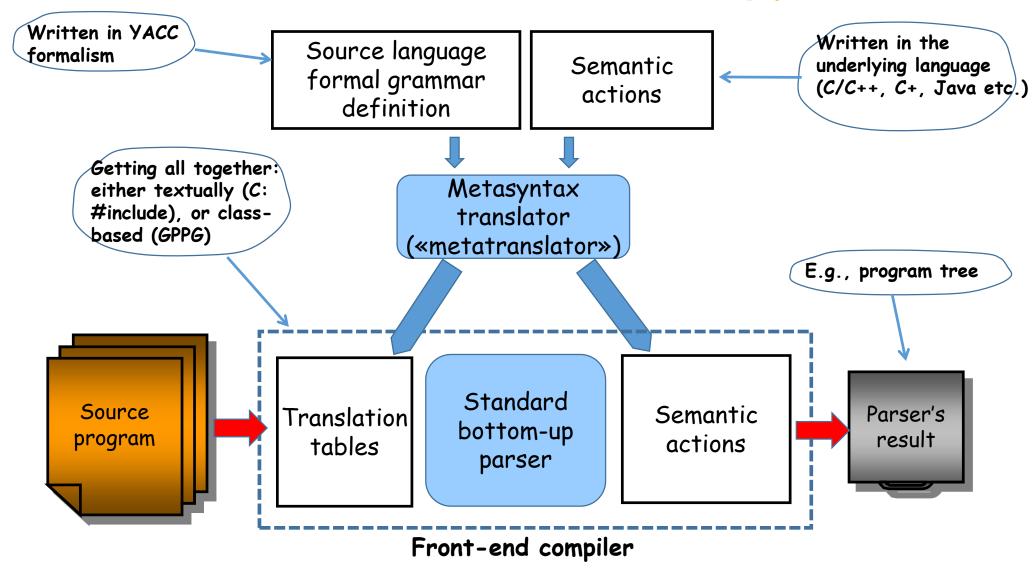
Compiler Construction: Practical Introduction

Case Study: The Complete Bison-based Java parser

Eugene Zouev

Fall Semester 2024 Innopolis University

Yacc based technology



The Polystat Project The Java Part

- Source language: Java
- · Implementation based on Bison.
- · Implementation language: Java
- Scanner: written from scratch.
- Parser: Bison grammar with semantic actions.
- Result: AST.

1. How scanner is implemented

- Token codes
- Token structure
- Reading source program
- Detecting tokens
- Token definitions in the grammar
- Token type.

yylex() - The wrapper over get() - Interface function for the parser - Returns the token code

These functions are defined by the scanner and used by the parser

getLVal()

- Returns the last taken token

get()

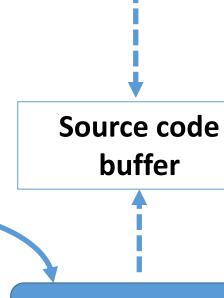
- Skips comments
- Returns the current token

forget()

- "Forgets" (nullifies) the current token

getToken()

- Skips whitespaces
- Detects the next token
- Returns the next token



readFile()

- Returns the next character from the source string

getChar()

- Keeps the current position

forgetChar()

2. How Grammar is implemented

- Grammar from the Java Reference
- Bison-complied Java grammar
- Types for grammar elements.

3. How AST is implemented

- The approach:

 A Java class for each grammar nonterminal.
- The structure of the typical tree node.
- How repetitions/lists are implemented.