COMPILER CONSTRUCTION ASSIGNMENT. PAPER WORK - GRAMMAR (LLGD).

1. The definetype smt > was unreachable in the original grammar. since the addinctype smt > is used in the type definition of constructed datatypes (second, cenion) = it is a talement which. would be present in the typed of intern section. Home, appropriate production sule: < type Definition> > definetype stant> introduced.

XASH 2019 A7 PSI 1381 20. The apsimite Datalype > in rule 17 is replaced with GUPTA language specifications? coute Type > in order to allow nested records faccosding to

GROUP NUMBER 18.

SANKMA 2019A7150029A

L- 2019AAPS0063

2019 A7PS 0243 F

2019 A7 PSOUS IP

MADHAV GUPTA.

DAS. MEENAL

GUPTA . PRATHAM

GVPTA

3 . Left Recursion is sufficiently handled.

4. Ambiguity removed by imbodiction of left anoclability & precedence rules 5. Left factoring done.

6. Related rules ore grouped together in blocks.

The modified grammer rules are written in red. The remaining are mentioned in blue

1. < program > > < other functions > < main Function>

2. < mountanition> > TK-MAIN < Strats>TK-END. 3. <older Functions>> < function × older Functions > 1 Exilon (E)

4. (fundion) -> TK_FUNID < input-per> < output_por> TK_SEN < stmts> TK_END

5. <input_par> > TK_INPUT. TK_PARAMETER TK_LIST. TK_SQL < porameter_list > TK_SQR.
6. <out put-par> > TK_OUTPUT. TK_PARAMETER. TK_LIST TK_SQL < porameter_list > TK_SQR &
7. <porameter_list >> <dataType> TK_D

3. data Type> > <pri> > Frim Hive Dalatype> (<constructed Dalatype>

9. < Primitive Dalatype> > TK-INTITK-REAL

10. <Constructed Datatype > > TK-RECORD TK_RUID | TK-WION TK-RUID

11. Eremaining-list >> TK-COMMA <porametr_list>18

12 estate> > < type Definitions > < declarations < obser State >< refurastant>

<type Definition> > <type Definition> < type Definition> < type Definition> > 1 &

<type Definition> > TK-RECORD TK-RUID < field Definition> TK-ENDUNION/
TK_UNION TK_UNION < field Definition> TK-ENDUNION/ < definety peshnt?

15. < field Definition > > < field Definition > < field Definition < field Definitions < more fields

corore Fields>> < field Definition> < morefielde> 1 &

Field Definition> > TK-TYPE < dataType> TK-COLON TK_FIELDID. TK_SEM

<declarations> > <declaration> <declarations > 1 &

<declaration> -> TK-TIPE <data Type> TK-COLON TK-ID TK-COLON Zglobal-on-orof>TK-SEM. 19.

20. Eglobal -02-not> +TK-GLOBAL/E

Cother stmls>> <stmt>cother stmts>1 &

25tmt>> <amign ment stmt> 1 <ile rative stmt> / <cendutional stmt> / <ioshnt>/ 22.

