# List of Abbrevations

|  |  |
| --- | --- |
| *Statement* | A |
| *Expression* | B |
| *Factor* | C |
| *Program* | D |
| *Identifier\_list* | E |
| *Declarations* | F |
| *Subprogram\_declarations* | G |
| *Compound\_statement* | H |
| *Type* | I |
| *Standard\_type* | J |
| *Subprogram\_declaration* | K |
| *Subprogram\_head* | L |
| *Arguments* | M |
| *Parameter\_list* | N |
| *Optional\_statements* | O |
| *Statement\_list* | P |
| *Variable* | Q |
| *Procedure\_statement* | R |
| *Expression\_list* | S |
| *Simple\_expression* | T |
| *Term* | U |
| *Sign* | V |

# Replaced Conventions

|  |  |
| --- | --- |
| *program* ->  **program id** ( *identifier\_list* ) ;  *declarations*  *subprogram\_declarations*  *compound\_statement* | D -> **program id** ( E ) ; F G H |
| *identifier\_list* ->  **id**  | *identifier-list* , **id** | E -> **id** | E , **id** |
| declarations ->  *declarations***var***identifier-list* : *type* ;  | Î | F -> F **var** E : I ; | Î |
| type ->  *standard\_type*  | **array [ num . . num ] of** *standard\_type* | I -> J | **array [ num . . num ] of** J |
| *standard\_type* ->  **integer**  |**real** | J -> **integer** | **real** |
| *subprogram\_declarations* ->  *subprogram\_declarations subprogram\_declaration ;*  | Î | G -> G K ; | Î |
| *subprogram\_declaration* ->  *subprogram\_head declarations compound\_statement* | K -> L F H |
| *subprogram\_head* ->  **function id** *arguments* : *standard\_type*;  | **procedure id** *arguments* ; | L -> **function id** M : J ; | **procedure id** M ; |
| *arguments* ->  ( *parameter-list* )  | Î | M -> ( N ) | Î |
| *parameter\_list* ->  *identifier\_list* : *type*  | *parameter\_list* ; *identifier\_list*: *type* | N -> E : I | N ; E : I |
| *compound\_statement* ->  **begin**  *optional\_statements*  **end** | H -> **begin** O **end** |
| *optional\_statements* ->  *statement\_list*  | Î | O -> P | Î |
| *statement\_list* ->  *statement*  | *statement\_list* ; *statement* | P -> A | P ; A |
| *statement* ->  *variable***assignop** *expression*  | *procedure\_statement*  | *compound-statement*  | **if** *expression***then** *statement***else** *statement*  | **while** *expression***do** *statement* | A -> Q **assignop** B | R | H | **if** B **then** A **else** A | **while** B **do** A |
| *variable* ->  **id**  | **id** [ *expression*] | Q -> **id** | **id** [ B ] |
| *procedure\_statement*->  **id**  | **id** ( *expression-list* ) | R -> **id** | **id** ( S ) |
| *expression\_list* ->  *expression*  | *expression\_list* , *expression* | S -> B | S , B |
| *expression* ->  *simple\_expression*  | *simple\_expression***relop** *simple\_expression* | B -> T | T **relop** T |
| *simple\_expression* ->  *term*  | *sign term*  | *simple\_expression***addop** *term* | T -> U | V U | T **addop** U |
| *term* ->  *factor*  | *term***mulop** *factor* | U -> C | U **mulop** C |
| *factor* ->  **id**  | **id** ( *expression\_list* )  | **num**  | ( *expression* )  |**not** *factor* | C -> **id** | **id** ( S ) | **num** | ( B ) | **not** C |
| *sign* ->  **+**  | **-** | V -> **+** | **-** |

