<program> ->

PLATYPUS { <opt\_statements> }

<opt\_statements> - >

<statements> | e

**----------------------------------------------------------------------------------------------------------------------------**

**Original:**

<statements> ->

<statement> | <statements> <statement>

**Modification:**

<statements> -> <statements> <statement> | <statement>

<statements> -> <statement> <statements’>

<statements’> -> <statement> <statements’> | ε

**----------------------------------------------------------------------------------------------------------------------------**

<statement> ->

<assignment statement>

| <selection statement>

| <iteration statement>

| <input statement>

| <output statement>

<assignment statement> ->

<assignment expression>;

< assignment expression> ->

AVID = <arithmetic expression>

| SVID = <string expression>

<selection statement> ->

IF <pre-condition> (<conditional expression>) THEN { <opt\_statements> }

ELSE { <opt\_statements> } ;

iteration statement> ->

WHILE <pre-condition> (<conditional expression>)

REPEAT { <statements>};

**<**pre-condition> ->

TRUE | FALSE

**----------------------------------------------------------------------------------------------------------------------------**

**Original:**

< input statement> ->

READ (<opt\_variable list>);

| READ(STR\_T);

**Modification:**

< input statement> ->

READ(< input list>);

< input list > ->

<opt\_variable list> | STR\_T

**Original:**

<variable list> ->

<variable identifier> | <variable list>,<variable identifier>

**Modification:**

<opt\_variable list>

<variable list> | ε

<variable list> ->

<variable identifier><variable list’>

<variable list’>->

,<variable identifier><variable list’>

| ε

**----------------------------------------------------------------------------------------------------------------------------**

**Original:**

<output statement> ->

WRITE (<opt\_variable list>);

| WRITE(STR\_T);

**Modification:**

<output statement> ->

WRITE(<outputlist>);

<output list > ->

<opt\_variable list> | STR\_T

<arithmetic expression> - >

<unary arithmetic expression>

| <additive arithmetic expression>

<unary arithmetic expression> ->

- <primary arithmetic expression>

| + <primary arithmetic expression>

**----------------------------------------------------------------------------------------------------------------------------**

**Original:**

<additive arithmetic expression> ->

<additive arithmetic expression> + <multiplicative arithmetic expression>

| <additive arithmetic expression> - <multiplicative arithmetic expression>

| <multiplicative arithmetic expression>

**Modification:**

<additive arithmetic expression> ->

<multiplicative arithmetic expression> <additive arithmetic expression’>

<additive arithmetic expression’>

+<multiplicative arithmetic expression><additive arithmetic expression’>

|-<multiplicative arithmetic expression><additive arithmetic expression’>

| ε

**----------------------------------------------------------------------------------------------------------------------------**

**Original:**

<multiplicative arithmetic expression> ->

<multiplicative arithmetic expression> \* <primary arithmetic expression>

| <multiplicative arithmetic expression> / <primary arithmetic expression>

| <primary arithmetic expression>

**Modification:**

<multiplicative arithmetic expression> ->

<primary arithmetic expression><multiplicative arithmetic expression’>

<multiplicative arithmetic expression’> ->

\*<primary arithmetic expression><multiplicative arithmetic expression’>

|\<primary arithmetic expression><multiplicative arithmetic expression’>

| ε

**----------------------------------------------------------------------------------------------------------------------------**

<primary arithmetic expression> ->

AVID\_T

| FPL\_T

| INL\_T

| (<arithmetic expression>)

<string expression> ->

<primary string expression>

| <string expression> << <primary string expression>

**Modification:**

<string expression> ->

<primary string expression><string expression’>

<string expression”> ->

<< <primary string expression><string expression’>

| ε

**----------------------------------------------------------------------------------------------------------------------------**

<primary string expression> ->

SVID\_T

| STR\_T

<conditional expression> ->

<logical OR expression>

**Original:**

<logical OR expression> ->

<logical AND expression>

| <logical OR expression> .OR. <logical AND expression>

**Modification:**

<logical OR expression> ->

<logical AND expression><logical OR expression’>

<logical OR expression’> ->

.OR.<logical AND expression><logical OR expression’>

| ε

**----------------------------------------------------------------------------------------------------------------------------**

**Original:**

<logical AND expression> ->

<relational expression>

| <logical AND expression> .AND. <relational expression>

**Modification:**

<logical AND expression> ->

<relational expression><logical AND expression’>

<logical and expression’> ->

.AND.<relational expression><logical AND expression’>

| ε

**----------------------------------------------------------------------------------------------------------------------------**

<relational expression> ->

<primary a\_relational expression> == <primary a\_relational expression>

| <primary a\_relational expression> <> <primary a\_relational expression>

| <primary a\_relational expression> > <primary a\_relational expression>

| <primary a\_relational expression> < <primary a\_relational expression>

| <primary s\_relational expression> == <primary s\_relational expression>

| <primary s\_relational expression> <> <primary s\_relational expression>

| <primary s\_relational expression> > <primary s\_relational expression>

| <primary s\_relational expression> < <primary s\_relational expression>