< furball Smt >

< funcall stmt > > Confort Parameters > TK_CALL TK-FUNID TK-WITH TK_PARAMETERS

27. Cauput Posamelos > > TK-SQL Cid List > TK-SQR TK-ASSIGNOP 18

28. Limpud Parameters > > TK-SPL < id List > TKSQR.

```
29. < [porativestmt > > TK-WHILE TK-OF Choolean Expression > TK-CL Estat > collerstand> TK-ELANDER
30 - Coonditional Start > -> TK-IF < booklane xpression> TK-THEN Comb> < other Starter
                        < conditional strat siglix>.
31. <conditionalshit suffix> > TK-ELSE <other shits > END IF | ENDIF
32. LOSIMT> > TK-READ TK-OP (VOO) TK-CL TK-SEMI
                TK. WRITE TK-OP ( VOST TK-CL TK-SEM.
      is eyl associativity and precidence information introduced.
        Enordes to Fernove ambiguity. 2 left recursion removed.
     conftrmetic Expression>> 
     Easi Hancuc Expression Prime> > 20ddorsub> 2001thmete Expression Polime> /8.
    coold 02 sub > > TK-PLUS | TK-MINUS.
34.
     ck>m Prime> > cmulospir>
ck>m Asime> = cmulospir>
    = k>m> -> < factor > < k>mp> me>
    <. mulor Div> -> TK_MULITK-DIV.
     cfacto> -> TK-OF con'th m ctic Expression> TK-CL | < vor>.
    ∠boolian Expression> > TK-OF < boolian Expression> TK-CL < logical op>.TK-OP.
                             Lvor > Zrelational Of > Lvor>. 1. TK's NOT Goodsan Expressions
 41" (VOO) > TK-IDITK-NUM TK-RNUM.
 43. < relational OPT > TK-LTITK-LEITK_EQITK-UTITK-GEI TK-NE
 44. Exelum Stont > -> TK-RETURN coptinal Return > TK-SEM.
 46. coptional Return > TK-SQ L cidList > TK-SQRIE.
· 46. <idList> -> TK-ID Cmore_ide>
 43. cmore ids> > TK-comma cid List> / E.
      <definetypeshnt> -> TK-DEFINETYPE <A> TK-RUID TK-AS
     CA> -3 TK- RECORD | TK-UNION.
```

```
- GROUP 18
CS F363
                    2019A7850063P
           GWPTA
  MADHAY
                   2019A7PS0243P
           GUPTA
  MEENAL
                   2019 A7PS OOSIP
           GUPTA
  PRATHAM
                   2019A7PS0029P
           DAS
  SANKHA
                   2019 ATPS 1138P
           GUPTA
  YASH
                    THE MO-DIFIED GRAMMAR
FIRST SETS
               FOR
 FIRST ( < psysom) = { TK - MAW, TK-FUNIO }
 FIRST (< monification>) = { TK - MAIN }
 FIRST ( < Other Functions > ) = { TK_FUNIO, E }
FIRST (< function>) = { TK_FUNIO }
FIRST (< input=fax >) = { TK-INPUT }
FIRST (< OUTPUT = pax >) = { TK-OUTPUT =}
FIRST (< parometer-list) = { TK-INT, TK-REAL, TK-REIGR D,
            TK_WON = FIRST ( < data Type > )
FIRST (Contitue Doubtype>) = {TK-INT, TK-REAL }
FIRST (< constructed Owntype >) = { TK_RECORD, TK_UNION }
FIRST (< runoning - bot >) = {TK- comma}
FIRST ( < STMTS > ) = { TK_RETURN, TK_UMON, TK_RECORD, TK_TYPE,
  TK_ DEFINETYPE, TK_WHILE, TK_ IF, TK_ READ, TK_WESTE,
    TK-CALL, TK-TO, TK-SOL }
FIRST (< type Definitions >) = {TK_KECORD, TK_UMON.
TK_CEFINETYTE, E}
FIRST ( < type Definition >) = { TK-RECORD, TK-UMON, TK-DEFINETIME }
FIRST (< fuld Definitions) = FIRST (< fuld Definition))
       = FIRST (Kdularab can) = {TK_TYPE}
FIRST (< monFulde >) = FIRST (< dutalstrons >)
              = {TK_TYME, E}
FIRST ( < global - ca - not >) = { TK - GLOBAL, E }
```

```
FIRST (< other start)) = { E, TK-WHILE, TK-IF, TK READ,
          TK-WRITE, TK-CALL, TK-10, TK-SOL }
FIRST ((stut)) = { TK-WHILE, TK-IF, TK-PERO, TK, WPINE, TK-CALL,
                                                  TK-10, TK-806]
FIRST ( < assignment forts) = FIRST ( < single Or Rezd >) = ETK_10}
FIRST ( < single OIRe Zd& # X>) = { E, TK-DOT }
FIRST (< fun Call (Int)) = { TK-CALL, TK-OLL}
FIRST ( ( Outfut Parameters) = { E TK-WL}
FIRST (< Mpw tasomeless) = { TK-SOL}
