Programming Assignment 3

Jaakko Koskela 526050, Leo Kivikunnas 525925

1

1.1 Pros

- We wrote a script to automatically test that our parser passes the given test cases
- We wrote our own parse table generator since the suggested tool was broken
- We refactored our scanner to improve its readability

1.2 Cons

- Our parser has has almost 300 lines of hard coded lists and tables in the start. It would be better to have them in separate files, but since we are supposed to turn in only the README, scanner.py and parser.py they need to be included that way.

2 First

Col1	Col2
"Program"	["void", "EPSILON", "int"]
"Declaration-list"	["void", "EPSILON", "int"]
"Declaration"	["void", "int"]
"Declaration-initial"	["void", "int"]
"Declaration-prime"	["(", "[", ":"]
"Var-declaration-prime"	["(", "[", ","] ["[", ","]
"Fun-declaration-prime"	["("]
"Type-specifier"	["void", "int"]
"Params"	["void", "int"]
"Param-list-void-abtar"	["EPSILON", "ID"]
"Param-list"	["EPSILON", ","]
"Param"	["void", "int"]
"Param-prime"	["EPSILON", "["]
"Compound-stmt"	["{"]
"Statement-list"	["NUM", "switch", ";", "{", "(", "ID", "EPSILON", "return", "if", "while", "continue", "break"]
"Statement"	["return", "NUM", "switch", ";", "{", "(", "ID", "continue", "if", "while", "break"]
"Expression-stmt"	["NUM", ";", "(", "ID", "continue", "break"]
"Selection-stmt"	["if"]
"Iteration-stmt"	["while"]
"Return-stmt"	["return"]
"Return-stmt-prime"	["(", "NUM", "ID", ";"]
"Switch-stmt"	["switch"]
"Case-stmts"	["case", "EPSILON"]
"Case-stmt"	["case"]
"Default-stmt"	["EPSILON", "default"]
"Expression"	["(", "NUM", "ID"]
"B"	["*", "-", "==", "<", "+", "=", "(", "EPSILON", "["]
"H"	["(", "NUM", "ID"] ["*", "-", "==", "<", "+", "=", "(", "EPSILON", "["] ["*", "-", "==", "<", "+", "=", "EPSILON"] ["(", "NUM"]
"Simple-expression-zegond"	["(", "NUM"]
"Simple-expression-prime"	["*", "-", "==", "<", "+", "(", "EPSILON"]
"C"	["*", "-", "==", "<", "+", "(", "EPSILON"] ["==", "<", "EPSILON"] ["==", "<"]
"Relop"	["==", "<"]
"Additive-expression"	["(", "NUM", "ID"]
"Additive-expression-prime"	["*", "-", "+", "(", "EPSILON"]
"Additive-expression-zegond"	["(", "NUM"]
"D"	["-", "EPSILON", "+"]
"Addop"	["-", "+"]
"Term"	["(", "NUM", "ID"]
"Term-prime"	["(", "EPSILON", "*"]
"Term-zegond"	["(", "NUM"]
"G"	["EPSILON", "*"]
"Factor"	["(", "NUM", "ID"]
"Var-call-prime"	["(", "EPSILON", "["]
"Var-prime"	["EPSILON", "["]
"Factor-prime"	["(", "EPSILON"]
"Factor-zegond"	["(", "NUM"]
"Args"	["(", "EPSILON", "NUM", "ID"]
"Arg-list"	["(", "NUM", "ID"]
"Arg-list-prime"	["EPSILON", ","]
11	

