First and Follow Sets

Group 9

Prithvi Raj - 2016A7PS0013P

Rahul Saxena - 2016A7PS0027P

Anuvind Bhat - 2016A7PS0098P

Siddhant Jain - 2016A7PS0113P

Assumptions

1) Some discrepancies were noticed in the given grammar and the examples given in the language specification.
   In the examples given, declarations of the following form have been used
   ```
   type real : c4 ;
   ```
   But the given grammar does not allow such a declaration. If global keyword is not used, even then a colon must follow the identifier, i.e. declarations must be of the following form
   ```
   type real : c4 : ;
   ```

2) We are also assuming that the logical operators @@@ (OR), &&& (AND) and ~ (NOT) have same precedence as the language specification has not defined any precedence and no precedence is implied by the grammar. It is observed that the boolean expressions given in the grammar enforce parenthesization so precedence is anyway never going to be used.

3) Another assumption is that for records, scalar multiplication allows arithmetic expression to be one operand and scalar multiplication/division has higher precedence over addition and subtraction of records. Eg-
   ```
   C3 <--- 4*C2 + C4*(1+3)
   ```

4) In rule 25 of the grammar, It is assumed there is a mistake and that the call statement ends with a semicolon (TK_SEM) as shown in the examples.

Changes to Grammar

- Left factoring in conditional statements. Note that there is no "dangling else" problem because of the " endif " keyword used at the end of every if and if-else.
  Replace 29 and 30 with.

```
<conditionalStmt> ===> TK_IF <booleanExpression> TK_THEN
     <stmt> <otherStmts> <elseStmt>
<elseStmt> ===> TK_ELSE <otherStmts> TK_ENDIF
     | TK_ENDIF
```

- Enforcing precedence of arithmetic operators
  Replace 33,34,35 with

```
<arithmeticExpression> ===> <term> <otherTerms>
<otherTerms> ===> TK_MINUS <term> <otherTerms>
     | TK_PLUS <term> <otherTerms> |eps
<term> ====> <factor> <otherFactors>
<otherFactors>===> TK_MUL <factors> <otherFactors>
     |TK_DIV <factors> <otherFactors> |eps
<factor> ===> <var> | TK_OP <arithmeticExpression> TK_CL
```

- Adding rules for record assignment and arithmetic of records. Rule 24 of the grammar is removed as a part of left factoring of the assignment statement

```
<assignmentStmt>===>TK_ID TK_ASSIGNOP <arithmeticExpression>
     TK_SEM|TK_ROCORDID <record_assignment>
<record_assignment> ===> TK_DOT TK_FIELDID TK_ASSIGNOP
     <arithmeticExpression> TK_SEM
     | TK_ASSIGNOP <record_expr> TK_SEM
<record_expr> ===> <record_term> <other_record_terms>
<other_record_terms> ===> TK_PLUS <record_term>
     <other_record_terms> | TK_MINUS <record_term>
     <other_record_terms> | eps
<record_term> ===> TK_RECORDID
     | TK_OP <arithmeticExpression> TK_CL TK_MUL TK_RECORDID
     | <var> TK_MUL TK_RECORDID
```

FIRST AND FOLLOW SETS

Non Terminal	First	Follow
<program>	TK_FUNID, TK_MAIN	$
<otherFunctions>	TK_FUNID, eps	TK_MAIN
<mainFunction>	TK_MAIN	$
<function>	TK_FUNID	TK_FUNID, TK_MAIN
<input_par>	TK_INPUT	TK_OUTPUT, TK_SEM
<output_par>	TK_OUTPUT, eps	TK_SEM
<parameter_list>	TK_INT, TK_REAL, TK_RECORD	TK_SQR
<dataType>	TK_INT, TK_REAL, TK_RECORD	TK_ID, TK_COLON
<primitiveDatatype>	TK_INT , TK_REAL	TK_ID, TK_COLON
<constructedDatatype>	TK_RECORD	TK_ID, TK_COLON
<remaining_list>	TK_COMMA, eps	TK_SQR
<stmts>	TK_RECORD, TK_TYPE, TK_ID, TK_RECORDID, TK_SQL, TK_CALL, TK_WHILE, TK_READ, TK_WRITE, TK_IF, TK_RETURN	TK_END
<typeDefinitions>	TK_RECORD,eps	TK_TYPE, TK_ID, TK_RECORDID, TK_SQL, TK_CALL, TK_WHILE, TK_READ , TK_WRITE, TK_IF, TK_RETURN
<typeDefinition>	TK_RECORD	TK_RECORD, TK_TYPE, TK_ID, TK_RECORDID, TK_SQL, TK_CALL, TK_WHILE, TK_READ ,

