Prof. Dr. Reinhard Wilhelm
Prof. Dr. Sebastian Hack
Dipl.-Inform. Daniel Grund
Michael Jacobs, M.Sc.

Compiler Construction Project WS11/12

Project task E. Semantic Analysis

In this project phase you will implement the semantic analysis for MiniJava.

- Name analysis, i.e., associate identifiers with declarations. Implement the usual shadowing and hiding rules of Java.
- Type analysis, i.e., associate expressions with types. E.g. `2 < 3` would be associated with the type `boolean` and `6 * 9` with type `int`. Also ensure that the program is properly typed, e.g., `true + 5` is to be rejected.
- Implement your analyses in such a way that a statement, which syntactically can be a *PrintStatement*, is a print statement only if there is no other valid interpretation for `System` at the point of its occurrence.
- The single parameter of the *PrintStatement* is of type `int`.
- Type hierarchy analysis: Construct a tree for the declared classes for checking whether fields and methods are hidden and overwritten, respectively. MiniJava does not support method overloading, hence each method name withing a class must must have a unique signature.
- Check for unreachable statements. A statement is unreachable if each path in the program to that statement contains a `return` statement; e.g. `foo(5);` is unreachable in the *Block* `{return 0; foo(5);}`. In your check, do not evaluate conditions to exclude program paths, e.g. `foo(5);` is considered reachable in `{while (true) {} foo(5);}`.
- Also check whether the end of a non-`void` function can be reached without encountering a `return` statement. This check can be implemented easily by leveraging the unreachable statement check.
- Invalid expression statements: Check for statements that are not valid statements in Java (e.g. `42;`) as well as illegal assignments (e.g. `a + b = 47;`). The root of a valid *ExpressionStatement*(-tree) is an assignment, a *MethodInvocationExpression*, or a *NewObjectExpression*.

You do not need to add another switch, just extend the functionality of `--ast` and `--astmin`, i.e. they will reject more inputs due to the semantic checks. Please check in your solution into your repository until 2011-12-08, 12:00, noon.