3 Follow

"Pechantion " "brook", "switch", "(", "default", "N. "NUM", "(", "lesses", "while", "S", "continue", "NUM")	Col1	Col2
"Declaration-list"	"Program"	["\$"]
Declaration Food*, "int*, switch*," (", break*, "default*, while*, NOM*, ", ID*, "f*, "return*,") * \$", "continue*, case"		["brook" "cwitch" "/" "default" "]" "." "ID" "[" "return" "eece" "while" "@" "continue" "NHM"]
"Declaration-prime"		["void" "int" "witch" "(" "break" "default" "while" "NIIM" "" "f" "f" "f" "return" "?" "continue" "case"]
"Tyne-pecifier"		["" ", "", "",", "",", "",", "",",",",",",",",",",",",",",",",",",","
"Tyne-pecifier"		[', (, [,], ,]
"Tyne-pecifier"	11	["void" "int" "switch" "(" "break" "default" "while" "NIIM" "" "ID" "(" "return" ")" "8" "continue" "case"]
"Type-specifier"		["void" "int" "switch" "(" "break" "default" "while" "NIIM" "" "ID" "(" "return" ")" "8" "continue" "case"]
"Params's		
Param-list* [7]		į j
"Param. st" "Param. prime" "Param. prime" "Compound-stmt" ["void", "int", "switch", "(", "break", "clss", "default", "while, "?", "ID", "[", "if", "return", "case", "}", "\$", "s", "continue", "NUM"] "Statement.ist" "Statement" ["break", "clse", "switch", "(", "default", "y", "ID", "[", "if", "return", "case", "while", "continue", "NUM"] "Selection-stmt" ["break", "clse", "switch", "(", "default", "y", "", "ID", "[", "if", "return", "case", "while", "continue", "NUM"] "Selection-stmt" ["break", "clse", "switch", "(", "default", "y", "", "ID", "[", "if", "return", "case", "while", "continue", "NUM"] "Return-stnt" ["break", "clse", "switch", "(", "default", "y", "", "D", "[", "if", "return", "case", "while", "continue", "NUM"] "Return-stnt" ["break", "clse", "switch", "(", "default", "y", "", "D", "[", "if", "return", "case", "while", "continue", "NUM"] "Return-stnt" ["break", "clse", "switch", "(", "default", "y", "", "D", "[", "if", "return", "case", "while", "continue", "NUM"] "Switch-stnt" ["break", "clse", "switch", "(", "default", "y", "", "D", "[", "if", "return", "case", "while", "continue", "NUM"] "Switch-stnt" ["break", "clse", "switch", "(", "default", "y", "", "D", "[", "if", "return", "case", "while", "continue", "NUM"] "Case-stnts" ["break", "clse", "switch", "(", "default", "y", "", "D", "[", "if", "return", "case", "while", "continue", "NUM"] "Case-stnts" ["break", "clse", "switch", "(", "default", "y", "", "D", "[", "if", "return", "case", "while", "continue", "NUM"] "Expression" ["break", "clse", "switch", "(", "default", "y", "", "D", "", "", "", "", "", "", "",	1	
"Param" "Param-prime" "Compound-stnt" "Void", 'init", "switch", "(", "break", "else", "default", "while", ", "ID", "(", "if", "return", 'case", ")", "\$", "continue", "NUM") "Statement-list" "Statement" "break", 'else", 'switch', "(", "default", "while", ", "ID", "(", "if", "return", "case", "NUM") "Selection-stnt" "break", 'else", 'switch', "(", "default", "yhile", "NUM", ", "1D", "(", "if", "return", "case", "while", continue", "NUM") "literation-stnt" "break", 'else", 'switch', "(", "default", "yhile", ", "D", "(", "if", "return", "case", "while", continue", "NUM") "literation-stnt" "break", 'else", 'switch', "(", "default", "yhile", ", "D", "(", "if", "return", "case", "while", continue", "NUM") "Return-stnt-prime" "break", 'else", 'switch', "(", "default", "yhile", ", "D", "(", "if", "return", "case", "while", continue", "NUM") "Switch-stnt" "break", 'else", 'switch', "(", "default", "yhile", ", "D", "(", "if", "return", "case", "while", continue", "NUM") "Switch-stnt" "break", 'else", 'switch', "(", "default", "yhile", ", "D", "(", "if", "return", "case", "while", continue", "NUM") "Switch-stnt" "break", 'else", 'switch', "(", "default", "yhile", ", "D", "(", "if", "return", "case", "while", continue", "NUM") "Switch-stnt" "break", 'else", 'switch', "(", "default", "yhile", ", "D", "(", "if", "return", "case", "while", continue", "NUM") "break", 'else", 'switch', "(", "default", "yhile", ", "D", "(", "if", "return", "case", "while", continue", "NUM") "break", 'else", 'switch', "(", "default", "yhile", ", "D", "(", "if", "return", "case", "while", continue", "NUM") "break", 'else", 'switch', "(", "default", "yhile", "phoreum, "phoreum, "while", continue", "NUM") "break", 'else", 'switch', "(", "default", "yhile", "phoreum, "phoreum, "while", continue", "NUM") "break", 'else", 'switch', "(", "default", "yhile", "phoreum, "phoreum, "while", continue", "NUM") "break", 'else", 'switch', "(", "default", "yhile", "phoreum, "phoreum, "phoreum, "phoreum, "phoreum, "phoreum, "phoreum, "ph		
"Param-prime" "Compound-stmt" ["void", "int", "switch", "(", "break", "else", "default", "while", "", "ID", "(", "if", "return", "ase", "), "", "", "", "", "", "", "", "", "		
"Statement-list"		
"Statement-list"		["void" "int" "switch" "(" "break" "else" "default" "while "." "ID" "{" "if" "return" "case" "}" "\$" "continue" "NIIM"]