# Substituted Rules

|  |  |
| --- | --- |
|  | D -> **program id** ( E ) ; F G H |
|  | ~~E ->~~ **~~id~~** ~~| E ,~~ **~~id~~**  E -> **id** E’  E’ -> , **id** E’ | Î |
|  | ~~F -> F~~ **~~var~~** ~~E : I ; | Î~~  F -> F’ | Î  F’ -> **var** E : I ; F’ | Î |
|  | I -> J | **array [ num . . num ] of** J |
|  | J -> **integer** | **real** |
|  | ~~G -> G K ; | Î~~  G -> G’ | Î  G’ -> K ; G’ | Î |
|  | K -> L F H |
|  | L -> **function id** M : J ; | **procedure id** M ; |
|  | M -> ( N ) | Î |
|  | ~~N -> E : I | N ; E : I~~  N -> E : I N’  N’ -> ; E : I N’ | Î |
|  | H -> **begin** O **end** |
|  | O -> P | Î |
|  | ~~P -> A | P ; A~~  P -> A P’  P’ -> ; A P’ | Î |
|  | A -> Q **assignop** B | R | H | **if** B **then** A **else** A | **while** B **do** A |
|  | ~~Q ->~~ **~~id~~** ~~|~~ **~~id~~** ~~[ B ]~~  Q -> **id** Q’  Q’ -> Î | [ B ] |
|  | ~~R ->~~ **~~id~~** ~~|~~ **~~id~~** ~~( S )~~  R -> **id** R’  R’ -> Î | ( S ) |
|  | ~~S -> B | S , B~~  S -> B S’  S’ -> , B S’ | Î |
|  | B -> T | T **relop** T |
|  | ~~T -> U | V U | T~~ **~~addop~~** ~~U~~  T -> U T’ | V U T’  T’ -> **addop** U T’ | Î |
|  | U -> C | U **mulop** C  U -> C U’  U’ -> **mulop** C U’ | Î |
|  | C -> **id** | **id** ( S ) | **num** | ( B ) | **not** C |
|  | V -> **+** | **-** |

