Cool: Lexical Analysis

Greta Yorsh

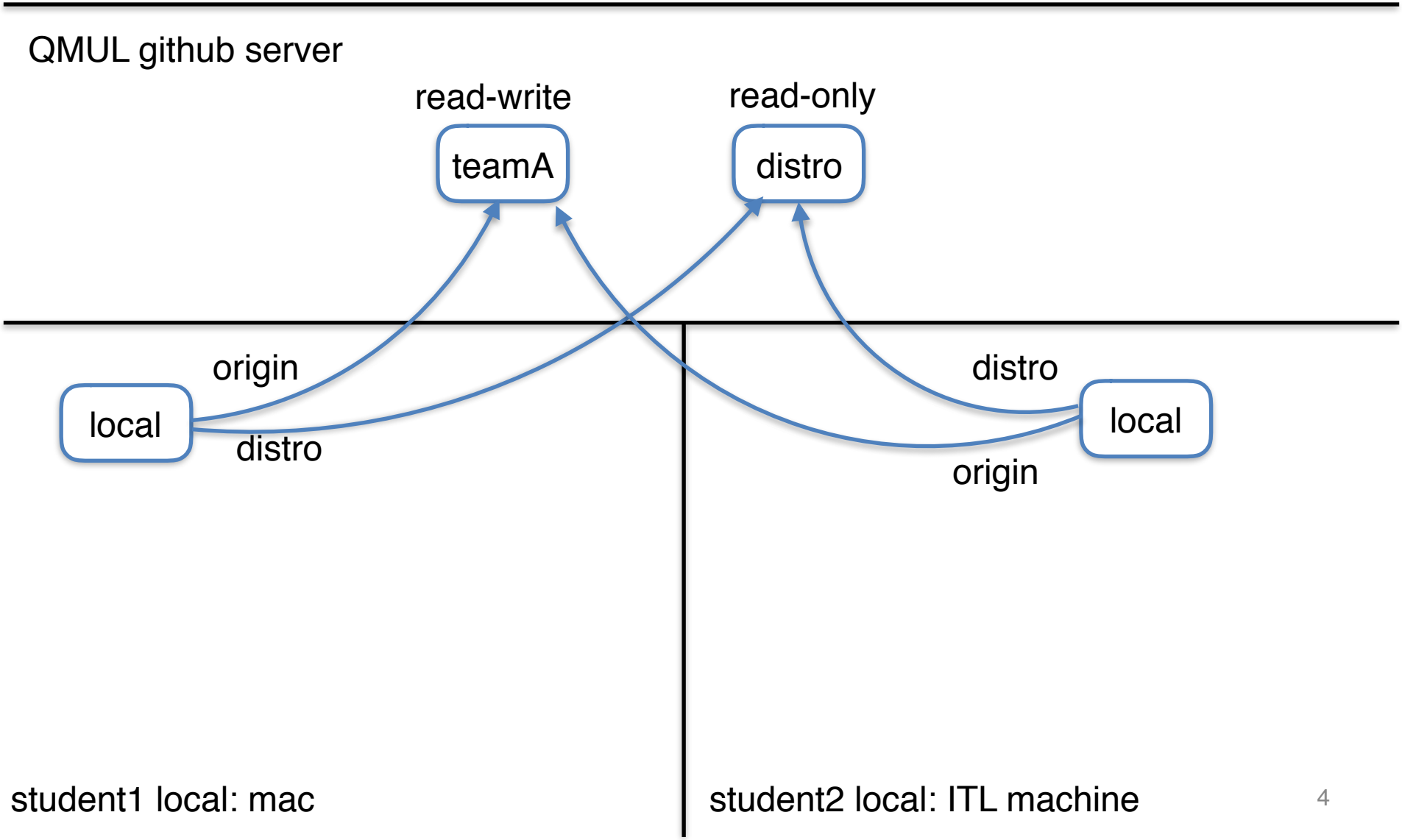
Marking PA1-PA3

- Team sign up
- PA1: START EARLY!
- PA1 marking guidelines
  - 80% testsuite
  - 20% code review
  - testsuite hurdle: 90% by submission deadline
- Feedback and final submission deadline
- Independent components

Today

- Slides and Q-review of tutorials
- GitHub repositories for PA1-PA3
- Antlr4 workflow and example
- PA1 Setup
- Testsuites and testscripts
- Inspect outputs not just PASS/FAIL counts!!