"Statement"		["3" "default" "case"]
"Expression-stmt"	1	
"Selection-stmt"	1	"break", "else", "switch", "(", "default", ")", ":", "ID", "{", "if", "return", "case", "while", "continue", "NUM"]
"Return-stmt-prime" ("break", "clse", "switch", "(", "default", "p", "p", "p", "p", "p", "if", "return", "case", "while", "continue", "NUM"] "Switch-stmt" ("break", "clse", "switch", "(", "default", "while", "p", "p", "case", "p", "continue", "NUM"] "Case-stmts" ("p", "default") "Case-stmt" ("p", "default") "Expression" ("p", "default") "Expression" ("p", "default") "Expression" ("p", "default") "Expression" ("p", "n", "p", "p") "B" ("p", "n", "p", "p") "Simple-expression-zegond" ("p", "n", "p", "p") "Case-stmt" ("p", "n", "p", "p") "Case-stmt" ("p", "n", "p", "p") "Simple-expression-prime" ("p", "n", "p", "p") "Additive-expression-prime" ("p", "n", "p", "p") "Additive-expression-prime" ("p", "p", "p", "p", "p") "Additive-expression-prime" ("p", "p", "p", "p", "p", "p", "p") "Additive-expression-prime" ("p", "p", "p", "p", "p", "p", "p", "p		["break", "else", "switch", "(", "default", ")", ";", "ID", "{", "if", "return", "case", "while", "continue", "NUM"]
"Return-stmt-prime" ("break", "clse", "switch", "(", "default", "p", "p", "p", "p", "p", "if", "return", "case", "while", "continue", "NUM"] "Switch-stmt" ("break", "clse", "switch", "(", "default", "while", "p", "p", "case", "p", "continue", "NUM"] "Case-stmts" ("p", "default") "Case-stmt" ("p", "default") "Expression" ("p", "default") "Expression" ("p", "default") "Expression" ("p", "default") "Expression" ("p", "n", "p", "p") "B" ("p", "n", "p", "p") "Simple-expression-zegond" ("p", "n", "p", "p") "Case-stmt" ("p", "n", "p", "p") "Case-stmt" ("p", "n", "p", "p") "Simple-expression-prime" ("p", "n", "p", "p") "Additive-expression-prime" ("p", "n", "p", "p") "Additive-expression-prime" ("p", "p", "p", "p", "p") "Additive-expression-prime" ("p", "p", "p", "p", "p", "p", "p") "Additive-expression-prime" ("p", "p", "p", "p", "p", "p", "p", "p		"break" "else" "switch" "(" "default" ")" ", "D" "(" "if" "return" "case" "while" "continue" "NUM"]
"Return-stnt-prime" ["break", "else", "switch", "(", "default", "while", ";", "ID", "(", "if", "return", "case", ")", "continue", "NUM"] "Case-stmt" ["break", "else", "switch", "(", "default", ")", ", ";", "TD", "(", "if", "return", "case", "while", "continue", "NUM"] "Case-stmt" ["]", "default", "case"] "Expression" ["]", "default", "case"] "Expression" ["]", "default", "case"] "B" ["]"] "B" ["]"] "Simple-expression-zegond" ["]", ", ", ", ", ", "]"] "Simple-expression-prime" ["]", ", ", ", ", ", "]"] "Relop" ["]", ", ", ", ", ", ", "]"] "Additive-expression-prime" [", ", ", ", ", ", ", "]"] "Additive-expression-prime" [", ", ", ", ", ", ", "]"] "Term [", ", ", ", ", ", ", ", ", ", ", ", ", "		"break" "else" "switch" "(" "default" ")" "" "ID" "(" "if" "return" "case" "while" "continue" "NUM"]
"Switch-stmt" ("break", "else", "switch", "(", "default", ")", ", ", ", "ID", "(", "it, "return", "case", "while", "continue", "NUM") "Case-stmts" (")", "default", "case") "Default-stmt" (")", ", ", ", ", ", ", ", ", ", ", ", ", "	III	"break" "else" "switch" "(" "default" "while" "" "TD" "(" "return" "case" ")" "continue" "NUM"
"Case-stmts" ["]", "default"] "Case-stmt" ["]"] "Expression" ["]", ", ", ", ", ", "]"] "B" ["]", ", ", ", ", ", "]"] "B" ["]", ", ", ", ", ", "]"] "Simple-expression-zegond" [", ", ", ", ", ", ", "]"] "Simple-expression-prime" [", ", ", ", ", ", "]"] "Relop" ["D, "(", "NUM"] "Additive-expression" [", ", ", ", ", ", "]"] "Additive-expression agond" [", ", ", ", ", ", "]"] "Additive-expression agond" [", ", ", ", ", ", "]"] "Additive-expression agond" [", ", ", ", ", ", ", "] "Additive-expression agond" [", ", ", ", ", ", ", ", ", "] "Addop" [", ", ", ", ", ", ", ", ", ", ", ", ", "	11	"break" "else" "switch" "(" "default" ")" "" ""]" "(" "if" "return" "case" "while" "continue" "NUM"]
"Case-stmt" "Default-stmt" "Expression" "Expression" "B" "Expression" "B" "F", ", ", ", ", ", ", ", ", ", ", ", ", "		["3" "default"]
"Default-stmt" "Expression" "B" "H" "Simple-expression-zegond" "Simple-expression-prime" "Relop" "Additive-expression-prime" "In """ "" "" "" "" "" "" "" "" "" "" "" "	III	
"Expression" "B" "H" "Simple-expression-zegond" "Simple-expression-prime" "Relop" "Additive-expression-prime" "Addop" "D" "Addop" "ID", "(", "NUM") "Term "Addop" "ID", "(", "NUM") "Term-prime" "C", ", ", "], ", ==", "], ", ", ", ", ", ", ", ", ", ", ", ", ",	U.	[7, 4666]
"B" "H" "H" [",",",")","]" "Simple-expression-zegond" [",",",")","]" "Simple-expression-prime" [",",",")","]" "Relop" "Additive-expression" [",",",",",","]" "Additive-expression-prime" [",",",",",","]" "Additive-expression-zegond" [",",",",",",","]" "Addop" [",",",",",",",",",",",",",",",",",","		[יין " (י', יין " יי, יין
