David Carlin

Clifford Black

Chris Houze

Program 5

Continue to work with the same team.

1. Derive an LR parsing table for Grammar 3.1 in your textbook. Show how your table parses the input:

x := (print(y), 2 + 3)

Use a $ to mark the end of the input.

LR Parsing Table for Grammar 3.1:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | id | num | print | ; | , | + | := | ( | ) | $ | S | E | L |
| 1 | s4 |  | s7 |  |  |  |  |  |  |  | g2 |  |  |
| 2 |  |  |  | s3 |  |  |  |  |  | a |  |  |  |
| 3 | s4 |  | s7 |  |  |  |  |  |  |  | g5 |  |  |
| 4 |  |  |  |  |  |  | s6 |  |  |  |  |  |  |
| 5 |  |  |  | r1 | r1 |  |  |  |  | r1 |  |  |  |
| 6 | s20 | s10 |  |  |  |  |  | s8 |  |  |  | g11 |  |
| 7 |  |  |  |  |  |  |  | s9 |  |  |  |  |  |
| 8 | s4 |  | s7 |  |  |  |  |  |  |  | g12 |  |  |
| 9 | s20 | s10 |  |  |  |  |  | s8 |  |  |  | g15 | g14 |
| 10 |  |  |  | r5 | r5 | r5 |  |  | r5 | r5 |  |  |  |
| 11 |  |  |  | r2 | r2 | s16 |  |  |  | r2 |  |  |  |
| 12 |  |  |  | s3 | s18 |  |  |  |  |  |  |  |  |
| 13 |  |  |  | r3 | r3 |  |  |  |  | r3 |  |  |  |
| 14 |  |  |  |  | s19 |  |  |  | s13 |  |  |  |  |
| 15 |  |  |  |  | r8 |  |  |  | r8 |  |  |  |  |
| 16 | s20 | s10 |  |  |  |  |  | s8 |  |  |  | g17 |  |
| 17 |  |  |  | r6 | r6 | s16 |  |  | r6 | r6 |  |  |  |
| 18 | s20 | s10 |  |  |  |  |  | s8 |  |  |  | g21 |  |
| 19 | s20 | s10 |  |  |  |  |  | s8 |  |  |  | g23 |  |
| 20 |  |  |  | r4 | r4 | r4 |  |  | r4 | r4 |  |  |  |
| 21 |  |  |  |  |  | s16 |  |  | s22 |  |  |  |  |
| 22 |  |  |  | r7 | r7 | r7 |  |  | r7 | r7 |  |  |  |
| 23 |  |  |  |  | r9 | s16 |  |  | r9 |  |  |  |  |