Repositories

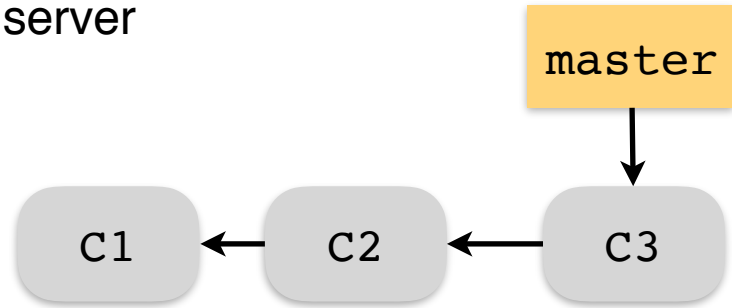


Branches

QMUL github server

read-only

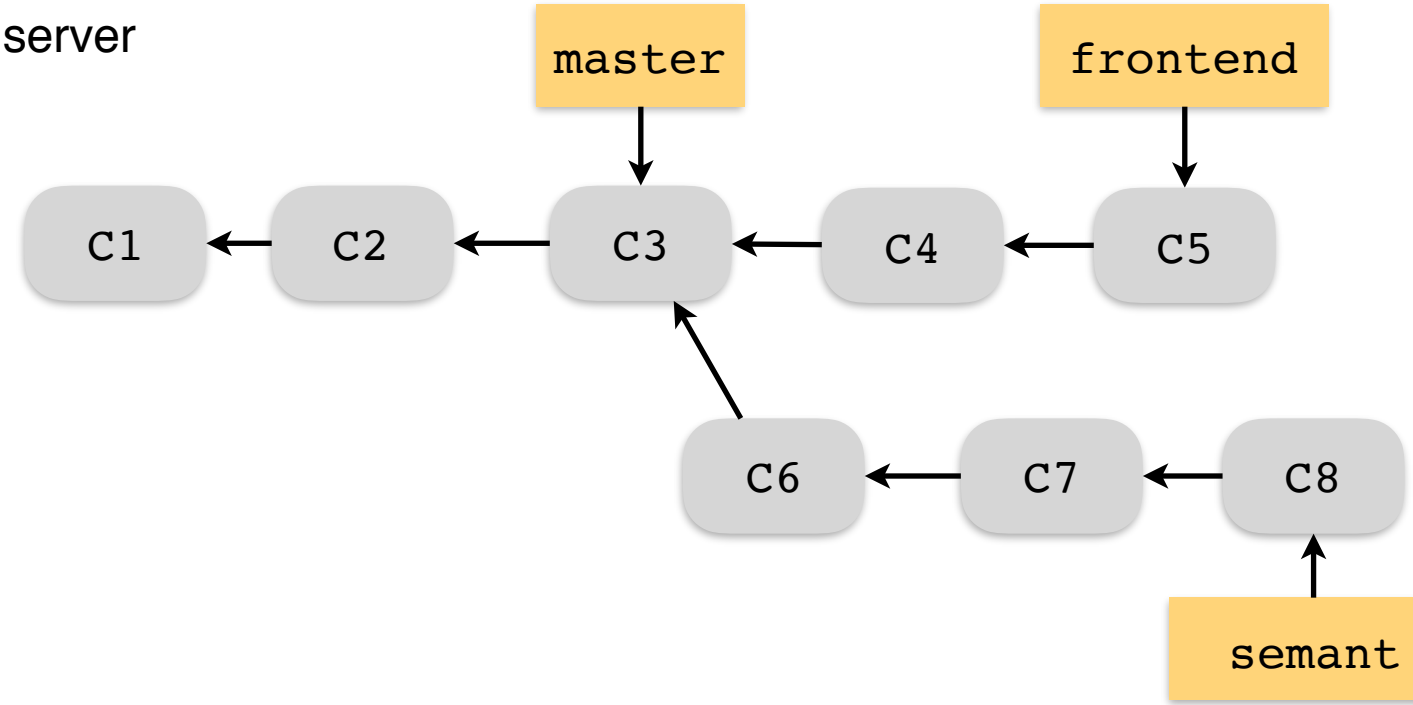
