CS 4110 Project #2 Report

Andrew Sison and Zachary Martin

Context-Free Grammar

1. program ::= decl declplus

;

1. declplus ::= decl declplus

|

;

1. decl ::= variable SEMICOLON

| functiondecl

| classdecl

| interfacedecl

;

1. variable ::= type ID

| ID ID

;

1. type ::= INT

| DOUBLE

| BOOLEAN

| STRING

| type LEFTBRACKET RIGHTBRACKET

;

1. functiondecl ::= type ID LEFTPAREN formals RIGHTPAREN stmtblock

| VOID ID LEFTPAREN formals RIGHTPAREN stmtblock

| ID ID LEFTPAREN formals RIGHTPAREN stmtblock

;

1. formals ::= variable variablecommplus

|

;

1. variablecommplus ::= COMMA variable variablecommplus

|

;

1. classdecl ::= CLASS ID EXTENDS ID IMPLEMENTS ID idcommplus LEFTBRACE fieldstar RIGHTBRACE

| CLASS ID EXTENDS ID IMPLEMENTS ID idcommplus LEFTBRACE RIGHTBRACE

| CLASS ID EXTENDS ID LEFTBRACE fieldstar RIGHTBRACE

| CLASS ID EXTENDS ID LEFTBRACE RIGHTBRACE

| CLASS ID IMPLEMENTS ID idcommplus LEFTBRACE fieldstar RIGHTBRACE

| CLASS ID IMPLEMENTS ID idcommplus LEFTBRACE RIGHTBRACE

| CLASS ID LEFTBRACE fieldstar RIGHTBRACE

| CLASS ID LEFTBRACE RIGHTBRACE

;

1. fieldstar ::= field fieldstar

| field

;

1. idcommplus ::= COMMA ID idcommplus

|

;

1. field ::= variable SEMICOLON

| functiondecl

;

1. interfacedecl ::= INTERFACE ID LEFTBRACE prototypestar RIGHTBRACE

| INTERFACE ID LEFTBRACE RIGHTBRACE

;

1. prototypestar ::= prototype prototypestar // FIXED '\*' PROBLEM

| prototype

;

1. prototype ::= type ID LEFTPAREN formals RIGHTPAREN SEMICOLON

| VOID ID LEFTPAREN formals RIGHTPAREN SEMICOLON

;

1. stmtblock ::= LEFTBRACE variabledeclstar stmtstar RIGHTBRACE

| LEFTBRACE stmtstar RIGHTBRACE

| LEFTBRACE variabledeclstar RIGHTBRACE

| LEFTBRACE RIGHTBRACE

;

1. variabledeclstar ::= variabledeclstar variable SEMICOLON

| variable SEMICOLON

;

1. stmtstar ::= stmt stmtstar

| stmt

;

1. stmt ::= expr SEMICOLON

| SEMICOLON

| ifstmt

| whilestmt

| forstmt

| breakstmt

| returnstmt

| printstmt

| stmtblock

;

1. ifstmt ::= IF LEFTPAREN expr RIGHTPAREN stmt ELSE stmt

| IF LEFTPAREN expr RIGHTPAREN stmt

;

1. whilestmt ::= WHILE LEFTPAREN expr RIGHTPAREN stmt

;

1. forstmt ::= FOR LEFTPAREN expr SEMICOLON expr SEMICOLON expr RIGHTPAREN stmt

| FOR LEFTPAREN expr SEMICOLON expr SEMICOLON RIGHTPAREN stmt

| FOR LEFTPAREN SEMICOLON expr SEMICOLON expr RIGHTPAREN stmt

| FOR LEFTPAREN SEMICOLON expr SEMICOLON RIGHTPAREN stmt

;

1. breakstmt ::= BREAK SEMICOLON

;

1. returnstmt ::= RETURN expr SEMICOLON

| RETURN SEMICOLON

;

1. printstmt ::= PRINTLN LEFTPAREN expr exprcommplus RIGHTPAREN SEMICOLON

;

1. exprcommplus ::= COMMA expr exprcommplus

|

;