"Simple-expression-zegond" "Simple-expression-prime" "C" "Relop" "Relop" "IID", "(", ",",")", "] "Additive-expression" "Additive-expression" "Additive-expression" "Additive-expression or		[מ] מיל (מי מי מ
"Simple-expression-zegond" "Simple-expression-prime" "C" "Relop" "Relop" "IID", "(", ",",")", "] "Additive-expression" "Additive-expression" "Additive-expression" "Additive-expression or	"H"	[מן מי מי מי מי]
"Simple-expression-prime" "C" "Relop" "Additive-expression" "Additive-expression-prime" "Additive-expression-prime" "Additive-expression-zegond" "P" "Additive-expression-zegond" "P" "Additive-expression-zegond" "P" "Additive-expression-zegond" "P" "P" "Addop" "ID", "(", "NUM") "Term-prime" "Term-prime" "C", ", ", ", ", ", ", ", ", ", ", ", ", "	"Simple-expression-zegond"	[פור פור פור פור פור פור פור פור פור פור
"Relop" ["!", ", ", "), "]"] "Additive-expression" ["i, ", ", "]"] "Additive-expression-prime" [", ", ", ", ", "]"] "Additive-expression-zegond" [", ", ", ", ", ", "] "D" [", ", ", ", ", ", ", "] "Addop" ["ID", "(", "NUM"] "Term" ["ID", "(", "NUM"] "Term-prime" ["2", ", ", "], "==", "]", ", ", "+", "<"] "Term-zegond" [", ", ", "], "==", "]", ", ", "+", "<"] "Term-zegond" [", ", ", "], "==", "]", ", ", "+", "<"] "Factor" [", ", ", "], "==", "]", ", ", "+", "<"] "Var-call-prime" [", ", ", ", ", ", ", ", ", ", ", ", ", "		[פן יי
"Relop" ["ID", "(", "NUM"] "Additive-expression" [",",",",",","] "Additive-expression-prime" [",",")", "==","]", ",", "<"]		[מ]", ", ", ", "]
"Additive-expression" "Additive-expression-prime" "Additive-expression-zegond" "D" "Addop" "Term" "Term-prime" "Term-zegond" "", ", ", ", ", ", ", ", ", ", ", ", ",	"Relop"	["ID", "(", "NUM"]
"Additive-expression-zegond" [",",")", "==","]", ";", "<"]	"Additive-expression"	[יון", יין ", יי, יי, יין "
"Additive-expression-zegond" [",",")", "==","]", ";", "<"]		[",","], $[",","]$, $[",",",","]$
"D" [",",")", "==","]", ",", "<"] "Addop" ["D", "(", "NUM"] "Term" ["-",","]", "==",")", ",", "+", "<"] "Term-prime" ["-",","]", "==",")", ",", "+", "<"] "Term-zegond" ["-",","]", "==",")", ",", "+", "<"] "G" ["-",","]", "==",")", ",", "+", "<"] "Factor" ["-",",","",",",",",",",",",",",",",",",		[",",")", "==","]", ",", "<"]
"Addop" "Term" "Term" "Term-prime" "Term-prime" "Term-zegond" "G" "Factor" "Var-call-prime" "Yar-prime" "Factor-prime" "Factor		[",",")", "==","]", ";", "<"]
"Term" "Term-prime" "Term-zegond" "G" "Factor" "Var-call-prime" "Yar-prime" "Factor-prime" "Factor-pri	"Addop"	["ID", "(", "NUM"]
"Term-prime" ["-", ", "]", "==", ")", ";", "+", "<"] "Term-zegond" ["-", ", "]", "==", ")", ";", "+", "<"] "G" ["-", ", "]", "==", ")", ";", "+", "<"] "Factor" ["-", ", "*", ")", "==", "]", ", ", "+", "<"] "Var-call-prime" ["-", ", "*", ")", "==", "]", ", ", "+", "<"] "Yar-prime" ["-", ", "*", ")", "==", "]", ", ", "+", "<"] "Factor-prime" ["-", ", "*", ")", "==", "]", ", ", "+", "<"] "Args" ["]" "Arg-list"	"Term"	["-",",","]","==",")",",","+","<"]
"Term-zegond" ["-", ", "]", "==", ")", ";", "+", "<"] "G" ["-", ", "]", "==", ")", ";", "+", "<"] "Factor" ["-", ", "*", ")", "==", "]", ",", "+", "<"] "Var-call-prime" ["-", ", "*", ")", "==", "]", ",", "+", "<"] "Yar-prime" ["-", ", "*", ")", "==", "]", ",", "+", "<"] "Factor-prime" ["-", ", "*", ")", "==", "]", ",", "+", "<"] "Factor-zegond" ["-", ", "*", ")", "==", "]", ",", "+", "<"] "Args" ["]" "Arg-list"	"Term-prime"	["-", ",", "]", "==", ")", ";", "+", "<"]
"G" ["-", ", "]", "==", ")", ";", "+", "<"] "Factor" ["-", ", "*", ")", "==", "]", ",", "+", "<"] "Var-call-prime" ["-", ", "*", ")", "==", "]", ",", "+", "<"] "Var-prime" ["-", ", "*", ")", "==", "]", ",", "+", "<"] "Factor-prime" ["-", ", "*", ")", "==", "]", ",", "+", "<"] "Factor-zegond" ["-", ", "*", ")", "==", "]", ",", "+", "<"] "Args" ["]" "Arg-list"		["-", ",", "]", "==", ")", ";", "+", "<"]
"Factor" ["-", ", "*", ")", "==", "]", ",", "+", "<"]	"G"	["-", ",", "]", "==", ")", ";", "+", "<"]
"Var-call-prime" ["-", ", "*", ")", "==", "]", ",", "+", "<"]	"Factor"	["-", ", ", "*", ")", "==", "]", ", ", "+", "<"]
"Var-prime" ["-", ", "*", ")", "==", "]", ",", "+", "<"] "Factor-prime" ["-", ", "*", ")", "==", "]", ",", "+", "<"] "Factor-zegond" ["-", ", "*", ")", "==", "]", ",", "+", "<"] "Args" [")"] "Arg-list" [")"]	11	["-", ",", "*", ")", "==", "]", ",", "+", "<"]
"Factor-prime" ["-", ", "*", ")", "==", "]", ",", "+", "<"] "Factor-zegond" ["-", ", "*", ")", "==", "]", ",", "+", "<"] "Args" [")"] "Arg-list" [")"]		[".", ", "*", "\" "==", "[", ", ", "+", "<"]
"Args" [")"] "Arg-list" [")"]		["-", ", ", "*", ")", "==", "]", ";", "+", "<"]
"Args" [")"] "Arg-list" [")"]	"Factor-zegond"	["-", ", ", "*", ")", "==", "]", ";", "+", "<"]
"Arg-list-prime" [")"]	"Arg-list"	[")"]
	"Arg-list-prime"	[")"]

