Design Decisions and Evaluation of Parser

Maryam Dabaghchian-u1078006

February 1, 2018

1 Grammar

$Exp \rightarrow LtLevel$
$Exp \rightarrow Exp\ "\&\&"\ LtLevel$
$LtLevel \rightarrow PlusLevel$
$LtLevel \rightarrow LtLevel\ "<"\ PlusLevel$
$PlusLevel \rightarrow MultLevel$
$PlusLevel \rightarrow PlusLevel\ ("+" | "-")\ MultLevel$
$MultLevel \rightarrow DotLevel$
$MultLevel \rightarrow MultLevel\ "*"\ DotLevel$
$DotLevel \rightarrow PrimaryExp\ (\ "["\ Exp\ "]"\ |\ "."\ "length"\ |\ "."\ Id\ "("\ (\ Exp\ (\ ","\ Exp\ )^*\ )?\ ")"\ )^*$
$DotLevel \rightarrow "!"\ DotLevel$
$PrimaryExp \rightarrow "true"\ |\ "false"\ |\ "this"\ |\ "new"\ "int"\ "["\ Exp\ "]"\ |\ "new"\ Id\ "("\ ")"\ |\ <$
$NUM >\ |\ "("\ Exp\ ")"$

1.1 Eliminate Left Recursion

$Goal \rightarrow MainClass\ (\ ClassDecl\ )^*\ < EOF >$
$MainClass \rightarrow "class"\ Id\ "\{"\ "public"\ "static"\ "void"\ "main"\ "("\ "String"\ "["\ "]"\ Id\ ")"\ "\{"\ Stmt\ "\}"\ "\}"$
$ClassDecl \rightarrow "class"\ Id\ (\ "extends"\ Id\ )?\ "\{"\ (\ VarDecl\ )^*\ (\ MethodDecl\ )^*\ "\}"$
$VarDecl \rightarrow Type\ Id\ ";"$
$MethodDecl \rightarrow "public"\ Type\ Id\ "("\ (\ Type\ Id\ (\ ","\ Type\ Id\ )^*\ )?\ ")"\ "\{"\ (\ VarDecl\ )^*\ (\ Stmt\ )^*\ "return"\ Exp\ ";"\ "\}"$
$--$
$Type \rightarrow "int"\ "["\ "]"\ |\ "boolean"\ |\ "int"\ |\ Id$
$--$
$Stmt \rightarrow "\{"\ (\ Stmt\ )^*\ "\}"\ |$
$"if"\ "("\ Exp\ ")"\ Stmt\ "else"\ Stmt\ |$
$"while"\ "("\ Exp\ ")"\ Stmt\ |$
$"System.out.println"\ "("\ Exp\ ")"\ ";"\ |$
$Id\ "="\ Exp\ ";"\ |$
$Id\ "["\ Exp\ "]"\ "="\ Exp\ ";"\ |$
$--$
$Exp \rightarrow LtLevel\ Exp'$
$Exp' \rightarrow "\&\&"\ LtLevel\ Exp'$
$Exp' \rightarrow$

$LtLevel \rightarrow PlusLevel\ LtLevel'$
$LtLevel' \rightarrow " < "\ PlusLevel\ LtLevel'$
$LtLevel' \rightarrow$
$PlusLevel \rightarrow MultLevel\ PlusLevel'$
$PlusLevel' \rightarrow ("+"|"-")\ MultLevel\ PlusLevel'$
$PlusLevel' \rightarrow$
$MultLevel \rightarrow DotLevel\ MultLevel'$
$MultLevel' \rightarrow " * "\ DotLevel\ MultLevel'$
$MultLevel' \rightarrow$
$DotLevel \rightarrow PrimaryExp\ (\ "["\ Exp\ "]"\ |\ "."\ "length"\ |\ "."\ Id\ "("\ (\ Exp\ (\ ","\ Exp\ )^*\ )?\ ")"\ )^*$
$DotLevel \rightarrow "!"\ DotLevel$
$PrimaryExp \rightarrow "true"\ |\ "false"\ |\ "this"\ |\ "new"\ "int"\ "["\ Exp\ "]"\ |\ "new"\ Id\ "("\ ")"\ |\ <$
$NUM >\ |\ "("\ Exp\ ")"$

1.2 Left Factoring

$Goal \rightarrow "class"\ Id\ MainClass\ (\ "class"\ Id\ RegClass\ )^*\ < EOF >$
$MainClass \rightarrow "\{"\ "public"\ "static"\ "void"\ "main"\ "("\ "String"\ "["\ "]"\ Id\ ")"\ "\{"\ Stmt\ "\}"\ "\}"$
$RegClass \rightarrow (\ "extends"\ Id\ )?\ "\{"\ (\ VarDecl\ )^*\ (\ MethodDecl\ )^*\ "\}"$
$VarDecl \rightarrow Type\ Id\ ";"$
$MethodDecl \rightarrow "public"\ Type\ Id\ "("\ (\ Type\ Id\ (\ ","\ Type\ Id\ )^*\ )?\ ")"\ "\{"\ (\ VarDecl\ )^*\ (\ Stmt\ )^*\ "return"\ Exp\ ";"\ "\}"$
$- -$
$Type \rightarrow "int"\ "["\ "]"\ |\ "boolean"\ |\ "int"\ |\ Id$
$- -$
$Stmt \rightarrow "\{"\ (\ Stmt\ )^*\ "\}"\ |$
$"if"\ "("\ Exp\ ")"\ Stmt\ "else"\ Stmt\ |$
$"while"\ "("\ Exp\ ")"\ Stmt\ |$
$"System.out.println"\ "("\ Exp\ ")"\ ";"\ |$
$Id\ " = "\ Exp\ ";"\ |$
$Id\ "["\ Exp\ "]"\ " = "\ Exp\ ";"\ |$
$- -$
$Exp \rightarrow LtLevel\ Exp'$
$Exp' \rightarrow "\&\&"\ LtLevel\ Exp'$
$Exp' \rightarrow$
$LtLevel \rightarrow PlusLevel\ LtLevel'$
$LtLevel' \rightarrow " < "\ PlusLevel\ LtLevel'$
$LtLevel' \rightarrow$
$PlusLevel \rightarrow MultLevel\ PlusLevel'$
$PlusLevel' \rightarrow ("+"|"-")\ MultLevel\ PlusLevel'$
$PlusLevel' \rightarrow$
$MultLevel \rightarrow DotLevel\ MultLevel'$
$MultLevel' \rightarrow " * "\ DotLevel\ MultLevel'$
$MultLevel' \rightarrow$
$DotLevel \rightarrow PrimaryExp\ (\ "["\ Exp\ "]"\ |\ "."\ "length"\ |\ "."\ Id\ "("\ (\ Exp\ (\ ","\ Exp\ )^*\ )?\ ")"\ )^*$
$DotLevel \rightarrow "!"\ DotLevel$
$PrimaryExp \rightarrow "true"\ |\ "false"\ |\ "this"\ |\ "new"\ NewExp\ |\ < NUM >\ |\ "("\ Exp\ ")"$

$NewExp \rightarrow "int" \; "[" \; Exp \; "]" \mid Id \; "(" \; ")"$

2 Design Decisions