed to run in constant time (for the sake of completed analysis). All statements are assumed run in constant time, excluding function calls and loops.

2   Weekly Tasks

In the coming week, Dylan will be able to complete the Analyze Module and help with completion of the Translate Module. Ryan will continue work on the Translate Module, and create the Analyze portion of the Standalone Implementation upon Dylan's completion of the core implementation's Analyze Module. Ryan will also begin cleanup work on the program, including documentation and fixing small bugs that have been found.