4 Parse Table

4.1 parsetable no.1

	ID	NUM	=	•	:
Duagnam	None	None	None	None	None
Program					
Declaration-list Declaration	EPSILON SYNCH	EPSILON SYNCH	EPSILON SYNCH	None None	None None
			SYNCH		SYNCH
Declaration-initial	None SYNCH	None SYNCH		None	
Declaration-prime	,0		Var-declaration-prime	None	None
Var-declaration-prime	SYNCH	SYNCH	;	None	None
Fun-declaration-prime	SYNCH	SYNCH	SYNCH	None	None
Type-specifier	SYNCH	None	None	None	None
Params	None	None	None	None	None
Param-list-void-abtar	ID Param-prime Param-list	None	None	None	None
Param-list	None	None	None	None	, Param Param-list
Param	None	None	None	None	SYNCH
Param-prime	None	None	None	None	EPSILON
Compound-stmt	SYNCH	SYNCH	SYNCH	None	None
Statement-list	Statement Statement-list	Statement Statement-list	Statement Statement-list	None	None
Statement	Expression-stmt	Expression-stmt	Expression-stmt	None	None
Expression-stmt	Expression;	Expression;	;	None	None
Selection-stmt	SYNCH	SYNCH	SYNCH	None	None
Iteration-stmt	SYNCH	SYNCH	SYNCH	None	None
Return-stmt	SYNCH	SYNCH	SYNCH	None	None
Return-stmt-prime	Expression;	Expression;	;	None	None
Switch-stmt	SYNCH	SYNCH	SYNCH	None	None
Case-stmts	None	None	None	None	None
Case-stmt	None	None	None	None	None
Default-stmt	None	None	None	None	None
Expression	ID B	Simple-expression-zegond	SYNCH	None	SYNCH
В	None	None	Simple-expression-prime	None	Simple-expression-prime
Н	None	None	GDC	None	GDC
Simple-expression-zegond	None	Additive-expression-zegond C	SYNCH	None	SYNCH
Simple-expression-prime	None	None	Additive-expression-prime C	None	Additive-expression-prime C
C	None	None	EPSILON	None	EPSILON
Relop	SYNCH	SYNCH	None	None	None
Additive-expression	Term D	Term D	SYNCH	None	SYNCH
Additive-expression-prime	None	None	Term-prime D	None	Term-prime D
Additive-expression-zegond	None	Term-zegond D	SYNCH	None	SYNCH
D	None	None	EPSILON	None	EPSILON
Addop	SYNCH	SYNCH	None	None	None
Term	Factor G	Factor G	SYNCH	None	SYNCH
Term-prime	None	None	Factor-prime G	None	Factor-prime G
Term-zegond	None	Factor-zegond G	SYNCH	None	SYNCH
G	None	None	EPSILON	None	EPSILON
Factor	ID Var-call-prime	NUM	SYNCH	None	SYNCH
Var-call-prime	None	None	Var-prime	None	Var-prime
Var-prime	None	None	EPSILON	None	EPSILON
Factor-prime	None	None	EPSILON	None	EPSILON
Factor-zegond	None	NUM	SYNCH	None	SYNCH
Args	Arg-list	Arg-list	None	None	None
Arg-list	Expression Arg-list-prime	Expression Arg-list-prime	None	None	None
Arg-list-prime	None	None	None	None	, Expression Arg-list-prime
mg not printe	Tione	Tione	Tione	110110	, Expression ring not-prime