distro



QMUL github server

read-write

teamA



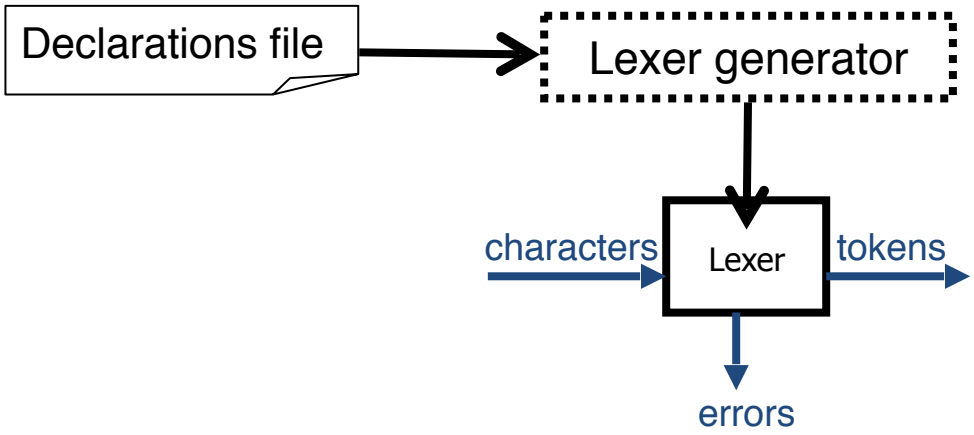
Integrated Development Environment (IDE)

