Xinrong Zhao & Anny Kong
zhaox29 & yk57
2018/3/28

CSE 403 Project 1: Pitch

Motivation

Debugging is always a headache problem to all programmers. Software bugs can cost a billions of dollars, even cost lives. In 1997, software is the cause for 225 deaths during a jet crash caused by radar software, as mentioned in lecture. Unfortunately, java's built-in type system and checker does not support checking sufficient errors. To eliminate a bug, for instance, debugging for a NullPointerException, it often takes hours or days for a developer to stare at their codes, reason manually about code correctness. NullnessLight Checker could be a possible solution to such a painful situation.

NullnessLight Checker is a pluggable type checker in java that allows more compile-time checking and detects null pointer exception errors statically. It is a fast and easy-to-use. It employs the powerful analysis of the Checker Framework and its Nullness Checker, but omits some of its more confusing or expensive features. So the tradeoff of code clutter and compile time will not be a problem. Also, NullnessLight Checker is a supplement to the original type system. It will not disrupt programmers' workflow, because developers only need to add a few annotations to their program. Hence, there will not be a lot time spent on maintaining the annotations. As a result, it can be adopted to improve the code quality by finding and preventing nullness errors in a fast and easy way.

Approach

NullnessLight Checker will be written in Java and use Checker Framework but only focus on Nullness Checker.

NullnessLight Checker has multiple running options for users to choose. It is run with no option selected by default, suppressing all confusing and expensive features from Nullness Checker. Instead of using those features, NullnessLight Checker under the default mode will provide static analysis for programs based on strong assumptions.

Each option links to each feature mentioned above. Thus, users can gain better verification by selecting features they want to add. The mobility equipped by NullnessLight Checker is a successful trait which other light nullness checkers do not provide.

One unique option of NullnessLight Checker will be detecting bugs in programs using generics. As a result, NullnessLight Checker can be more competitive compared to other nullness detectors such as NullAway.

NullnessLight Checker will give more informative report about bugs detected. And it will also warn users for potential bugs uncaught by current options.

NullnessLight Checker will primarily function as an Eclipse plugin, so it is also a user-friendly tool. And it will be deployed through Git, where users can find useful information and give feedback.

Challenges and risks

It might post a challenge for us to fully understand Checker Framework, learn how to build a new type checker based on that, learn how to build an Eclipse plugin, and study the Checker Framework manual, especially the chapter with Nullness Checker. It might be a good idea to look over other existing nullness checkers like NullAway and FindBugs, to get a basic sense of how our checker should look like before starting to build. And we have to work hard to eliminate as many false warnings as possible.