

Modified Grammar

GROUP NO: 40

Siddharth Sharma	2019A7PS0064P
Archit Bhatnagar	2019A7PS0133P
Suchismita Tripathy	2019A7PS0554P
Srujan Deolasee	2019A7PS1139P
Atharva Chandak	2019A7PS0062P

- ① $\langle \text{program} \rangle \rightarrow \langle \text{otherFunctions} \rangle \langle \text{mainFunction} \rangle$
- ② $\langle \text{mainFunction} \rangle \rightarrow \text{TK-MAIN} \langle \text{stmts} \rangle \text{TK-END}$
- ③ $\langle \text{otherFunctions} \rangle \rightarrow \langle \text{function} \rangle \langle \text{otherFunctions} \rangle \mid \epsilon$
- ④ $\langle \text{function} \rangle \rightarrow \text{TK-FUNID} \langle \text{input-par} \rangle \langle \text{output-par} \rangle \text{TK-SEM} \langle \text{stmts} \rangle \text{TK-END}$
- ⑤ $\langle \text{input-par} \rangle \rightarrow \text{TK-INPUT} \text{TK-PARAMETER} \text{TK-LIST} \text{TK-SQL} \langle \text{parameter-list} \rangle \text{TK-SQR}$
- ⑥ $\langle \text{output-par} \rangle \rightarrow \text{TK-OUTPUT} \text{TK-PARAMETER} \text{TK-LIST} \text{TK-SQL} \langle \text{parameters-list} \rangle \text{TK-SQR} \mid \epsilon$
- ⑦ $\langle \text{parameters-list} \rangle \rightarrow \langle \text{dataType} \rangle \text{TK-ID} \langle \text{remaining-list} \rangle$
- ⑧ $\langle \text{dataType} \rangle \rightarrow \langle \text{primitiveDataType} \rangle \mid \langle \text{constructedDataType} \rangle$
- ⑨ $\langle \text{primitiveDataType} \rangle \rightarrow \text{TK-INT} \mid \text{TK-REAL}$
- ⑩ $\langle \text{constructedDataType} \rangle \rightarrow \text{TK-RECORD} \text{TK-AUID} \mid \text{TK-UNION} \text{TK-AUID}$
- ⑪ $\langle \text{remaining-list} \rangle \rightarrow \text{TK-COMMA} \langle \text{parameter-list} \rangle \mid \epsilon$
- ⑫ $\langle \text{stmts} \rangle \rightarrow \langle \text{typeDefinitions} \rangle \langle \text{declarations} \rangle \langle \text{otherStmts} \rangle \langle \text{returnStmt} \rangle$
- ⑬ $\langle \text{typeDefinitions} \rangle \rightarrow \langle \text{typeDefinition} \rangle \langle \text{typeDefinitions} \rangle \mid \epsilon$
- ⑭ $\langle \text{typeDefinition} \rangle \rightarrow \text{TK-RECORD} \text{TK-AUID} \langle \text{fieldDefinitions} \rangle \text{TK-ENDRECORD}$
- ⑮ $\langle \text{typeDefinition} \rangle \rightarrow \text{TK-UNION} \text{TK-AUID} \langle \text{fieldDefinitions} \rangle \text{TK-ENDUNION}$
- ⑯ $\langle \text{fieldDefinitions} \rangle \rightarrow \langle \text{fieldDefinition} \rangle \langle \text{fieldDefinition} \rangle \langle \text{moreFields} \rangle$
- ⑰ $\langle \text{fieldDefinition} \rangle \rightarrow \text{TK-TYPE} \langle \text{dataType} \rangle \text{TK-COLON} \text{TK-ID} \text{TK-COLON} \langle \text{global-or-not} \rangle \text{TK-SEM}$

