Karlsruher Institut für Technologie

Institute of Information Security and Dependability (KASTEL)
Application-oriented Formal Verification
Prof. Dr. Bernhard Beckert

Bachelor / Master Thesis

Dependent Types for Java

Background. *Property types* are a Java type system developed at our chair. A property type consists of a base type and an annotation which is associated with a boolean property. In the example, the annotation `@Interval` is associated with the property `min <= v <= max`. We have developed a type checker that can prove the correctness of most simple parts of the program. For the difficult-to-prove parts, it emits a translation of the types to the *Java Modeling Language* (JML). This translated specification can then be proven in KeY, a deductive verification tool for Java co-developed at KIT. Combining our type checker with KeY allows our type system to be more powerful and expressive than comparable systems, while still being faster and easier than just using KeY for everything.

```
@Interval(1,arg) int foo(int arg) {
  if (arg <= 1) throw new Exception();
  return arg - 1;
}
```

Task. *Dependent types* are types that depend on a program variable. For example, a method with a parameter `arg` may have the return type `@Interval(1,arg)`, indicating that its result must be between `1` and `arg`.

The property type checker currently does not deal with such dependent types itself; it just translates them into JML for KeY to deal with. This is inefficient, since many uses of dependent types could be discharged using a faster, albeit less powerful, SMT ("satisfiability modulo theories") solver, which is a tool that can check the satisfiability or validity of certain first-order formulas. To check the correctness of the function `foo()` in the example, the SMT solver has to prove the validity of the following formula: $\neg(arg \leq 1) \rightarrow 1 \leq arg - 1 \leq arg$

Your task is to implement this SMT integration as well as to formalize it and prove its soundness in the theoretical type system.

Requirements. You have experience in Java programming and basic knowledge of logic and formal verification, e.g., from the *Formal Systems* lecture. Knowledge of type systems and dependent types is also useful.

Kontakt
Florian Lanzinger lanzinger@kit.edu Office 50.34R227