4.2 parsetable no.2

	,	[]	()
Program	None	None	None	None	None
Declaration-list	None	None	EPSILON	None	EPSILON
Declaration	None	None	SYNCH	None	SYNCH
Declaration-initial	SYNCH	None	SYNCH	SYNCH	None
Declaration-prime	Var-declaration-prime	None	Fun-declaration-prime	None	SYNCH
Var-declaration-prime	[NUM] ;	None	SYNCH	None	SYNCH
Fun-declaration-prime	None	None	(Params) Compound-stmt	None	SYNCH
Type-specifier	None	None	None	None	None
Params	None	None	None	SYNCH	None
Param-list-void-abtar	None	None	None	EPSILON	None
Param-list	None	None	None	EPSILON	None
Param	None	None	None	SYNCH	None
Param-prime		None	None	EPSILON	None
Compound-stmt	None	None	SYNCH	None	{ Declaration-list Statement-list
Statement-list	None	None	Statement Statement-list	None	Statement Statement-list
Statement	None	None	Expression-stmt	None	Compound-stmt
Expression-stmt	None	None	Expression;	None	SYNCH
Selection-stmt	None	None	SYNCH	None	SYNCH
Iteration-stmt	None	None	SYNCH	None	SYNCH
Return-stmt	None	None	SYNCH	None	SYNCH
Return-stmt-prime	None	None	Expression;	None	SYNCH
Switch-stmt	None	None	SYNCH	None	SYNCH
Case-stmts	None	None	None	None	None
Case-stmt	None	None	None	None	None
Default-stmt	None	None	None	None	None
Expression	None	SYNCH	Simple-expression-zegond	SYNCH	None
В	[Expression] H	Simple-expression-prime	Simple-expression-prime	Simple-expression-prime	None
Н	None	GDC	None	GDC	None
Simple-expression-zegond	None	SYNCH	Additive-expression-zegond C	SYNCH	None
Simple-expression-prime	None	Additive-expression-prime C	Additive-expression-prime C	Additive-expression-prime C	None
C	None	EPSILON	None	EPSILON	None
Relop	None	None	SYNCH	None	None
Additive-expression	None	SYNCH	Term D	SYNCH	None
Additive-expression-prime	None	Term-prime D	Term-prime D	Term-prime D	None
Additive-expression-zegond	None	SYNCH	Term-zegond D	SYNCH	None
D	None	EPSILON	None	EPSILON	None
Addop	None	None	SYNCH	None	None
Term	None	SYNCH	Factor G	SYNCH	None
Term-prime	None	Factor-prime G	Factor-prime G	Factor-prime G	None
Term-zegond	None	SYNCH	Factor-zegond G	SYNCH	None
G	None	EPSILON	None	EPSILON	None
Factor	None	SYNCH	(Expression)	SYNCH	None
Var-call-prime	Var-prime	Var-prime	(Args)	Var-prime	None
Var-prime	[Expression]	EPSILON	None	EPSILON	None
Factor-prime	None	EPSILON	(Args)	EPSILON	None
Factor-zegond	None	SYNCH	(Expression)	SYNCH	None
Args	None	None	Arg-list	EPSILON	None
Arg-list	None	None	Expression Arg-list-prime	SYNCH	None
Arg-list-prime	None	None	None	EPSILON	None
8 not prime	110110	110110	110110	LI DILOIT	110110