- (18) $\langle \text{more Fields} \rangle \rightarrow \langle \text{Field Definition} \rangle \langle \text{more Fields} \rangle \mid \epsilon$
- (19) $\langle \text{declarations} \rangle \rightarrow \langle \text{declaration} \rangle \langle \text{declarations} \rangle \mid \epsilon$
- (20) $\langle \text{declaration} \rangle \rightarrow \text{TK-TYPE} \langle \text{dataType} \rangle \text{TK-COLON} \text{TK-ID} \text{TK-COLON}$
 $\langle \text{global-or-not} \rangle \text{TK-SEM}$
- (21) $\langle \text{global-or-not} \rangle \rightarrow \text{TK-GLOBAL} \mid \epsilon$
- (22) $\langle \text{other Stmt's} \rangle \rightarrow \langle \text{stmt} \rangle \langle \text{other Stmt's} \rangle \mid \epsilon$
- (23) $\langle \text{stmt} \rangle \rightarrow \langle \text{assignment Stmt} \rangle \mid \langle \text{iterative Stmt} \rangle \mid \langle \text{conditional Stmt} \rangle \mid \langle \text{io Stmt} \rangle \mid$
 $\langle \text{funCall Stmt} \rangle \mid \langle \text{define type Stmt} \rangle$
- (24) $\langle \text{assignment Stmt} \rangle \rightarrow \langle \text{Single Or Rec Id} \rangle \text{TK-ASSIGNOP} \langle \text{arithmetic Expression} \rangle \text{TK-SEM}$
- (25) $\langle \text{Single Or Rec Id} \rangle \rightarrow \text{TK-ID} \langle \text{NEW Single Or Rec Id} \rangle$
- (26) $\langle \text{NEW Single Or Rec Id} \rangle \rightarrow \text{TK-DOT} \text{TK-FIELDID} \mid \epsilon$
- (27) $\langle \text{output Parameters} \rangle \rightarrow \text{TK-SQL} \langle \text{idList} \rangle \text{TK-SQR} \text{TK-ASSIGNOP} \mid \epsilon$
- (28) $\langle \text{input Parameters} \rangle \rightarrow \text{TK-SQL} \langle \text{idList} \rangle \text{TK-SQR}$
- (29) $\langle \text{iterative Stmt} \rangle \rightarrow \text{TK-WHILE} \text{TK-OP} \langle \text{boolean Expression} \rangle \text{TK-CL} \langle \text{stmt} \rangle$
 $\langle \text{other Stmt's} \rangle \text{TK-ENDWHILE}$
- (30) $\langle \text{conditional Stmt} \rangle \rightarrow \text{TK-IF} \langle \text{boolean Expression} \rangle \text{TK-THEN} \langle \text{stmt} \rangle \langle \text{other Stmt's} \rangle$
 $\langle \text{conditional Stmt END} \rangle$
- (31) $\langle \text{conditional Stmt END} \rangle \rightarrow \text{TK-ELSE} \langle \text{other Stmt's} \rangle \text{TK-ENDIF} \mid \text{TK-ENDIF}$
- (32) $\langle \text{io Stmt} \rangle \rightarrow \text{TK-READ} \text{TK-OP} \langle \text{var} \rangle \text{TK-CL} \text{TK-SEM} \mid \text{TK-WRITE} \text{TK-OP}$
 $\langle \text{var} \rangle \text{TK-CL} \text{TK-SEM}$
- (33) $\langle \text{funCall Stmt} \rangle \rightarrow \langle \text{output Parameters} \rangle \text{TK-CALL} \text{TK-FUNID} \text{TK-WITH}$
 $\text{TK-PARAMETERS} \langle \text{input Parameters} \rangle$
- (34) $\langle \text{arithmetic Expression} \rangle \rightarrow \langle \text{arithmetic Expression 1} \rangle \langle A' \rangle$
- (35) $\langle A' \rangle \rightarrow \langle \text{op1} \rangle \langle \text{arithmetic Expression 1} \rangle \langle A' \rangle \mid \epsilon$
- (36) $\langle \text{arithmetic Expression 1} \rangle \rightarrow \langle \text{arithmetic Expression 2} \rangle \langle A'' \rangle$
- (37) $\langle A'' \rangle \rightarrow \langle \text{op2} \rangle \langle \text{arithmetic Expression 2} \rangle \langle A'' \rangle \mid \epsilon$