- IntelliJ - free for students
- NetBeans
- Eclipse

- Antlr4 plugin for each IDE

Lexer Generation

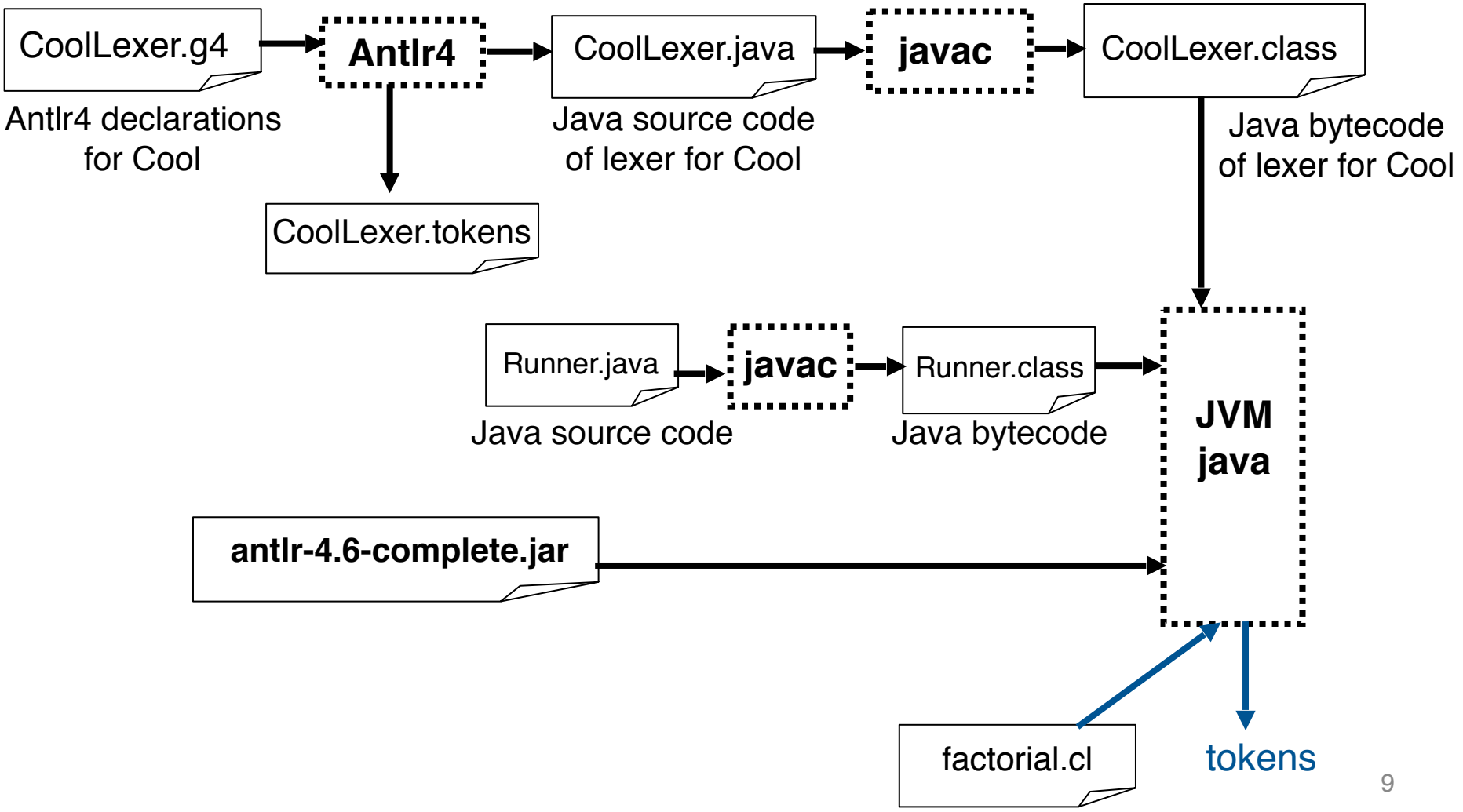
- Terminology: lexical analysis = lexical analyzer = lexer = scanner
- Describe tokens as regular expressions
- Decide which attributes are saved for each token
- Construction is done automatically by common tools
- **Lexer generator** automatically creates **lexer** from **declarations file**
  - lexer generator builds DFA table
  - lexer simulates (runs) the DFA on a given input
- We will use **Antlr** lexer generator



Declarations file in Antlr4

```
lexer grammar ExampleLexer;

INT                 : DIGIT+ ;
fragment
DIGIT               : [0-9] ;

ID                  :  [a-z] IDENTIFIER* ;
fragment
LETTER              :  [a-zA-Z];

COMMENT    : '/*' .*? '*/'   -> skip;

WHITESPACE : (' ' | '\n' | '\r' | '\t' | '\u000B')+ -> skip;

ERROR               : . ;
```