1. expr ::= lvalue ASSIGNOP expr

| constant

| THIS

| call

| LEFTPAREN expr RIGHTPAREN

| expr PLUS expr

| expr MINUS expr

| expr MULTIPLICATION expr

| expr DIVISION expr

| expr MOD expr

| MINUS expr

| expr LESS expr

| expr LESSEQUAL expr

| expr GREATER expr

| expr GREATEREQUAL expr

| expr EQUAL expr

| expr NOTEQUAL expr

| expr AND expr

| expr OR expr

| NOT expr

| READLN LEFTPAREN RIGHTPAREN

| NEW LEFTPAREN ID RIGHTPAREN

| NEWARRAY LEFTPAREN INTCONSTANT COMMA type RIGHTPAREN

| lvaluenotid

| ID

;

1. lvalue ::= ID

| lvaluenotid

;

1. call ::= ID LEFTPAREN actuals RIGHTPAREN

| ID PERIOD ID LEFTPAREN actuals RIGHTPAREN

;

1. actuals ::= expr exprcommplus

|

;

1. constant ::= INTCONSTANT

| DOUBLECONSTANT

| STRINGCONSTANT

| BOOLEANCONSTANT

| NULL

;

32. lvaluenotid ::= lvalue LEFTBRACKET expr RIGHTBRACKET

| lvaluenotid LEFTBRACKET expr RIGHTBRACKET

| lvalue PERIOD ID

S/R and R/R Conflicts

Warning : \*\*\* Shift/Reduce conflict found in state #57

between variabledeclstar ::= (\*)

and type ::= (\*) ID

under symbol ID

Resolved in favor of shifting.

Warning : \*\*\* Shift/Reduce conflict found in state #58

between variabledeclstar ::= (\*)

and type ::= (\*) ID

under symbol ID

Resolved in favor of shifting.

These conflicts were fixed by making the variabledeclstar nonterminal e-free by adding more productions to the stmtblock nonterminal. Also added another production in variabledeclstar calling a single variabledecl. This fixed the conflicts here.

Warning : \*\*\* Shift/Reduce conflict found in state #181

between ifstmt ::= IF LEFTPAREN expr RIGHTPAREN stmt (\*)

and ifstmt ::= IF LEFTPAREN expr RIGHTPAREN stmt (\*) ELSE stmt

under symbol ELSE

Resolved in favor of shifting.

This conflict was because of the dangling else problem. We had to look up how to fix this issue and found that by making the ELSE terminal the highest precedence value in the list it would fix the conflict. This conflict was fixed.

Warning : \*\*\* Reduce/Reduce conflict found in state #102

between type ::= ID (\*)

and lvalue ::= ID (\*)

under symbols: {LEFTBRACKET}

Resolved in favor of the first production.

Warning : \*\*\* Shift/Reduce conflict found in state #102

between type ::= ID (\*)

under symbol LEFTBRACKET

Resolved in favor of shifting.

Warning : \*\*\* Shift/Reduce conflict found in state #102

between lvalue ::= ID (\*)

under symbol LEFTBRACKET

Resolved in favor of shifting.

These conflicts occurred because of the lone ID production in the lvalue nonterminal. We fixed it by removing all productions of lvalue except for ID and creating a new nonterminal lvaluenotid that would have the other productions previously in lvalue. These conflicts were fixed.

Warning : \*\*\* Shift/Reduce conflict found in state #87  
between variabledeclstar ::= variabledecl ()and type ::= () ID  
under symbol ID  
Resolved in favor of shifting.

This conflict was fixed by adding a precedence for the ID terminal.

In the end all 6 of these conflicts were fixed and none remained unsolved.

Test Cases

Expected Outcome for input # 1,2,3:

A close up of text on a white background

Description automatically generated

Program Result for # 1,2,3:

Result #1:

DOUBLE [shift]

[reduce 602]

ID [shift]

DOUBLE [shift]

[reduce 602]

ID [shift]

COMMA [shift]

[reduce 501] [reduce 902]

[reduce 501] [reduce 901]

LEFTBRACE [shift]

RIGHTBRACE [shift]

VOID [shift]

ID [shift]

LEFTPAREN [shift]

[reduce 801]

RIGHTPAREN [shift]

[reduce 1704]

[reduce 702]

[reduce 302] [reduce 202]

[accept]

Result #2:

INT [shift]

instead expected token classes are []

[reject]

This rejects because there is no ASSIGNOP in variabledecl.