- 38) $\langle \text{Arithmetic Expression 2} \rangle \rightarrow (\text{Arithmetic Expression}) \mid \text{Var}$
- 39) $\langle \text{op1} \rangle \rightarrow \text{TK-PLUS} \mid \text{TK-MINUS}$
- 40) $\langle \text{op2} \rangle \rightarrow \text{TK-DIV} \mid \text{TK-MUL}$
- 41) $\langle \text{boolean Expression} \rangle \rightarrow \text{TK-OP} \langle \text{boolean Expression} \rangle \text{TK-CL} \langle \text{Logical Op} \rangle$
 $\text{TK-OP} \langle \text{boolean Expression} \rangle \text{TK-CL}$
- 42) $\langle \text{boolean Expression} \rangle \rightarrow \langle \text{Var} \rangle \langle \text{relational Op} \rangle \langle \text{Var} \rangle$
- 43) $\langle \text{boolean Expression} \rangle \rightarrow \text{TK-NOT} \langle \text{boolean Expression} \rangle$
- 44) $\langle \text{Var} \rangle \rightarrow \text{TK-NUM} \mid \text{TK-RNUM} \mid \langle \text{Single Or RecId} \rangle$
- 45) $\langle \text{Logical Op} \rangle \rightarrow \text{TK-AND} \mid \text{TK-OR}$
- 46) $\langle \text{relational Op} \rangle \rightarrow \text{TK-LT} \mid \text{TK-LE} \mid \text{TK-EQ} \mid \text{TK-GT} \mid \text{TK-GE} \mid \text{TK-NE}$
- 47) $\langle \text{return Stmt} \rangle \rightarrow \text{TK-RETURN} \langle \text{optional Return} \rangle \text{TK-SEM}$
- 48) $\langle \text{idList} \rangle \rightarrow \text{TK-ID} \langle \text{more-ids} \rangle$
- 49) $\langle \text{more-ids} \rangle \rightarrow \text{TK-COMMA} \langle \text{idList} \rangle \mid \epsilon$
- 50) $\langle \text{define type Stmt} \rangle \rightarrow \text{TK-DEFINETYPE} \langle A \rangle \text{TK-RUID} \text{TK-AS} \text{TK-RUID}$
- 51) $\langle A \rangle \rightarrow \text{TK-RECORD} \mid \text{TK-UNION}$

Changes:

- 52) $\langle \text{optional Return} \rangle \rightarrow \text{TK-SQL} \langle \text{idList} \rangle \text{TK-SQR} \mid \epsilon$

Changes: (Explanation)

- ① We replaced $\langle \text{constructedDataType} \rangle$ with $\langle \text{dataType} \rangle$ in the production rule of $\langle \text{fieldDefinition} \rangle$ so that nested records can be obtained from the language.
- ② We included the non-terminal $\langle \text{defnTypestmt} \rangle$ in the production rule of $\langle \text{stmt} \rangle$ else it would have been an unreachable state in the grammar.
- ③ We changed the production rule of $\langle \text{SingleOrRecId} \rangle$ so that it becomes left factored.
- ④ We changed the production rule of $\langle \text{Arithmetic Expression} \rangle$ so that there is precedence among the operators and to remove left recursion.
- ⑤ The non-terminal $\langle \text{operator} \rangle$ is replaced with $\langle \text{op1} \rangle$ and $\langle \text{op2} \rangle$ based on the precedence of operators.
- ⑥ The non-terminal $\langle \text{TK-ID} \rangle$ in the production rule of $\langle \text{vars} \rangle$ is replaced with $\langle \text{SingleOrRecId} \rangle$ so that arithmetic operations can also be performed on records. The type checking of operands in an arithmetic expression is handled by the semantic analyzer.
- ⑦ The two production rules of $\langle \text{conditional Stmt} \rangle$ are modified so that it becomes left factored.

Assumptions:

- ① It is assumed that a record is already defined before it is nested inside another record.

FIRST AND FOLLOW SETS:

Non - Terminal	FIRST SET	FOLLOW SET
<program>	{TK-FUNID, TK-MAIN}	{ \$ }
<main Function>	{TK-MAIN}	{ \$ }
<other Functions>	{TK-FUNID, ϵ }	{TK-MAIN}
<Function>	{TK-FUNID}	{TK-FUNID, TK-MAIN}
<input-par>	{TK-INPUT}	{TK-OUTPUT, TK-SEM}
<output-par>	{TK-OUTPUT, ϵ }	{TK-SEM}
<parameter_list>	{TK-INT, TK-REAL, TK-RECORD, TK-UNION}	{TK-SQR}
<data Type>	{TK-INT, TK-REAL, TK-RECORD, TK-UNION}	{TK-ID ₂ , TK-COLON}
<primitive Data type>	{TK-INT, TK-REAL}	{TK-ID ₂ , TK-COLON}
<constructed Data type>	{TK-RECORD, TK-UNION}	{TK-ID, TK-COLON}
<remaining_list>	{TK-COMMA, ϵ }	{TK-SQR}
<stmts>	{TK-RECORD, TK-UNION, TK-TYPE, TK-ID, TK-WHILE, TK-IF, TK-READ, TK-WRITE, TK-SQL, TK-CALL, TK-DEFINETYPE, TK-ENDWHILE, TK-ENDIF, TK-ELSE, TK-RETURN}	{TK-END}
<type Definitions>	{TK-RECORD, TK-UNION, ϵ }	{TK-TYPE, TK-RECORD, TK-UNION, TK-ID, TK-WHILE, TK-IF, TK-READ, TK-WRITE, TK-SQL, TK-CALL, TK-DEFINETYPE, TK-ENDIF, TK-ENDWHILE, TK-ELSE, TK-RETURN}