		TK_WRITE, TK_IF, TK_RETURN
<declarations>	TK_TYPE, eps	TK_ID, TK_RECORDID, TK_SQL, TK_CALL, TK_WHILE, TK_READ , TK_WRITE, TK_IF, TK_RETURN
<declaration>	TK_TYPE	TK_TYPE, TK_ID, TK_RECORDID, TK_SQL, TK_CALL, TK_WHILE, TK_READ , TK_WRITE, TK_IF, TK_RETURN
<otherStmts>	TK_ID, TK_RECORDID, TK_SQL, TK_CALL, TK_WHILE, TK_READ , TK_WRITE, TK_IF, eps	TK_RETURN, TK_ELSE, TK_ENDIF, TK_ENDWHILE
<stmt>	TK_ID, TK_RECORDID, TK_SQL, TK_CALL, TK_WHILE, TK_READ , TK_WRITE, TK_IF	TK_ID, TK_RECORDID, TK_SQL, TK_CALL, TK_WHILE, TK_READ , TK_WRITE, TK_IF, TK_RETURN, TK_ELSE, TK_ENDIF, TK_ENDWHILE
<assignmentStmt>	TK_ID, TK_RECORDID	TK_ID, TK_RECORDID, TK_SQL, TK_CALL, TK_WHILE, TK_READ , TK_WRITE, TK_IF, TK_RETURN, TK_ELSE, TK_ENDIF, TK_ENDWHILE
<funCallStmt>	TK_SQL , TK_CALL	TK_ID, TK_RECORDID, TK_SQL, TK_CALL, TK_WHILE, TK_READ , TK_WRITE, TK_IF, TK_RETURN, TK_ELSE, TK_ENDIF, TK_ENDWHILE
<outputParameters>	TK_SQL,eps	TK_CALL

<inputParameters>	TK_SQL	TK_SEM
<iterativeStmt>	TK_WHILE	TK_ID, TK_RECORDID, TK_SQL, TK_CALL, TK_WHILE, TK_READ , TK_WRITE, TK_IF, TK_RETURN, TK_ELSE, TK_ENDIF, TK_ENDWHILE
<ioStmt>	TK_READ , TK_WRITE	TK_ID, TK_RECORDID, TK_SQL, TK_CALL, TK_WHILE, TK_READ , TK_WRITE, TK_IF, TK_RETURN, TK_ELSE, TK_ENDIF, TK_ENDWHILE
<conditionalStmt>	TK_IF	TK_ID, TK_RECORDID, TK_SQL, TK_CALL, TK_WHILE, TK_READ , TK_WRITE, TK_IF, TK_RETURN, TK_ELSE, TK_ENDIF, TK_ENDWHILE
<elseStmt>	TK_ELSE, TK_ENDIF	TK_ID, TK_RECORDID, TK_SQL, TK_CALL, TK_WHILE, TK_READ , TK_WRITE, TK_IF, TK_RETURN, TK_ELSE, TK_ENDIF, TK_ENDWHILE
<allVar>	TK_RECORDID,TK_ID, TK_NUM, TK_RNUM	TK_CL
<var>	TK_ID, TK_NUM, TK_RNUM	TK_CL, TK_LT, TK_LE, TK_EQ, TK_GT, TK_GE, TK_NE, TK_SEM, TK_PLUS, TK_MUL, TK_MINUS, TK_DIV, TK_CL
<booleanExpression>	TK_OP, TK_ID, TK_NUM, TK_RNUM, TK_NOT	TK_CL

<logicalOp>	TK_AND, TK_OR	TK_OP
<relationalOp>	TK_LT, TK_LE, TK_EQ, TK_GT, TK_GE, TK_NE	TK_ID, TK_NUM, TK_RNUM
<returnStmt>	TK_RETURN	TK_END
<optionalReturn>	TK_SQL, eps	TK_SEM
<idList>	TK_ID	TK_SQR
<more_ids>	TK_COMMA, eps	TK_SQR
<global_or_not>	TK_GLOBAL,eps	TK_SEM
<arithmeticExpression>	TK_ID, TK_NUM, TK_RNUM,TK_OP	TK_SEM, TK_CL
<otherTerms>	TK_MINUS, TK_PLUS, eps	TK_SEM, TK_CL
<term>	TK_ID, TK_NUM, TK_RNUM, TK_OP	TK_MINUS, TK_PLUS, TK_SEM, TK_CL
<otherFactors>	TK_MUL, TK_DIV, eps	TK_MINUS, TK_PLUS, TK_SEM, TK_CL
<factor>	TK_ID, TK_NUM, TK_RNUM, TK_OP	TK_MUL, TK_DIV, TK_MINUS, TK_PLUS, TK_SEM, TK_CL
<record_assignment>	TK_DOT, TK_ASSIGNOP	TK_ID, TK_RECORDID, TK_SQL, TK_CALL, TK_WHILE, TK_READ , TK_WRITE, TK_IF, TK_RETURN, TK_ELSE, TK_ENDIF, TK_ENDWHILE
<record_expr>	TK_RECORDID, TK_OP, TK_ID, TK_NUM, TK_RNUM	TK_SEM

<other_record_terms>	TK_PLUS, TK_MINUS, eps	TK_SEM
<record_term>	TK_RECORDID, TK_OP, TK_ID, TK_NUM, TK_RNUM	TK_PLUS, TK_MINUS, TK_SEM