

Compiler Construction Assignment 4b - Report

Ambiguity Removal:

PROG **PROG** DEC | DEC
DEC KEY IDFR (VARDEC) COL BLOCK
COL :
KEY def
VARDEC ϵ | VARDECNE
VARDECNE IDFR | **VARDECNE**, IDFR
BLOCK { ENE }
ENE EXP | ϵ
EXP **IDFR** | INTLIT | **IDFR** = EXP | (EXP BINOP EXP) | **IDFR** (ARGS) | if EXP then
COL BLOCK else COL BLOCK | while EXP COL BLOCK | for IDFR OPLIT (ARGS) COL
BLOCK | IDFR = input (QUOTES STRING QUOTES) | print (QUOTES STRING
QUOTES) | KEYWORD (INTLIT)
KEYWORD int | float | str
QUOTES "
STRING INTLIT | **STRING** IDFR
ARGS ARGSNE | ϵ
ARGSNE EXP | **ARGSNE**, EXP
OPLIT in | not in
IDFR id
INTLIT num
BINOP == | < | > | <= | >= | + | * | - | % | / | and | or | !=

Removing left recursion in

PROG **PROG** DEC | DEC
VARDECNE IDFR | **VARDECNE**, IDFR
STRING INTLIT | **STRING** IDFR
ARGSNE EXP | **ARGSNE**, EXP

Removing left factoring in

EXP IDFR | IDFR = EXP | IDFR (ARGS)

PROG DEC PROG'

PROG' DEC PROG' | ϵ

DEC KEY IDFR (VARDEC) COL BLOCK

COL :

KEY def

VARDEC ϵ | VARDECNE

VARDECNE IDFR VARDECNE'

VARDECNE' , IDFR VARDECNE' | ϵ

BLOCK { ENE }

ENE EXP | ϵ

EXP' ϵ | = EXP | (ARGS)

EXP IDFR EXP' | INTLIT | (EXP BINOP EXP) | if EXP then
COL BLOCK else COL BLOCK | while EXP COL BLOCK | for IDFR OPLIT (ARGS) COL
BLOCK | IDFR = input (QUOTES STRING QUOTES) | print (QUOTES STRING
QUOTES) | KEYWORD (INTLIT)

KEYWORD int | float | str

QUOTES "

STRING INLIT STRING'

STRING' IDFR STRING' | ϵ

ARGS ARGSNE | ϵ

ARGSNE EXP ARGSNE'

ARGSNE' , EXP ARGSNE' | ϵ

OPLIT in | not in

IDFR id

INTLIT num

BINOP == | < | > | <= | >= | + | * | - | % | / | and | or | !=

Removing left factoring in the part highlighted red above:

PROG	DEC PROG'
PROG'	DEC PROG' ϵ
DEC	KEY IDFR (VARDEC) COL BLOCK
COL	:
KEY	def
VARDEC	ϵ VARDECNE
VARDECNE	IDFR VARDECNE'
VARDECNE'	, IDFR VARDECNE' ϵ
BLOCK	{ ENE }
ENE	EXP ϵ
EXP	(EXP BINOP EXP) INTLIT if EXP then COL BLOCK else COL BLOCK while EXP COL BLOCK for IDFR OPLIT (ARGS) COL BLOCK print (QUOTES STRING QUOTES) KEYWORD (INTLIT) IDFR EXP'
EXP'	ϵ (ARGS) =EXP''
EXP''	EXP input (QUOTES STRING QUOTES)
KEYWORD	int float str
QUOTES	"
STRING	INLIT STRING'
STRING'	IDFR STRING' ϵ
ARGS	ARGSNE ϵ
ARGSNE	EXP ARGSNE'
ARGSNE'	, EXP ARGSNE' ϵ
OPLIT	in not in
IDFR	id
INTLIT	num
BINOP	== < > <= >= + * - % / and or !=

*Note: In text file for coding, , has been replaced by Comma and " has been replaced by q.
The code to be parsed is first passed through flex in ubuntu. After getting the tokens list from flex output,
it is given as input to the parser.*