Non-Terminal	FIRST SET	FOLLOW SET
<type Definition>	{TK-RECORD, TK-UNION}	{TK-RECORD, TK-UNION, TK-TYPE, TK-ID, TK-WHILE, TK-IF, TK-READ, TK-WRITE, TK-SQL, TK-CALL, TK-ELSE, TK-ENDWHILE, TK-ENDIF, TK-RETURN, TK-DEFINETYPE }
<field Definitions>	{TK-TYPE}	{TK-ENDRECORD, TK-UNION } TK-ENDUNION
<field Definition>	{TK-TYPE}	{TK-TYPE, TK-ENDRECORD, TK-ENDUNION}
<more fields> <declarations>	{TK-TYPE, ϵ }	{TK-ID, TK-WHILE, TK-IF, TK-READ, TK-WRITE, TK-SQL, TK-CALL, TK-ELSE, TK-ENDWHILE, TK-ENDIF, TK-RETURN, TK-DEFINETYPE}
<declarations> <more Fields>	{TK-TYPE, ϵ }	{TK-ENDRECORD, TK-ENDUNION}
<declaration>	{TK-TYPE}	{TK-TYPE, TK-ID, TK-WHILE, TK-IF, TK-READ, TK-WRITE, TK-SQL, TK-CALL, TK-ELSE, TK-ENDWHILE, TK-ENDIF, TK-RETURN, TK-DEFINETYPE}
<global-or-not>	{TK-GLOBAL, ϵ }	{TK-SEM}
<other stmts>	{TK-ID, TK-WHILE, TK-IF, TK-READ, TK-WRITE, TK-SQL, TK-CALL, TK-DEFINETYPE, ϵ }	{TK-ENDWHILE, TK-ELSE, TK-ENDIF, TK-RETURN}

Non Terminal	FIRST SET	FOLLOW SET
$\langle \text{stmt} \rangle$	$\{ \text{TK-ID}, \text{TK-WHILE}, \text{TK-IF}, \text{TK-READ}, \text{TK-WRITE}, \text{TK-SQL}, \text{TK-CALL}, \text{TK-DEFINETYPE} \}$	$\{ \text{TK-ID}, \text{TK-WHILE}, \text{TK-IF}, \text{TK-READ}, \text{TK-WRITE}, \text{TK-SQL}, \text{TK-CALL}, \text{TK-ELSE}, \text{TK-ENDWHILE}, \text{TK-ENDIF}, \text{TK-DEFINETYPE}, \text{TK-RETURN} \}$
$\langle \text{assignment Stmt} \rangle$	$\{ \text{TK-ID} \}$	$\{ \text{TK-ID}, \text{TK-WHILE}, \text{TK-IF}, \text{TK-READ}, \text{TK-WRITE}, \text{TK-SQL}, \text{TK-CALL}, \text{TK-ELSE}, \text{TK-ENDWHILE}, \text{TK-ENDIF}, \text{TK-RETURN}, \text{TK-DEFINETYPE} \}$
$\langle \text{single Or RecId} \rangle$	$\{ \text{TK-ID} \}$	$\{ \text{TK-ASSIGNOP} \}$
$\langle \text{NewSingle Or RecId} \rangle$	$\{ \text{TK-DOT}, \epsilon \}$	$\{ \text{TK-ASSIGNOP} \}$
$\langle \text{fun call Stmt} \rangle$	$\{ \text{TK-SQL}, \text{TK-CALL} \}$	$\text{FOLLOW}(\langle \text{stmt} \rangle)$ (Please refer above)
$\langle \text{out put Parameters} \rangle$	$\{ \text{TK-SQL}, \epsilon \}$	$\{ \text{TK-CALL} \}$
$\langle \text{input Parameters} \rangle$	$\{ \text{TK-SQL} \}$	$\text{FOLLOW}(\langle \text{stmt} \rangle)$ (Please refer above)
$\langle \text{iterative Stmt} \rangle$	$\{ \text{TK-WHILE} \}$	$\text{FOLLOW}(\langle \text{stmt} \rangle)$ (Please refer above)
$\langle \text{conditional Stmt} \rangle$	$\{ \text{TK-IF} \}$	$\text{FOLLOW}(\langle \text{stmt} \rangle)$ (Please refer above)
$\langle \text{conditional Stmt END} \rangle$	$\{ \text{TK-ELSE}, \text{TK-ENDIF} \}$	$\text{FOLLOW}(\langle \text{stmt} \rangle)$ (Please refer above)
$\langle \text{io Stmt} \rangle$	$\{ \text{TK-READ}, \text{TK-WRITE} \}$	$\text{FOLLOW}(\langle \text{stmt} \rangle)$ (Please refer above)
$\langle \text{arithmetic Expression} \rangle$	$\{ \text{TK-OP}, \text{TK-ID}, \text{TK-NUM}, \text{TK-RNUM} \}$	$\{ \text{TK-CL}, \text{TK-SEM} \}$
$\langle \text{arithmetic Expression 1} \rangle$	$\{ \text{TK-OP}, \text{TK-ID}, \text{TK-NUM}, \text{TK-RNUM} \}$	$\{ \text{TK-PLUS}, \text{TK-MINUS}, \text{TK-CL}, \text{TK-SEM} \}$