# Assignment1

# Removal of Left Factoring and Left Recursion

|  |  |
| --- | --- |
|  | PROGRAM -> **program id** ( IDENTIFIER\_LIST ) ; DECLARATIONS SUBPROGRAM\_DECLARATIONS COMPOUND\_STATEMENT |
|  | ~~IDENTIFIER\_LIST ->~~ **~~id~~** ~~| IDENTIFIER\_LIST ,~~ **~~id~~**  IDENTIFIER\_LIST -> **id** IDENTIFIER\_LIST’  IDENTIFIER\_LIST’ -> , **id** IDENTIFIER\_LIST’ | Î |
|  | ~~DECLARATIONS -> DECLARATIONS~~ **~~var~~** ~~IDENTIFIER\_LIST : TYPE ; | Î~~  DECLARATIONS -> DECLARATIONS’ | Î  DECLARATIONS’ -> **var** IDENTIFIER\_LIST : TYPE ; DECLARATIONS’ | Î |
|  | TYPE -> STANDARD\_TYPE | **array [ num . . num ] of** STANDARD\_TYPE |
|  | STANDARD\_TYPE -> **integer** | **real** |
|  | ~~SUBPROGRAM\_DECLARATIONS -> SUBPROGRAM\_DECLARATIONS SUBPROGRAM\_DECLARATION ; | Î~~  SUBPROGRAM\_DECLARATIONS -> SUBPROGRAM\_DECLARATIONS’ | Î  SUBPROGRAM\_DECLARATIONS’ -> SUBPROGRAM\_DECLARATION ; SUBPROGRAM\_DECLARATIONS’ | Î |
|  | SUBPROGRAM\_DECLARATION -> SUBPROGRAM\_HEAD DECLARATIONS COMPOUND\_STATEMENT |
|  | SUBPROGRAM\_HEAD -> **function id** ARGUMENTS : STANDARD\_TYPE ; | **procedure id** ARGUMENTS ; |
|  | ARGUMENTS -> ( PARAMETER\_LIST ) | Î |
|  | ~~PARAMETER\_LIST -> IDENTIFIER\_LIST : TYPE | PARAMETER\_LIST ; IDENTIFIER\_LIST : TYPE~~  PARAMETER\_LIST -> IDENTIFIER\_LIST : TYPE PARAMETER\_LIST’  PARAMETER\_LIST’ -> ; IDENTIFIER\_LIST : TYPE PARAMETER\_LIST’ | Î |
|  | COMPOUND\_STATEMENT -> **begin** OPTIONAL\_STATEMENTS **end** |
|  | OPTIONAL\_STATEMENTS -> STATEMENT\_LIST | Î |
|  | ~~STATEMENT\_LIST -> STATEMENT | STATEMENT\_LIST ; STATEMENT~~  STATEMENT\_LIST -> STATEMENT STATEMENT\_LIST’  STATEMENT\_LIST’ -> ; STATEMENT STATEMENT\_LIST’ | Î |
|  | STATEMENT -> VARIABLE **assignop** EXPRESSION | PROCEDURE\_STATEMENT | COMPOUND\_STATEMENT | **if** EXPRESSION **then** STATEMENT **else** STATEMENT | **while** EXPRESSION **do** STATEMENT |
|  | ~~VARIABLE ->~~ **~~id~~** ~~|~~ **~~id~~** ~~[ EXPRESSION ]~~  VARIABLE -> **id** VARIABLE’  VARIABLE’ -> Î | [ EXPRESSION ] |
|  | ~~PROCEDURE\_STATEMENT ->~~ **~~id~~** ~~|~~ **~~id~~** ~~( EXPRESSION\_LIST )~~  PROCEDURE\_STATEMENT -> **id** PROCEDURE\_STATEMENT’  PROCEDURE\_STATEMENT’ -> Î | ( EXPRESSION\_LIST ) |
|  | ~~EXPRESSION\_LIST -> EXPRESSION | EXPRESSION\_LIST , EXPRESSION~~  EXPRESSION\_LIST -> EXPRESSION EXPRESSION\_LIST’  EXPRESSION\_LIST’ -> , EXPRESSION EXPRESSION\_LIST’ | Î |
|  | ~~EXPRESSION -> SIMPLE\_EXPRESSION | SIMPLE\_EXPRESSION~~ **~~relop~~** ~~SIMPLE\_EXPRESSION~~  EXPRESSION -> SIMPLE\_EXPRESSION EXPRESSION’  EXPRESSION’ -> **relop** SIMPLE\_EXPRESSION | Î |
|  | ~~SIMPLE\_EXPRESSION -> TERM | SIGN TERM | SIMPLE\_EXPRESSION~~ **~~addop~~** ~~TERM~~  SIMPLE\_EXPRESSION -> TERM SIMPLE\_EXPRESSION’ | SIGN TERM SIMPLE\_EXPRESSION’  SIMPLE\_EXPRESSION’ -> **addop** TERM SIMPLE\_EXPRESSION’ | Î |
|  | ~~TERM -> FACTOR | TERM~~ **~~mulop~~** ~~FACTOR~~  TERM -> FACTOR TERM’  TERM’ -> **mulop** FACTOR TERM’ | Î |
|  | ~~FACTOR ->~~ **~~id~~** ~~|~~ **~~id~~** ~~( EXPRESSION\_LIST ) |~~ **~~num~~** ~~| ( expression ) |~~ **~~not~~** ~~| FACTOR~~  FACTOR -> **id** FACTOR’ | **num** | (EXPRESSION) | **not** | FACTOR’  FACTOR’ -> ( EXPRESSION\_LIST ) | Î |
|  | SIGN -> **+** | **-** |