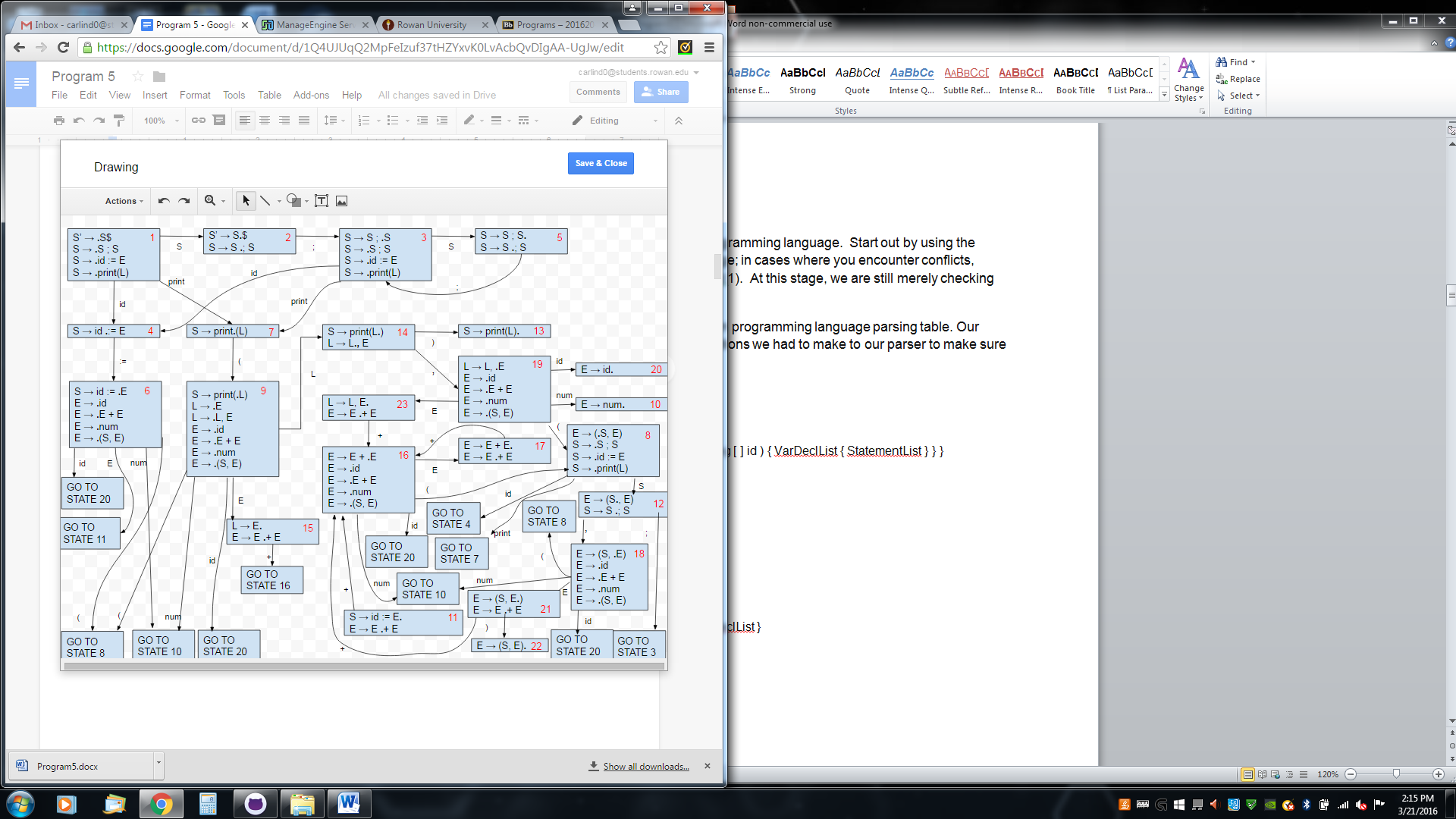
How the table parses the input:

|  |  |  |
| --- | --- | --- |
| Stack | Input | Action |
| 1 | x := (print(y), 2 + 3)$ | shift |
| 1 id4 | := (print(y), 2 + 3)$ | shift |
| 1id4:=6 | (print(y), 2 + 3)$ | shift |
| 1id4:=6(8 | print(y), 2 + 3)$ | shift |
| 1id4:=6(8print7 | (y), 2 + 3)$ | shift |
| 1id4:=6(8print7(9 | y), 2 + 3)$ | shift |
| 1id4:=6(8print7(9id20 | ), 2 + 3)$ | Reduce E → id |
| 1id4:=6(8print7(9E15 | ), 2 + 3)$ | Reduce L → E |
| 1id4:=6(8print7(9L14 | ), 2 + 3)$ | shift |
| 1id4:=6(8print7(9L14)13 | , 2 + 3)$ | Reduce S → print(L) |
| 1id4:=6(8S12 | , 2 + 3)$ | shift |
| 1id4:=6(8S12,18 | 2 + 3)$ | shift |
| 1id4:=6(8S12,18num10 | + 3)$ | Reduce E → num |
| 1id4:=6(8S12,18E21 | + 3)$ | shift |
| 1id4:=6(8S12,18E21+16 | 3)$ | shift |
| 1id4:=6(8S12,18E21+16num10 | )$ | Reduce E → num |
| 1id4:=6(8S12,18E21+16E17 | )$ | Reduce E → E + E |
| 1id4:=6(8S12,18E21 | )$ | shift |
| 1id4:=6(8S12,18E21)22 | $ | Reduce E → (S,E) |
| 1id4:=6E11 | $ | Reduce S → id := E |
| 1S2 | $ | Accept |