Result #3:

Syntax error

Couldn't repair and continue parse

java.lang.Exception: Can't recover from previous error(s)

at java\_cup.runtime.lr\_parser.report\_fatal\_error(lr\_parser.java:392)

at java\_cup.runtime.lr\_parser.unrecovered\_syntax\_error(lr\_parser.java:539)

at java\_cup.runtime.lr\_parser.parse(lr\_parser.java:731)

at main.Main.main(Main.java:43)

instead expected token classes are [DOUBLE, INTERFACE, STRING, VOID, INT, ID]

[reject]

This rejects because 5 is an INTCONSTANT and not an ID.

Expected Outcome for # 4,6:

A close up of text on a white background

Description automatically generated

Program Result for # 4,6:

Result #4:

ID [shift]

ID [shift]

PERIOD [shift]

ID [shift]

LEFTPAREN [shift]

[reduce 3102]

RIGHTPAREN [shift]

[reduce 3002]

[reduce 2902]

ASSIGNOP [shift]

[reduce 2805]

[reduce 2801]

SEMICOLON [shift]

[reduce 2001]

LEFTBRACE [shift]

[reduce 1902]

RIGHTBRACE [shift]

VOID [shift]

ID [shift]

LEFTPAREN [shift]

[reduce 802]

RIGHTPAREN [shift]

[reduce 1702]

[reduce 702]

[reduce 302] [reduce 202]

[accept]

Result #6:

INT [shift]

[reduce 601]

LEFTBRACKET [shift]

RIGHTBRACKET [shift]

[reduce 605]

LEFTBRACKET [shift]

RIGHTBRACKET [shift]

[reduce 605]

LEFTBRACKET [shift]

RIGHTBRACKET [shift]

[reduce 605]

ID [shift]

[reduce 501]

[reduce 301] [reduce 202]

[accept]

Expected Outcome for # 5,7,8:

A close up of text on a white background

Description automatically generated

Program Result for # 5,7,8:

Result #5:

DOUBLE [shift]

[reduce 602]

ID [shift]

[reduce 501] [reduce 902]

instead expected token classes are [LEFTPAREN, LEFTBRACKET, ID]

[reject]

This test input is supposed to accept however it does not possibly still issues in the grammar.

Result #7:

ID [shift]

Syntax error

instead expected token classes are []

Couldn't repair and continue parse

java.lang.Exception: Can't recover from previous error(s)

at java\_cup.runtime.lr\_parser.report\_fatal\_error(lr\_parser.java:392)

at java\_cup.runtime.lr\_parser.unrecovered\_syntax\_error(lr\_parser.java:539)

at java\_cup.runtime.lr\_parser.parse(lr\_parser.java:731)

at main.Main.main(Main.java:35)

[reject]

This rejects because no productions from decl produces anything with ASSIGNOP.

Result #8:

Syntax error

Couldn't repair and continue parse

instead expected token classes are [READLN, THIS, NULL, MINUS, NOT, LEFTPAREN, INTCONSTANT, ID, STRINGCONSTANT, DOUBLECONSTANT, BOOLEANCONSTANT]

java.lang.Exception: Can't recover from previous error(s)

at java\_cup.runtime.lr\_parser.report\_fatal\_error(lr\_parser.java:392)

at java\_cup.runtime.lr\_parser.unrecovered\_syntax\_error(lr\_parser.java:539)

at java\_cup.runtime.lr\_parser.parse(lr\_parser.java:731)

at main.Main.main(Main.java:35)

[reject]

This rejects because there at least needs to be one expr in the middle of a functiondecl.

Expected Outcome for #9:

A close up of a map

Description automatically generated

Program Result for #9:

Result #9:

ID [shift]

ID [shift]

[reduce 2825]

GREATER [shift]

[reduce 2825]

ID [shift]

ID [shift]

[reduce null]

SEMICOLON [shift]

ID [shift]

ID [shift]

Syntax error

instead expected token classes are [NEW, NEWARRAY, PRINTLN, READLN, RETURN, THIS, WHILE, BREAK, NULL, MINUS, NOT, SEMICOLON, LEFTPAREN, LEFTBRACE, RIGHTBRACE, INTCONSTANT, ID, STRINGCONSTANT, DOUBLECONSTANT, BOOLEANCONSTANT]

[reject]

This rejects because there needs to be braces ({}) around “double a;” in which there isn’t in this input.