FIRST (< statue stant)) = {TK-WHILE}
FIRST (< conditional Omt)) = } TK-IR]
FROT (< conditional ofmitselfix >) = { TK- FEDE, TK-ENDIE }
FIRST ((idofut)) = { TK-REND, TK-WRITE}
FIRST (< coult metic Explession)) = { TK_OB, TK_10, TK_10M, TK_PNOTH}
FIRST ( < Olithmetic Explession Prince >) = {TK_PLUS, TK_MUNUS, E}
FIRST (< add OsSub->) = { TK_PLUS, TK_MINES }
FIRST ( < log TK - P, TK - 10, TK - RUM)

FIRST ( < log TK - RUM)

= KIRST ( < log TK - RUM)
FIRST (< mell(0x0iv)) = {TK-MELLS TK-DIV}
FIRST (< books = Expression >) = { TK-OP, TK-NOT, TK-10, TK-NUM, TK PNEM }
FIRST (< LOS) = { TK-10, TK-RUM, TK-RVOM}
FIRST (< legical Op)) = { TK-4NO, TK-OR }
FIRST (< substituted of )) = { TK-LT, TK-LE; TK EN TK-NE,
TK-GE, TK-OF}
 FIRST ( ( Setuln Sport )) ? ETK-RETURN }
PURST (< optional Return >) = { E, TK-SUL }
FIRST (<iSLA>) 2 } TK-10}
FIRST (< Mollids >) = { E, TK-COTUMA }
FIRST (< definityle Gas): {TK-DEFINETYPE}
 FIRST (<A7) = & TK-PEROS, TK_ CANON)
```

```
CS F363 - GARAUP 18
  FOLLOW SETS FOR THE MODIFIED GRAMMAR
  FOLLOW ( < program >) = FOLLOW ( < miniFrantion >): {$}
  FOLLOW ( Cathestructions >) = { TK-MAIN}
 FOLOW (< bunction>): {TK-MAIN, TK-FUNIO}
FOLOW (< niput-for>): {TK-OEM, TK-OTIPUT}
 FOLLOW (< promder-box>) = { TK-SEM }
FOLLOW (< promder-lot->) = { TK-SDR }
 FOLLOW (< dutatype >) 2 FOLLOW (< printitue Davatype >)
            = FOLLOW (< constructed outsityles) = {TK-10, TK-cozon}
 FOLLOW ( < Semanting-LB->) = { TK-DR }
 FOLLOW (< PMS>) = {TK-END}
FOLOW ( Stype Degnillone ) = {TK-READY, TK-WHILE, TK-IT, TK-READ, TK. WAITE, TK-COLL, TK-W, TK-WAILE, TK-TYPE }
FOLOW ( < by Definition >): STK-RETURNS TK-WHIES TK-IE

TK-KEAD, TK-WRITE, TK-COLL, TK-10, TK-DR, TK-TYPE, TK-ROWERS
                                             TK- UNON, TK-REF-NETHEZ
POLOW (< field Definitione >): Facor ((mesofield))
FOLLOW ( Chald Deficition >): { TK-ENDREGORD, TK-ENDLINON, TK-TIRE)
FOLLOW (< Sedentions): { TK-READ, TK-WHILE, TK-IF,
TK-SKL, TK-READ, TK-WRITE, TK-CALL, TK-10}
FOLLOW ((doclasution)) = FOLLOW ((declasations)) U {TK-TIPE}
FOLLOW (< global - O1 - NOT>): ETK-SEMS
FORW (< Ottos (Amts)): { TK-RETURNS TK-ENDWHILE, TK-BOS, TK-ENDE }
FOLOW (< englos fatd) : Forkow (< englos ketd Septix))
= STK_ ASSIGNOP }
```

```
FOLLOW (<Gont): POLLOW (< als grown out 57mt>): Follow (id dut))
   = FOLOW (< Absolite OPM+)) = FOLOW (< conditional OPM+))
   = Parow (< conditional on sylx>) = Follow (< byen Call onn+>)
   = FOLLOW < hipur Paromolers))
    - { TK-RETURN, TK-WHILES TK-IF, TK-READS TK-WRITE,
TK-CALLS TK-IDS TK-SELS TK-ENDWHILE, TK-ESE, TK-ENDIF
FOLOW (< cut ful farometers ) ] = { Tk-can }
FOLOW (Salttmat Expression): Forcow (Salttmat Expression Dina)
FOLLOW ( Ladd Os Sub->) = FOLLOW ( mul OLDIV >)
               = {TK-OB TK-10, TK-NM, TK-RNM}
FRION (< tem s): Falor (< tem Buine)
           = { TK-SEM, TK-PLUS, TK_MINUS, TK_CC}
FOLLOW ( < fortos >) = ETK-EM, TK-PUS, TK-MUL, TK-DING TK-MING,
FOLLOW (< bodson Expression>) = {TK_CL, TK_THERY}
FOLGW (<UN>): { TK-CL, TK-SEM, TK-PLUS, TK-MINUS, TK-MUZ,
TK-OV, TK-THEN, TK-LT, TK-CE, TK-LQ, TK-OT,
TK-GIE, TK-NE}

FOLLOW (<legical Op)) = {TK-OP}
FOLOW ( < lehtomop>) = { TK-10, TK-NUM, TK-RVUM }
FOLLOW (< return STAT >) = { TK-ENO}
FOLOW ( Optional Return >) = { TK-OEM }
                                                ETK-SUR)
PRION (< idlig>): Falar (mole_ids>):
FOLOW ( Colypietyke (Put >) = { TK-RETURNS TK-WHILE. TK-IF,
     TK- READ, TK-WRITE, TK-CALL, TK-10, TK-JOL,
      TK- TK- RECORDS - OK LONONS TK-DEFINE )
FOLLON (A>): { TK_RUIO}
```