

Plan for Today

Structure of the MiniJava Compiler

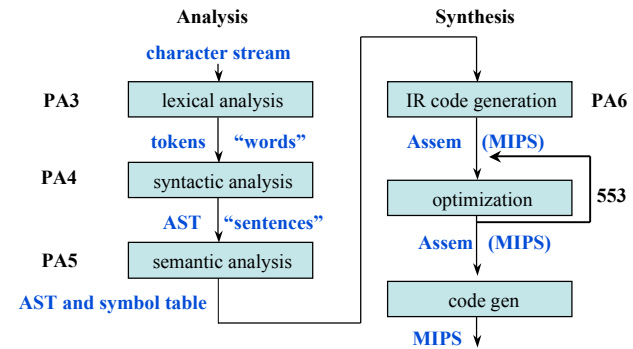
- Lexer
- Interface between lexer and parser: Symbol and TokenValue
- Parser
- Interface between parser and semantic analysis: ast.node.*
- Interface between semantic analysis and code generation: symbol table

CS453 Lecture

MiniJava Intro

1

Structure of the MiniJava Compiler



CS453 Lecture

MiniJava Intro

2

Specifying Tokens with JFlex

JFlex example input file:

```
package mparser;
import java_cup.runtime.Symbol;

%%
%line
%char
%cup
%public

%eofval{
    return new Symbol(sym.EOF, new
    TokenValue("EOF", yline, ychar));
}eofval;
```

```
LETTER=[A-Za-z]
DIGIT=[0-9]
UNDERSCORE="_"
LETT_DIG_UND={LETTER}|{DIGIT}|{UNDERSCORE}
ID={LETTER}{LETT_DIG_UND}*

%%
"&&" { return new Symbol(sym.AND, new
    TokenValue(yytext(), yline, yychar)); }

"boolean" {return new
    Symbol(sym.BOOLEAN,...

{ID} { return new Symbol(sym.ID, new ...
```

CS453 Lecture

MiniJava Intro

3

Specifying Grammar with JavaCUP

JavaCUP example input file:

```
package mparser;
import java_cup.runtime.*;
import ast.node.*;
import mparser.TokenValue

terminal AND, ASSIGN, INT;
terminal mparser.TokenValue
    NUMBER;
...
non terminal Program program;

non terminal List<ClassDecl>
    class_decl_list;

non terminal MainClass
    main_class;
```

```
start with program;

program ::=
    main_class:m class_decl_list:l
    {: RESULT = new Program(m,l); :}
    ;

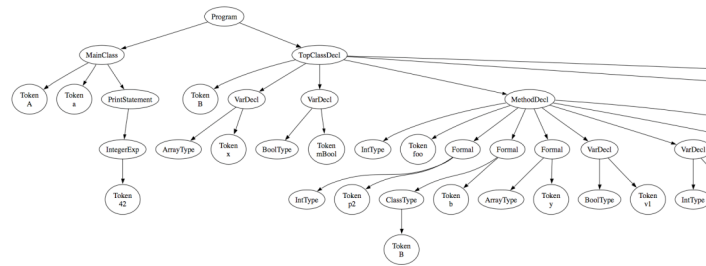
exp ::=
    | NUMBER:n
    {: Token token = new Token(n.text,
    n.line, n.pos);
    RESULT = new IntegerExp( token );
    :}
    ...
```

CS453 Lecture

MiniJava Intro

4

Abstract Syntax Tree for Memory Layout Example

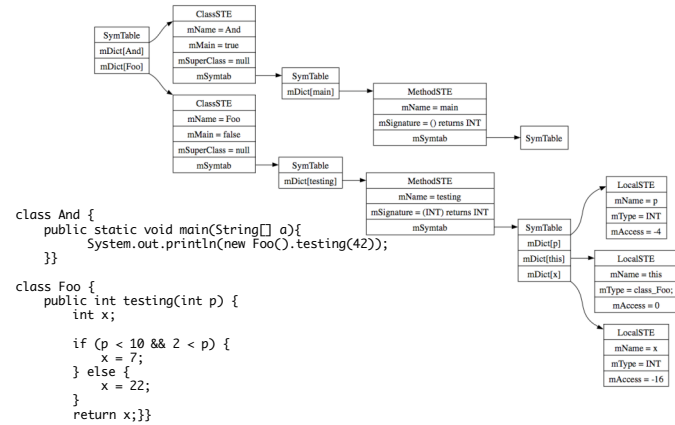


CS453 Lecture

MiniJava Intro

5

Example Symbol Table



CS453 Lecture

MiniJava Intro

6

Assem(MIPS)

```

        .text
main:
main_framesize=24
main_paramsNregsaves=8
    sw $ra, 0($sp)
    subu $sp, $sp, 4
    sw $fp, 0($sp)
    subu $sp, $sp, 4
    addu $fp, $sp, main_paramsNregsaves
    subu $sp, $fp, main_framesize
    # ExpCALL
    # ExpCONST
    li $t2, 0
    sw $t2, -8($fp)
    lw $t2, -8($fp)
    # push parameter onto stack
    sw $t2, 0($sp)
    subu $sp, $sp, 4
    jal _halloc
    #StmMOVE(ExpTEMP(t1), e)
    move $t0, $v0
    sw $t0, -12($fp)

done1:
    lw $ra, 0($fp)
    move $t0, $fp
    lw $fp, -4($fp)
    move $sp, $t0
    jr $ra

        # ExpCALL
        lw $t2, -12($fp)
        # push parameter onto stack
        sw $t2, 0($sp)
        subu $sp, $sp, 4
        # ExpCONST
        li $t2, 42
        sw $t2, -16($fp)
        lw $t2, -16($fp)
        # push parameter onto stack
        sw $t2, 0($sp)
        subu $sp, $sp, 4
        jal Foo.testing
        #StmMOVE(ExpTEMP(t1), e)
        move $t0, $v0
        sw $t0, -20($fp)
        # ExpCALL
        lw $t2, -20($fp)
        # push parameter onto stack
        sw $t2, 0($sp)
        subu $sp, $sp, 4
        jal _printint
        # epilogue

```

CS453 Lecture

MiniJava Intro

7