Here is the work we did to get the parsing table:

0. S’ → S$

1. S → S ; S
2. S → id := E
3. S → print ( L )
4. E → id
5. E → num
6. E → E + E
7. E → (S, E)
8. L → E
9. L → L, E



We shift / reduce conflict so we use the follow symbol to determine where the reduce action takes place.

Fol(S) = $ or ; or ,

Fol(E) = + or ) or $ or ; or ,

Fol(L) = ) or ,

2. Derive an LR parsing table for your own programming language. Start out by using the original grammar for your programming language; in cases where you encounter conflicts, modify the grammar until it is no worse than LR(1). At this stage, we are still merely checking for proper syntax; no output is produced.

Please see the attached excel sheet for our own programming language parsing table. Our original grammar is pasted below with modifications we had to make to our parser to make sure it was no worse than LR(1).

S' -> Program $

Program -> MainClass ClassDeclList

MainClass -> class id { public static void main ( String [ ] id ) { VarDeclList { StatementList } } }

ClassDeclList -> ClassDeclList ClassDecl

ClassDeclList ->

ClassDecl -> ClassDeclSpec ClassDeclDeff

ClassDeclSpec -> class id

ClassDeclDeff -> { VarDeclList MethodDeclList }

ClassDeclDeff -> extends id { VarDeclList MethodDeclList }

VarDeclList -> VarDeclList VarDecl

VarDeclList ->

VarDecl -> Type VarDeclID MultiDeclList ;

VarDeclID -> id VarDeclAssign

VarDeclAssign -> = Exp

VarDeclAssign ->

MultiDeclList -> MultiDeclList MultiDecl

MultiDeclList ->

MultiDecl -> , id VarDeclAssign

MethodDecl -> public Type id ( FormalList ) { VarDeclList StatementList return Exp ; }

MethodDeclList -> MethodDeclList MethodDecl

MethodDeclList ->

FormalList -> Type id FormalRest

FormalList ->

FormalRest -> , Type id FormalRest

FormalRest ->

Type -> int IntType

Type -> boolean

Type -> id

IntType -> [ ]

IntType ->

Statement -> { StatementList }

Statement -> if ( Exp ) { Statement } ElseIfList

Statement -> do { Statement } while ( Exp ) ;

Statement -> while ( Exp ) Statement

Statement -> for ( InitializationStm ; Exp ; IncrementStm ) Statement

Statement -> switch ( id ) { CaseList }

Statement -> System.out.println ( Exp ) ;

Statement -> id Assign

Statement -> ( Type id = Exp ) FormalVarExp ;

StatementList -> StatementList Statement

StatementList ->

Assign -> = Exp ;

Assign -> [ Exp ] = Exp ;

InitializationStm -> id = Exp

InitializationStm -> id [ Exp ] = Exp

IncrementStm -> id = Exp

IncrementStm -> id [ Exp ] = Exp

ElseIf -> else if ( Exp ) Statement

ElseIfList -> ElseIfList ElseIf

ElseIfList ->

FormalVarExp -> , ( Type id = Exp ) FormalVarExp

FormalVarExp ->

CaseList -> case Exp : Statement CaseList

CaseList -> default : Statement

Exp -> And Elist

Elist -> && And Elist

Elist ->

And -> Less Alist

Alist -> < Less Alist

Alist ->

Less -> Term Llist

Llist -> + Term Llist

Llist -> - Term Llist

Llist ->

Term -> Not Tlist

Tlist -> \* Not Tlist

Tlist ->

Not -> ! Factor DotArrayList

Not -> Factor DotArrayList

DotArrayList -> DotArrayList DotArray

DotArrayList ->

DotArray -> . Member

DotArray -> [ Exp ]

Member -> length

Member -> id ( ExpList )

ExpList -> Exp ExpRest

ExpList ->

ExpRest -> , Exp ExpRest

ExpRest ->

Factor -> number

Factor -> true

Factor -> false

Factor -> id

Factor -> this

Factor -> new New

Factor -> ( Exp )

New -> int [ Exp ]

New -> id ( )

3. Test both parsing tables from problems 1 and 2 with an LR parser which uses a stack. Diagnostic print statements may be used to demonstrate that it is parsing correctly.

