FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA

Institute for Informatics
System Software Group
Prof. Dr. Clemens Grelck
Compiler Construction
November 16, 2023

Assignment 3

Familiarisation with Compiler Framework
Intermediate Representation / Abstract Syntax Tree

Assignment 3.1: Compiler Construction Framework: lookup tables

Implement a compiler traversal that counts the number of occurrences of each identifier (left or right hand side). Again, the `info` structure shall be used to collect the information, and the use of global variables is not permitted.

In contrast to Assignment 2.2 you do not a-priori know the number of different counters needed, which makes this assignment truly different.

Instead of a fixed set of counters, make use of the lookup table provided with the framework. These lookup tables can store associations (i.e. key-value pairs) between character strings (or pointers) and heap-allocated objects, e.g. counters.

Print the inferred information at the end of the traversal, i.e. inside the root node handler. Do not store the lookup table in the abstract syntax tree.

Assignment 3.2: Abstract Syntax Tree Definition

Design an appropriate intermediate representation (abstract syntax tree) for your CiviC compiler by extending the file `ast.xml` coming with the compiler framework.

Focus on the language core first and only then consider the extensions. If you need further attribute types (usually enumeration types), provide their implementations alongside.

For this assignment, submit the following files:

- `ast.xml` as written by you,
- `ast.html` as generated by the framework in the `doc` directory,
- `ast.png` as generated by the framework in the `doc` directory.

Note:

We will discuss the design of syntax trees during the lab on Thursday, November 30, and provide you with an example solution. It is up to you whether you build your compiler on your own syntax tree design or make use of ours.

Assignment due date: Monday, November 27, 2023