Non-Terminal	FIRST SET	FOLLOW SET
$\langle A' \rangle$	$\{TK-PLUS, TK-MINUS, \epsilon\}$	$\{TK-CL, TK-SEM\}$
$\langle A'' \rangle$	$\{TK-MUL, TK-DIV, \epsilon\}$	$\{TK-PLUS, TK-MINUS, TK-CL, TK-SEM\}$
$\langle \text{Arithmetic Expression} \rangle$	$\{TK-OP, TK-ID, TK-NUM, TK-RNUM\}$	$\{TK-MUL, TK-DIV, TK-PLUS, TK-MINUS, TK-CL, TK-SEM\}$
$\langle op1 \rangle$	$\{TK-PLUS, TK-MINUS\}$	$\{TK-OP, TK-ID, TK-NUM, TK-RNUM\}$
$\langle op2 \rangle$	$\{TK-MUL, TK-DIV\}$	$\{TK-OP, TK-ID, TK-NUM, TK-RNUM\}$
$\langle \text{Boolean Expression} \rangle$	$\{TK-OP, TK-ID, TK-NUM, TK-RNUM, TK-NOT\}$	$\{TK-CL, TK-THEN\}$
$\langle Var \rangle$	$\{TK-ID, TK-NUM, TK-RNUM\}$	$\{TK-LT, TK-LE, TK-EQ, TK-GT, TK-GE, TK-NE, TK-CL, TK-THEN, TK-PLUS, TK-MINUS, TK-MUL, TK-DIV, TK-SEM\}$
$\langle \text{Logical Op} \rangle$	$\{TK-AND, TK-OR\}$	$\{TK-OP\}$
$\langle \text{Relational Op} \rangle$	$\{TK-LT, TK-LE, TK-EQ, TK-GT, TK-GE, TK-NE\}$	$\{TK-ID, TK-NUM, TK-RNUM\}$
$\langle \text{return stmt} \rangle$	$\{TK-RETURN\}$	$\{TK-END\}$
$\langle \text{optional return} \rangle$	$\{TK-SQL?, \epsilon\}$	$\{TK-SEM\}$
$\langle \text{idList} \rangle$	$\{TK-ID\}$	$\{TK-SQR\}$
$\langle \text{more-ids} \rangle$	$\{TK-COMMA?, \epsilon\}$	$\{TK-SQR\}$
$\langle \text{define type stmt} \rangle$	$\{TK-DEFINETYPE\}$	Follow($\langle \text{stmt} \rangle$) (Please refer above)
$\langle A \rangle$	$\{TK-RECORD, TK-UNION\}$	$\{TK-RUID\}$