**Test parsing table from problem 1 with an LR parser:**

**Input:** id := ( print ( id ) , num + num ) $

**Output:** Current state is: 1 action on token: id value id is s20

[id]

Current state is: 20 action on token: := value := is s21

[id, :=]

Current state is: 21 action on token: ( value ( is s14

[id, :=, (]

Current state is: 14 action on token: print value print is s5

[id, :=, (, print]

Current state is: 5 action on token: ( value ( is s6

[id, :=, (, print, (]

Current state is: 6 action on token: id value id is s19

[id, :=, (, print, (, id]

Current state is: 19 action on token: ) value ) is r4

[id, :=, (, print, (, E]

Current state is: 23 action on token: ) value ) is r8

[id, :=, (, print, (, L]

Current state is: 7 action on token: ) value ) is s8

[id, :=, (, print, (, L, )]

Current state is: 8 action on token: , value , is r3

[id, :=, (, S]

Current state is: 15 action on token: , value , is s16

[id, :=, (, S, ,]

Current state is: 16 action on token: num value num is s13

[id, :=, (, S, ,, num]

Current state is: 13 action on token: + value + is r5

[id, :=, (, S, ,, E]

Current state is: 17 action on token: + value + is s11

[id, :=, (, S, ,, E, +]

Current state is: 11 action on token: num value num is s13

[id, :=, (, S, ,, E, +, num]

Current state is: 13 action on token: ) value ) is r5

[id, :=, (, S, ,, E, +, E]

Current state is: 12 action on token: ) value ) is r6

[id, :=, (, S, ,, E]

Current state is: 17 action on token: ) value ) is s18

[id, :=, (, S, ,, E, )]

Current state is: 18 action on token: $ value $ is r7

[id, :=, E]

Current state is: 22 action on token: $ value $ is r2

[S]

Current state is: 2 action on token: $ value $ is a

No syntax errors detected

**Test parsing table from problem 2 with an LR parser:**

class Test

{

public static void main ( String [] args )

{

}

}

class AddNum

{

public int addNumber( int a, int b)

{

int total, firstInt, secondInt;

{

total = 0;

firstInt = a;

secondInt = b;

if(i < 10)

{

do

{

while(i < 10)

{

for(i = 1; i < 10; i = i + 10)

{

System.out.println(i < 10);

}

}

} while(i < 10);

}

else if(i < 10)

System.out.println(i < 10);

}

return total;

}

}

**For Output, please see LongOutput.txt file in folder.**

**Input:** class id { public static void main ( String [ ] id ) { { System.out.println ( number ) ; } } } $

**Output:** Current state is: 1 action on token: class value class is s226

[class]

Current state is: 226 action on token: id value id is s227

[class, id]

Current state is: 227 action on token: { value { is s228

[class, id, {]

Current state is: 228 action on token: public value public is s229

[class, id, {, public]

Current state is: 229 action on token: static value static is s230

[class, id, {, public, static]

Current state is: 230 action on token: void value void is s231

[class, id, {, public, static, void]

Current state is: 231 action on token: main value main is s232

[class, id, {, public, static, void, main]

Current state is: 232 action on token: ( value ( is s233

[class, id, {, public, static, void, main, (]

Current state is: 233 action on token: String value String is s234

[class, id, {, public, static, void, main, (, String]

Current state is: 234 action on token: [ value [ is s235

[class, id, {, public, static, void, main, (, String, []

Current state is: 235 action on token: ] value ] is s236

[class, id, {, public, static, void, main, (, String, [, ]]

