**Primeros**

P(<program>)= { var,<id> , *ε*}

P(<def-list>)= {var, <id>, *ε* }

P(<*def-list-P*>) ={var, <id>, *ε*}

P(<def>) = { var,<id>}

P(<var-def>) = {var}

P(<var-list>) = {<id>}

P(<id-list>) = {<id>}

P(<id-list-cont>) = {, *ε* }

P(<fun-def>)= {<id>}

P(<param-list>)= { *ε* ,<id>}

P(<var-def-list>)= { *ε,* var}

P(<var-def-list-P>)={ *ε*, var}

P(<stmt-list>)={ *ε* ,<id>,inc, dec, if, while, do, break, return, ;}

P(<stmt-list-P)={ *ε* ,<id>,inc, dec, if, while, do, break, return, ;}

P(<stmt>)={ *ε* ,<id>,inc, dec, if, while, do, break, return, ;}

P(<stmt-assign>)= {<id>}

P(<stmt-incr>)= {inc}

P(<stmt-decr>)={dec}

P(<stmt-fun-call>)={<id>}

P(<fun-call>)={<id>}

P(<expr-list>)={ *ε*, +, -, not, <id>,[ , (, <lit-bool>, <lit-int>, <lit-char>, <lit-str>}

P(<expr-list-cont>)={, , *ε* }

P(<stmt-if>) = {if}

P(<else-if-list>)={elif}

P(<else-if-list-P>)={elif}

P(<else>)={else, *ε* }

P(<stmt-while>)={while}

P(<stmt-do-while>)={do}

P(<stmt-break>)={break}

P(<stmt-return>)={return}

P(<stmt-empty>)={;}

P(<expr>)={+, -, not, <id>,[ , (, <lit-bool>, <lit-int>, <lit-char>, <lit-str>}

P(<expr-or>)={+, -, not, <id>,[ , (, <lit-bool>, <lit-int>, <lit-char>, <lit-str>}

P(<expr-or-P>)={or, *ε* }

P(<expr-and>)={+, -, not, <id>,[ , (, <lit-bool>, <lit-int>, <lit-char>, <lit-str>}

P(<expr-and-P>)={and, *ε* }

P(<expr-comp>)={+, -, not, <id>,[ , (, <lit-bool>, <lit-int>, <lit-char>, <lit-str>}

P(<expr-comp-P>)={==, <>, *ε* }

P(<op-comp>)={==,<>}

P(<expr-rel>) = {+, -, not, <id>,[ , (, <lit-bool>, <lit-int>, <lit-char>, <lit-str>}

P(<expr-rel-P>)={ *ε,* <, <=, >, >=)

P(<op-rel>)={<, <=, >, >=}

P(<expr-add>)={+, -, not, <id>,[ , (, <lit-bool>, <lit-int>, <lit-char>, <lit-str>}

P(<expr-add-P>)={+, -, *ε* }

P(<op-add>)={+,-}

P(<expr-mul>)={+, -, not, <id>,[ , (, <lit-bool>, <lit-int>, <lit-char>, <lit-str>}

P(<expr-mul-P>)={\*, /, %, *ε* }

P(<op-mul>)={\*, /, %}

P(<expr-unary>)={+, -, not, <id>,[ , (, <lit-bool>, <lit-int>, <lit-char>, <lit-str>}

P(<op-unary>)={+, -, not}

P(<expr-primary>)={<id>,[ , (, <lit-bool>, <lit-int>, <lit-char>, <lit-str>}

P(<array>)={ [ }

P(<lit>)={<lit-bool>, <lit-int>, <lit-char>, <lit-str>}