# Assignment 2

# First and Follow

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Non-Terminals** | **First** | **Follow** |
|  | PROGRAM | {**program**} | {**$**} |
|  | IDENTIFIER\_LIST | {**id**} | {**)**, **:**} |
|  | IDENTIFIER\_LIST’ | {**,**, Î} | {**)**, **:**} |
|  | DECLARATIONS | {**var**, Î} | {**function**, **procedure**, **begin**} |
|  | DECLARATIONS’ | {**var**, Î} | {**function**, **procedure**, **begin**} |
|  | TYPE | {**integer**, **real**, **array**} | {**;**, **)**} |
|  | STANDARD\_TYPE | {**integer**, **real**} | {**;**, **)**} |
|  | SUBPROGRAM\_DECLARATIONS | {**function**, **procedure**, Î} | {**begin**} |
|  | SUBPROGRAM\_DECLARATIONS’ | {**function**, **procedure**, Î} | {**begin**} |
|  | SUBPROGRAM\_DECLARATION | {**function**, **procedure**} | {**;**} |
|  | SUBPROGRAM\_HEAD | {**function**, **procedure**} | {**var**, **begin**} |
|  | ARGUMENTS | {**(**, Î} | {**:**, **;**} |
|  | PARAMETER\_LIST | {**id**} | {**)**} |
|  | PARAMETER\_LIST’ | {**;**, Î} | {**)**} |
|  | COMPOUND\_STATEMENT | {**begin**} | {**$**, **;**, **else, end**} |
|  | OPTIONAL\_STATEMENTS | {**id**, **begin**, **if**, **while**, Î} | {**end**} |
|  | STATEMENT\_LIST | {**id**, **begin**, **if**, **while**} | {**end**} |
|  | STATEMENT\_LIST’ | {**;,** Î} | {**end**} |
|  | STATEMENT | {**id**, **begin**, **if**, **while**} | {**;**, **else**, **end**} |
|  | VARIABLE | {**id**} | {**assignop**} |
|  | VARIABLE’ | {**[**, Î} | {**assignop**} |
|  | PROCEDURE\_STATEMENT | {**id**} | {**;**, **else**, **end**} |
|  | PROCEDURE\_STATEMENT’ | {**(**, Î} | {**;**, **else**, **end**} |
|  | EXPRESSION\_LIST | {**id**, **num**, **(**, **not**, **mulop**, **addop**, **+**, **-**, Î} | {**)**} |
|  | EXPRESSION\_LIST’ | {**,**, Î} | {**)**} |
|  | EXPRESSION | {**id**, **num**, **(**, **not**, **mulop**, **addop**, **+**, **-**, Î} | {**;**, **else**, **end**, **then**, **do**, **]**, **,**, **)**} |
|  | EXPRESSION’ | {**relop**, Î} | {**;**, **else**, **end**, **then**, **do**, **]**, **,**, **)**} |
|  | SIMPLE\_EXPRESSION | {**id**, **num**, **(**, **not**, **mulop**, **addop**, **+**, **-**, Î} | {**relop**, **;**, **else, end**, **then**, **do**, **]**, **,**, **)**} |
|  | SIMPLE\_EXPRESSION’ | {**addop,** Î} | {**relop**, **;**, **else, end**, **then**, **do**, **]**, **,**, **)**} |
|  | TERM | {**id**, **num**, **(**, **not**, **mulop**, Î} | {**addop**, **relop**, **;**, **else, end**, **then**, **do**, **]**, **,**, **)**} |
|  | TERM’ | {**mulop**, Î} | {**addop**, **relop**, **;**, **else, end**, **then**, **do**, **]**, **,**, **)**} |
|  | FACTOR | {**id**, **num**, **(**, **not**, Î} | {**mulop**, **addop**, **relop**, **;**, **else, end**, **then**, **do**, **]**, **,**, **)**} |
|  | FACTOR’ | {**(**, Î} | {**mulop**, **addop**, **relop**, **;**, **else, end**, **then**, **do**, **]**, **,**, **)**} |
|  | SIGN | {**+**, **-**} | {**id**, **num**, **(**, **not**, **mulop**, **addop**, **relop**, **;**, **else**, **end**, **then**, **do**, **]**, **,**,**)**} |