4.3 parsetable no.3

	{	}	+	-	*
Program	None	None	None	None	None
Declaration-list	EPSILON	None	None	None	None
Declaration	SYNCH	None	None	None	None
Declaration-initial	None	None	None	None	None
Declaration-prime	SYNCH	None	None	None	None
Var-declaration-prime	SYNCH	None	None	None	None
Fun-declaration-prime	SYNCH	None	None	None	None
Type-specifier	None	None	None	None	None
Params	None	None	None	None	None
Param-list-void-abtar	None	None	None	None	None
Param-list	None	None	None	None	None
Param	None	None	None	None	None
Param-prime	None	None	None	None	None
Compound-stmt	SYNCH	None	None	None	None
Statement-list	EPSILON	None	None	None	None
Statement	SYNCH	None	None	None	None
Expression-stmt	SYNCH	None	None	None	None
Selection-stmt	SYNCH	None	None	None	None
Iteration-stmt	SYNCH	None	None	None	None
Return-stmt	SYNCH	None	None	None	None
Return-stmt-prime	SYNCH	None	None	None	None
Switch-stmt	SYNCH	None	None	None	None
Case-stmts	EPSILON	None	None	None	None
Case-stmt	SYNCH	None	None	None	None
Default-stmt	EPSILON	None	None	None	None
	None	None	None	None	None
Expression B	None	Simple-expression-prime	Simple-expression-prime	Simple-expression-prime	= Expression
В Н	None	G D C	G D C	G D C	= Expression = Expression
Simple-expression-zegond	None	None	None	None	None
<u> </u>	None	Additive-expression-prime C	Additive-expression-prime C	Additive-expression-prime C	None
Simple-expression-prime C	None	None	None	None	None
v					
Relop	None	None	None	None	None
Additive-expression	None	None	None	None	None
Additive-expression-prime	None	Term-prime D	Term-prime D	Term-prime D	None
Additive-expression-zegond	None	None	None	None	None
D	None	Addop Term D	Addop Term D	None	None
Addop	None	+	-	None	None
Term	None	SYNCH	SYNCH	None	None
Term-prime	None	Factor-prime G	Factor-prime G	Factor-prime G	None
Term-zegond	None	SYNCH	SYNCH	None	None
G	None	EPSILON	EPSILON	* Factor G	None
	None	SYNCH	SYNCH	SYNCH	None
Factor					
Var-call-prime	None	Var-prime	Var-prime	Var-prime	None
Var-call-prime Var-prime	None None	Var-prime EPSILON	EPSILON	EPSILON	None
Var-call-prime Var-prime Factor-prime	None None None	Var-prime EPSILON EPSILON	EPSILON EPSILON	EPSILON EPSILON	None None
Var-call-prime Var-prime Factor-prime Factor-zegond	None None None	Var-prime EPSILON EPSILON SYNCH	EPSILON EPSILON SYNCH	EPSILON EPSILON SYNCH	None None None
Var-call-prime Var-prime Factor-prime Factor-zegond Args	None None None None None	Var-prime EPSILON EPSILON SYNCH None	EPSILON EPSILON SYNCH None	EPSILON EPSILON SYNCH None	None None None
Var-call-prime Var-prime Factor-prime Factor-zegond	None None None	Var-prime EPSILON EPSILON SYNCH	EPSILON EPSILON SYNCH	EPSILON EPSILON SYNCH	None None None

4.4 parsetable no.4

	<	==	if	else	void
D					
Program	None	None	None	None	Declaration-list
Declaration-list	None	None	EPSILON	None	Declaration Declaration-list
Declaration	None	None	SYNCH	None	Declaration-initial Declaration-prim
Declaration-initial	None	None	None	None	Type-specifier ID
Declaration-prime	None	None	SYNCH	None	SYNCH
Var-declaration-prime	None	None	SYNCH	None	SYNCH
Fun-declaration-prime	None	None	SYNCH	None	SYNCH
Type-specifier	None	None	None	None	void
Params	None	None	None	None	void Param-list-void-abtar
Param-list-void-abtar	None	None	None	None	None
Param-list	None	None	None	None	None
Param	None	None	None	None	Declaration-initial Param-prime
Param-prime	None	None	None	None	None
Compound-stmt	None	None	SYNCH	SYNCH	SYNCH
Statement-list	None	None	Statement Statement-list	None	None
Statement	None	None	Selection-stmt	SYNCH	None
Expression-stmt	None	None	SYNCH	SYNCH	None
Selection-stmt	None	None	if (Expression) Statement else Statement	SYNCH	None
Iteration-stmt	None	None	SYNCH	SYNCH	None
Return-stmt	None	None	SYNCH	SYNCH	None
Return-stmt-prime	None	None	SYNCH	SYNCH	None
Switch-stmt	None	None	SYNCH	SYNCH	None
					None
Case-stmts Case-stmt	None	None	None	None	
	None	None	None	None	None
Default-stmt	None	None	None	None	None
Expression	None	None	None	None	None
В	Simple-expression-prime	Simple-expression-prime	None	None	None
Н	G D C	G D C	None	None	None
Simple-expression-zegond	None	None	None	None	None
Simple-expression-prime	Additive-expression-prime C	Additive-expression-prime C	None	None	None
C	Relop Additive-expression	Relop Additive-expression	None	None	None
Relop	<	==	None	None	None
Additive-expression	None	None	None	None	None
Additive-expression-prime	Term-prime D	Term-prime D	None	None	None
dditive-expression-zegond	SYNCH	SYNCH	None	None	None
D	EPSILON	EPSILON	None	None	None
Addop	None	None	None	None	None
Term	SYNCH	SYNCH	None	None	None
Term-prime	Factor-prime G	Factor-prime G	None	None	None
Term-zegond	SYNCH	SYNCH	None	None	None
G	EPSILON	EPSILON	None	None	None
Factor	SYNCH	SYNCH	None	None	None
Var-call-prime	Var-prime	Var-prime	None	None	None
Var-prime Var-prime	EPSILON	EPSILON	None	None	None
Factor-prime	EPSILON	EPSILON	None	None	None
	SYNCH	SYNCH	None	None	None
Factor-zegond					
Args	None	None	None	None	None
Arg-list	None	None	None	None	None
Arg-list-prime	None	None	None	None	None

