

Semantic Analysis and Type Checking

Using JavaCC, JTB and Visitor design pattern

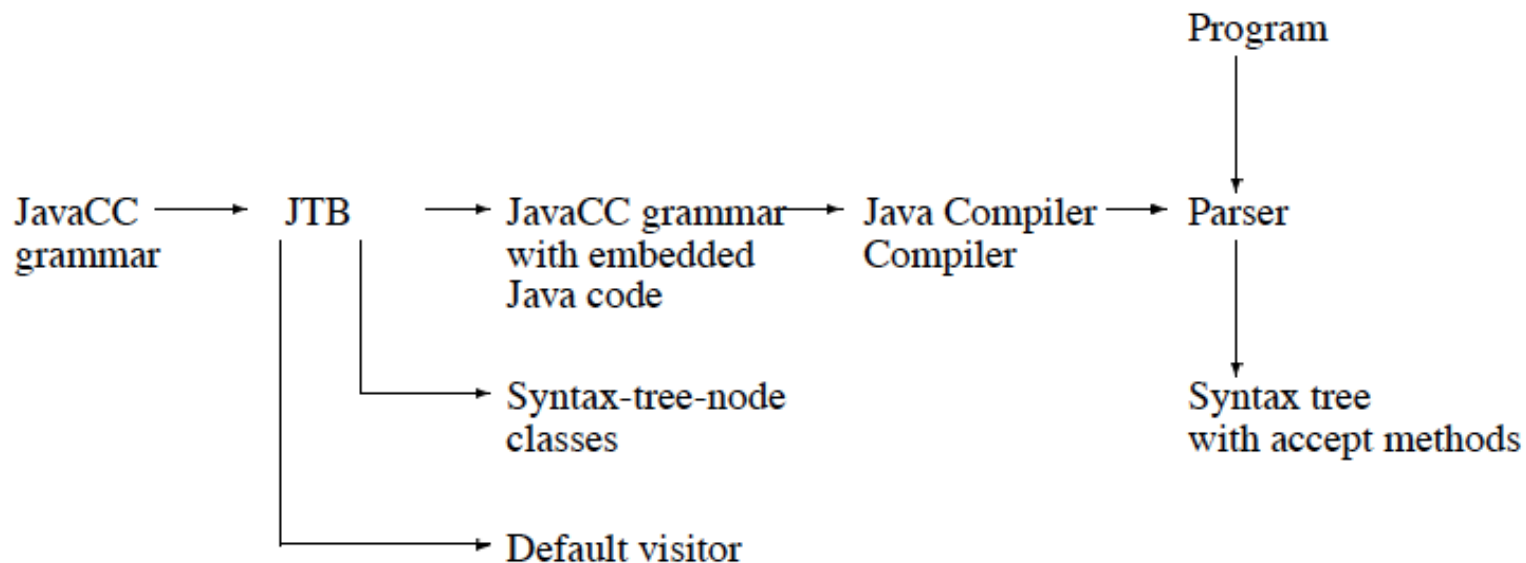
JavaCC (Java Compiler Compiler)

- parser generator and lexical analyzer generator
- generates top-down LL(k) parsers
- reads a grammar specification and converts it to a Java program that can recognize matches to the grammar

JTB (Java Tree Builder)

- takes a plain JavaCC grammar file (.jj) as input and automatically generates the following:
 1. The subdirectory ***syntaxtree*** containing a set of syntax tree classes based on the productions in the grammar (plus 8 automatically generated classes), utilizing the Visitor design pattern
 2. The subdirectory ***visitor*** containing visitor classes (4 interfaces, and 4 depth-first visitors whose default methods simply visit the children of the current node)
 3. A JavaCC grammar with the proper annotations to build the syntax tree during parsing

Big Picture



Example (jtb-calc directory)

```
java -jar jtb132di.jar calc.jj
```

```
javacc calc-jtb.jj
```

```
javac Main.java
```

```
java Main test_file
```

JTB output

- "calc-jtb.jj" grammar
- Syntax tree Java source files generated to directory "syntaxtree"
- Visitors generated to directory "visitor":
 - "GJVisitor.java"
 - "Visitor.java"
 - "GJNoArguVisitor.java"
 - "GJVoidVisitor.java"
 - "GJDepthFirst.java"
 - "DepthFirstVisitor.java"
 - "GJNoArguDepthFirst.java"
 - "GJVoidDepthFirst.java"

JavaCC output

- **TokenMgrError** is a simple error class; it is used for errors detected by the lexical analyser and is a subclass of Throwable.
- **ParseException** is another error class; it is used for errors detected by the parser and is a subclass of Exception and hence of Throwable.
- **Token** is a class representing tokens. Each Token object has an integer field kind that represents the kind of the token (PLUS, NUMBER, etc) and a String field image, which represents the sequence of characters from the input file that the token represents.
- **JavaCharStream** delivers characters to the lexical analyser.
- **CalcParserConstants** is an interface that defines a number of classes used in both the lexical analyser and the parser, associating token classes with symbolic names.
- **CalcParserTokenManager** is the lexical analyser.
- **CalcParser** is the parser.