
PROGRAMMING LANGUAGE DESIGN

COMPILERS COURSE
CSE419

Nikhil Pinnaparaju
20161118

Ramkishore S
20161092

1 Micro syntax

id := [A-Za-z][[A-Za-z_][0-9]]*

signed_integer := -?[0-9]⁺

unsigned_integer := [0-9]⁺

bool := True | False

char := ' '

keywords := while | for | if | else | input | output | int | uint | bool | char | True | False | return

2 Macro syntax

Program := Decl⁺

Decl := variableDecl | functionDecl

variableDecl := varTypeDecl ;

varTypeDecl := varTypeDecl, varname

varTypeDecl := type varname

varname := varname [unsigned_integer] | id

type := 'int' | 'uint' | 'char' | 'bool'

functionDecl := type id (args) statementBlock | void id (args) statementBlock

args := args, varTypeDecl | varTypeDecl | ϵ

statementBlock ::= { variableDecl* statement* }

statement := <expr>; | ifBlock | whileBlock | forBlock | conditionOp | return expr | assignExpr;

expr := expr binOp expr | unaryOp expr | id | funcCall

expr := conditionExpr

conditionExpr := (expr)? expr : expr

assignExpr := varname equalOp expr

ifBlock := if(expr) statementBlock {else statementBlock}?

forLoop := for(<expr>; expr; <expr>) statementBlock

whileLoop := while<expr> statementBlock

binOp := > | < | % | '/' | * | + | - | >= | <= | & | '|' | == | !=

unaryOp := - | !

equalOp := =

funcCall := id '(' (varname,)* ')'