4.5 parsetable no.5

	int	while	break	continue	switch
Program	Declaration-list	None	None	None	None
Declaration-list	Declaration Declaration-list	EPSILON	EPSILON	EPSILON	EPSILON
Declaration	Declaration-initial Declaration-prime	SYNCH	SYNCH	SYNCH	SYNCH
Declaration-initial	Type-specifier ID	None	None	None	None
Declaration-prime	SYNCH	SYNCH	SYNCH	SYNCH	SYNCH
Var-declaration-prime	SYNCH	SYNCH	SYNCH	SYNCH	SYNCH
Fun-declaration-prime	SYNCH	SYNCH	SYNCH	SYNCH	SYNCH
Type-specifier	int	None	None	None	None
Params	int ID Param-prime Param-list	None	None	None	None
Param-list-void-abtar	None	None	None	None	None
Param-list	None	None	None	None	None
Param	Declaration-initial Param-prime	None	None	None	None
Param-prime	None	None	None	None	None
Compound-stmt	SYNCH	SYNCH	SYNCH	SYNCH	SYNCH
Statement-list	None	Statement Statement-list	Statement Statement-list	Statement Statement-list	Statement Statement-list
Statement	None	Iteration-stmt	Expression-stmt	Expression-stmt	Switch-stmt
Expression-stmt	None	SYNCH	break ;	continue ;	SYNCH
Selection-stmt	None	SYNCH	SYNCH	SYNCH	SYNCH
Iteration-stmt	None	while (Expression) Statement	SYNCH	SYNCH	SYNCH
Return-stmt	None	SYNCH	SYNCH	SYNCH	SYNCH
Return-stmt-prime	None	SYNCH	SYNCH	SYNCH	SYNCH
Switch-stmt	None	SYNCH	SYNCH	SYNCH	switch (Expression) { Case-stmts Default-stmt
Case-stmts	None	None	None	None	None
Case-stmts Case-stmt	None	None	None	None	None
Default-stmt	None	None	None	None	None
	None	None	None	None	None
Expression B	None	None	None	None	None
Н Н					
	None None	None None	None None	None None	None None
Simple-expression-zegond					
Simple-expression-prime	None	None	None	None	None
C	None	None	None	None	None
Relop	None	None	None	None	None
Additive-expression	None	None	None	None	None
Additive-expression-prime	None	None	None	None	None
Additive-expression-zegond	None	None	None	None	None
D	None	None	None	None	None
Addop	None	None	None	None	None
Term	None	None	None	None	None
Term-prime	None	None	None	None	None
Term-zegond	None	None	None	None	None
G	None	None	None	None	None
Factor	None	None	None	None	None
Var-call-prime	None	None	None	None	None
Var-prime	None	None	None	None	None
Factor-prime	None	None	None	None	None
Factor-zegond	None	None	None	None	None
Args	None	None	None	None	None
Arg-list	None	None	None	None	None
Arg-list-prime	None	None	None	None	None

4.6 parsetable no.6

	default	case	return	\$
Program	None	None	None	Declaration-lis
Declaration-list	EPSILON	EPSILON	EPSILON	EPSILON
Declaration	SYNCH	SYNCH	SYNCH	SYNCH
Declaration-initial	None	None	None	None
Declaration-prime	SYNCH	SYNCH	SYNCH	SYNCH
Var-declaration-prime	SYNCH	SYNCH	SYNCH	SYNCH
Fun-declaration-prime	SYNCH	SYNCH	SYNCH	SYNCH
Type-specifier	None	None	None	None
Params	None	None	None	None
Param-list-void-abtar	None	None	None	None
Param-list	None	None	None	None
Param	None	None	None	None
Param-prime	None	None	None	None
Compound-stmt	SYNCH	SYNCH	SYNCH	SYNCH
Statement-list	EPSILON	EPSILON	Statement Statement-list	None
Statement	SYNCH	SYNCH	Return-stmt	None
Expression-stmt	SYNCH	SYNCH	SYNCH	None
Selection-stmt	SYNCH	SYNCH	SYNCH	None
Iteration-stmt	SYNCH	SYNCH	SYNCH	None
Return-stmt	SYNCH	SYNCH	return Return-stmt-prime	None
Return-stmt-prime	SYNCH	SYNCH	SYNCH	None
Switch-stmt	SYNCH	SYNCH	SYNCH	None
Case-stmts	EPSILON	Case-stmt Case-stmts	None	None
Case-stmt	SYNCH	case NUM : Statement-list	None	None
Default-stmt	default : Statement-list	None	None	None
Expression	None	None	None	None
В	None	None	None	None
 H	None	None	None	None
Simple-expression-zegond	None	None	None	None
Simple-expression-prime	None	None	None	None
С	None	None	None	None
Relop	None	None	None	None
Additive-expression	None	None	None	None
Additive-expression-prime	None	None	None	None
Additive-expression-zegond	None	None	None	None
D	None	None	None	None
Addop	None	None	None	None
Term	None	None	None	None
Term-prime	None	None	None	None
Term-zegond	None	None	None	None
G	None	None	None	None
Factor	None	None	None	None
Var-call-prime	None	None	None	None
Var-prime	None	None	None	None
Factor-prime	None	None	None	None
Factor-zegond	None	None	None	None
Args	None	None	None	None
Arg-list	None	None	None	None
vig-ner	None	TOHE	TAOHE	TAOHE