Expected Outcome for #10,11:

A close up of text on a white background

Description automatically generated

Program Result for #10,11:

Result #10:

BOOLEAN [shift]

[reduce 603]

ID [shift]

[reduce 501]

LEFTBRACE [shift]

RIGHTBRACE [shift]

ID [shift]

ID [shift]

LEFTPAREN [shift]

[reduce 802]

RIGHTPAREN [shift]

[reduce 1704]

[reduce 703]

DOUBLE [shift]

[reduce 602]

ID [shift]

[reduce 501]

LEFTBRACE [shift]

RIGHTBRACE [shift]

ID [shift]

ID [shift]

LEFTPAREN [shift]

[reduce 802]

RIGHTPAREN [shift]

[reduce 1704]

[reduce 703]

[reduce 1302]

[reduce 1301] [reduce 1102]

[reduce 1302] [reduce 1101]

[reduce 1301] [reduce 1101]

CLASS [shift]

ID [shift]

LEFTBRACE [shift]

[reduce 1101]

RIGHTBRACE [shift]

[reduce 1007]

[reduce 303] [reduce 202]

[accept]

Result #11:

INT [shift]

Syntax error

instead expected token classes are [FOR, IF, NEW, NEWARRAY, PRINTLN, READLN, RETURN, STRING, THIS, WHILE, BREAK, NULL, INT, MINUS, NOT, SEMICOLON, LEFTPAREN, LEFTBRACE, RIGHTBRACE, INTCONSTANT, ID, STRINGCONSTANT, DOUBLECONSTANT, BOOLEANCONSTANT]

[reject]

This rejects because there cannot be a classdecl within a functiondecl.

Expected Outcome for #12:

A close up of text on a white background

Description automatically generated

Program Result for #12:

Result #12:

INT [shift]

[reduce 601]

ID [shift]

[reduce 501]

SEMICOLON [shift]

[reduce 401]

DOUBLE [shift]

Syntax error

instead expected token classes are [MINUS, MULTIPLICATION, DIVISION, MOD, LESS, GREATER, LESSEQUAL, GREATEREQUAL, EQUAL, NOTEQUAL, AND, OR, SEMICOLON]

Couldn't repair and continue parse

java.lang.Exception: Can't recover from previous error(s)

at java\_cup.runtime.lr\_parser.report\_fatal\_error(lr\_parser.java:392)

at java\_cup.runtime.lr\_parser.unrecovered\_syntax\_error(lr\_parser.java:539)

at java\_cup.runtime.lr\_parser.parse(lr\_parser.java:731)

at main.Main.main(Main.java:35)

[reject]

This rejects because lvalue cannot produce THIS.

A close up of text on a white background

Description automatically generatedExpected Outcome for #13,14:

A close up of text on a white background

Description automatically generated

A close up of text on a white background

Description automatically generated

Program Results for #13,14:

Result #13:

INT [shift]

INT [shift]

[reduce 601]

ID [shift]

[reduce 501] [reduce 902]

Syntax error

instead expected token classes are [LEFTPAREN, LEFTBRACKET]

[reject]

This rejects because “flag = true” cannot be produced from the field productions.

Result #14:

INT [shift]

INT [shift]

[reduce 601]

ID [shift]

[reduce 501] [reduce 902]

Syntax error

instead expected token classes are [LEFTPAREN, LEFTBRACKET]

[reject]

This rejects because “flag = true” cannot be produced from the field productions.

Expected Outcome for #15:

A close up of a map

Description automatically generated

Program Result for #15:

Result #15:

BOOLEAN [shift]

BOOLEAN [shift]

[reduce 603]

ID [shift]

[reduce 501] [reduce 902]

[reduce 603]

ID [shift]

LEFTPAREN [shift]

[reduce 801]

RIGHTPAREN [shift]

SEMICOLON [shift]

[reduce 1601]

INTERFACE [shift]

ID [shift]

LEFTBRACE [shift]

[reduce 1502]

RIGHTBRACE [shift]

[reduce 1401]

[reduce 304] [reduce 202]

[accept]

Strengths/Constraints

Strengths of Project:

* Working Lexical Analyzer
* Working Parser with fixed SR/RR conflicts
* Effective Communication between ourselves
* Able to output shifts and reduces

Constraints of Project:

* Poor time management on project
* Terminal ID giving us issues in the grammar
* Outputting in a unique way not like the examples

Team-Member Effort

Andrew and Zachary both effectively worked on this project with an equal share of the effort and workload.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Andrew Sison Zachary Martin