Current state is: 236 action on token: id value id is s237

[class, id, {, public, static, void, main, (, String, [, ], id]

Current state is: 237 action on token: ) value ) is s238

[class, id, {, public, static, void, main, (, String, [, ], id, )]

Current state is: 238 action on token: { value { is s239

[class, id, {, public, static, void, main, (, String, [, ], id, ), {]

Current state is: 239 action on token: { value { is r10

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList]

Current state is: 240 action on token: { value { is s241

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {]

Current state is: 241 action on token: System.out.println value System.out.println is r40

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList]

Current state is: 242 action on token: System.out.println value System.out.println is s142

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList, System.out.println]

Current state is: 142 action on token: ( value ( is s143

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList, System.out.println, (]

Current state is: 143 action on token: number value number is s81

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList, System.out.println, (, number]

Current state is: 81 action on token: ) value ) is r79

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList, System.out.println, (, Factor]

Current state is: 66 action on token: ) value ) is r70

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList, System.out.println, (, Factor, DotArrayList]

Current state is: 67 action on token: ) value ) is r68

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList, System.out.println, (, Not]

Current state is: 46 action on token: ) value ) is r66

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList, System.out.println, (, Not, Tlist]

Current state is: 47 action on token: ) value ) is r64

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList, System.out.println, (, Term]

Current state is: 38 action on token: ) value ) is r63

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList, System.out.println, (, Term, Llist]

Current state is: 39 action on token: ) value ) is r60

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList, System.out.println, (, Less]

Current state is: 33 action on token: ) value ) is r59

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList, System.out.println, (, Less, Alist]

Current state is: 34 action on token: ) value ) is r57

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList, System.out.println, (, And]

Current state is: 28 action on token: ) value ) is r56

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList, System.out.println, (, And, Elist]

Current state is: 29 action on token: ) value ) is r54

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList, System.out.println, (, Exp]

Current state is: 144 action on token: ) value ) is s145

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList, System.out.println, (, Exp, )]

Current state is: 145 action on token: ; value ; is s146

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList, System.out.println, (, Exp, ), ;]

Current state is: 146 action on token: } value } is r36

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList, Statement]

Current state is: 89 action on token: } value } is r39

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList]

Current state is: 242 action on token: } value } is s243

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList, }]

Current state is: 243 action on token: } value } is s244

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList, }, }]

Current state is: 244 action on token: } value } is s245

[class, id, {, public, static, void, main, (, String, [, ], id, ), {, VarDeclList, {, StatementList, }, }, }]

Current state is: 245 action on token: $ value $ is r2

[MainClass]

Current state is: 3 action on token: $ value $ is r4

[MainClass, ClassDeclList]

Current state is: 4 action on token: $ value $ is r1

[Program]

Current state is: 2 action on token: $ value $ is a

No syntax errors detected

**Faulty input, String [ ] is missing its “[“**

**Input:** class id { public static void main ( String ] id ) { { System.out.println ( number ) ; } } } $

**Output:** class id { public static void main ( String ] id ) { { System.out.println ( number ) ; } } } $

Current state is: 1 action on token: class value class is s226

[class]

Current state is: 226 action on token: id value id is s227

[class, id]

Current state is: 227 action on token: { value { is s228

[class, id, {]

Current state is: 228 action on token: public value public is s229

[class, id, {, public]

Current state is: 229 action on token: static value static is s230

[class, id, {, public, static]

Current state is: 230 action on token: void value void is s231

[class, id, {, public, static, void]

Current state is: 231 action on token: main value main is s232

[class, id, {, public, static, void, main]

Current state is: 232 action on token: ( value ( is s233

[class, id, {, public, static, void, main, (]

Current state is: 233 action on token: String value String is s234

[class, id, {, public, static, void, main, (, String]

Current state is: 234 action on token: ] value ] is

Exception in thread "main" ParsingException: Table does not contain information for the input: ]

at state: 234

at Parser.parseInput(Parser.java:121)

at ParserDriver.main(ParserDriver.java:16)