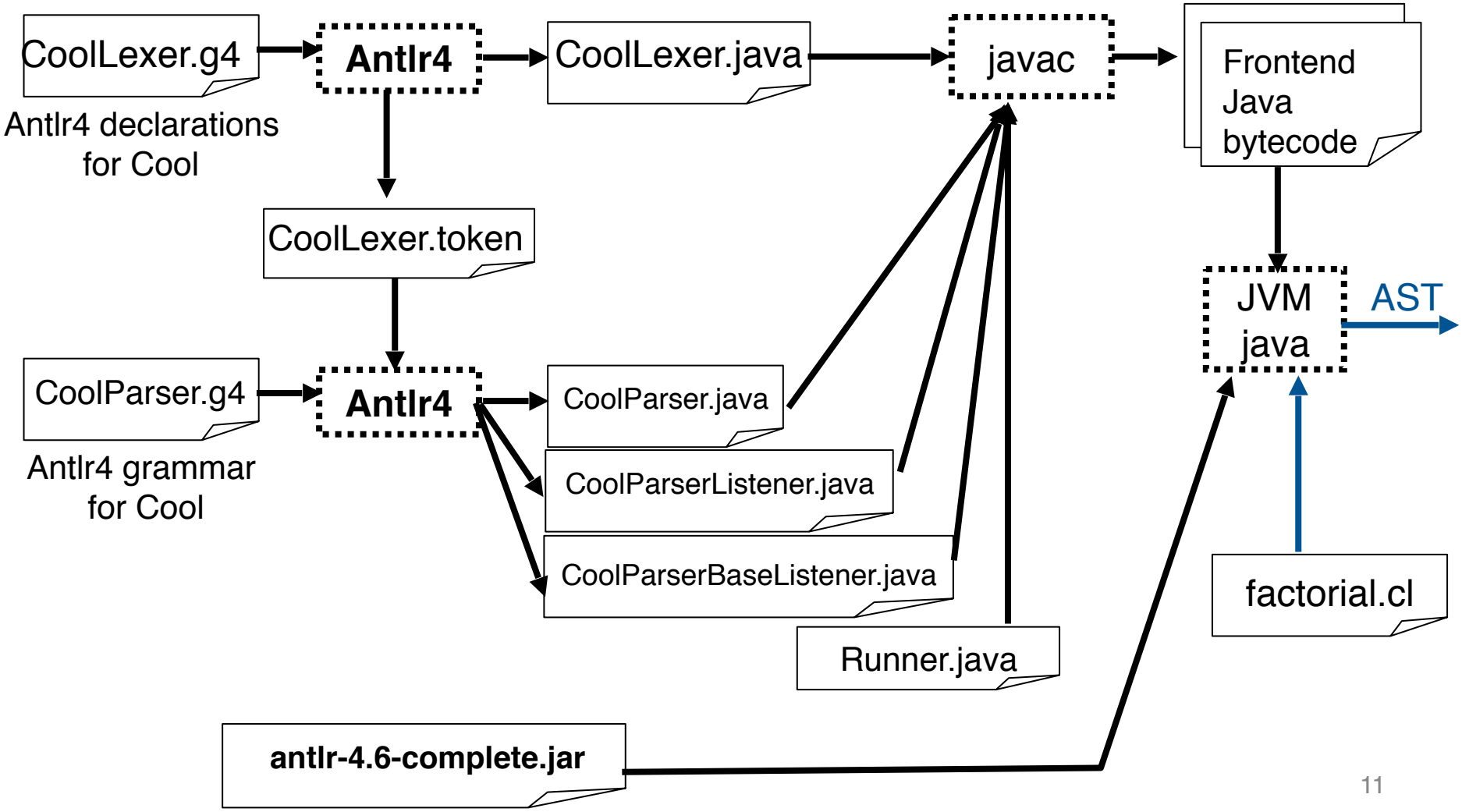
Antlr: build and run a lexer



Runner

```
import org.antlr.v4.runtime.*;
import org.antlr.v4.runtime.tree.*;
public class Runner {
  public static void main(String[] args) throws Exception {
    ANTLRInputStream input = new ANTLRInputStream(System.in);
    CoolLexer lexer = new CoolLexer(input);
    CommonTokenStream tokens = new CommonTokenStream(lexer);
    for (Token t : tokens.getTokens()) {
          System.out.println(t.toString());
    }
  }
}
```

Antlr: build and run a parser



Runner

```
import org.antlr.v4.runtime.*;
import org.antlr.v4.runtime.tree.*;
public class Runner {
  public static void main(String[] args) throws Exception {
    ANTLRInputStream input = new ANTLRInputStream(System.in);
    CoolLexer lexer = new CoolLexer(input);
    CommonTokenStream tokens = new CommonTokenStream(lexer);
    CoolParser parser = new CoolParser(tokens);
    ParseTree tree = parser.init();
   System.out.println(tree.toStringTree(parser));
  }
}
```

Grammar file in Antlr4

```
grammar ExampleParser;


program : method* EOF;

method : TYPE ID LPAREN formals RPAREN
body;


TYPE : [A-Z]LETTER*;
```