EJERCICIO

PROG := SENT L\_SENTS fin;

L\_SENTS := SENT L\_SENTS| λ;

SENT := num\_inst INST;

INST := ASIG|IN|OUT|SALT\_C|SALT\_I;

-IN := entra N\_MEM;

-OUT := salida N\_MEM;

N\_MEM := dir\_mem REST\_NUM;

REST\_NUM := coma N\_MEM | λ;

-SALT\_I := goto num\_inst;

-SALT\_C := if EXP\_LOG num\_inst;

-ASIG := dir\_mem sim\_asig EXP\_ARIT;

EXP\_ARIT := OPER RESTO;

RESTO := op\_arit OPER| λ;

EXP\_LOG := OPER op\_log OPER; /\* SALT\_C := if EXP\_LOG num\_inst; \*/?

OPER := con\_mem | const\_int; /\* EXP\_LOG := OPER op\_log OPER;

EXP\_ARIT := OPER RESTO;

RESTO := op\_arit OPER| λ;

ANULABLES

PROG -> false;

L\_SENTS -> true;

SENT -> false;

INST -> false;

IN -> false;

OUT -> false;

N\_MEM -> false;

REST\_NUM -> true;

SALT\_I -> false;

SALT\_C->false;

ASIG-> false;

EXP\_ARIT-> false;

RESTO-> true;

EXP\_LOG->false;

OPER-> false;

INICIALES

PROG = {num\_inst};

L\_SENTS = {num\_inst};

SENT = {num\_inst};

INST = {dir\_mem, entra, salida, if, goto};

IN = {entra};

OUT = {salida};

N\_MEM = {dir\_mem};

REST\_NUM ={coma};

SALT\_I = {goto};

SALT\_C = {if};

ASIG = {dir\_mem};

EXP\_ARIT = {con\_mem, const\_int};

RESTO = {op\_arit};

EXP\_LOG = {con\_mem, const\_int};

OPER ={ con\_mem, const\_int};

Seguidores

PROG = {$};

L\_SENTS = {fin};

SENT = {num\_inst, fin};

INST = {num\_inst, fin};

IN = {num\_inst, fin};

OUT = {num\_inst, fin};

N\_MEM = {num\_inst, fin};

REST\_NUM ={num\_inst, fin};

SALT\_I = {num\_inst, fin};

SALT\_C = {num\_inst, fin};

ASIG = {num\_inst, fin};

EXP\_ARIT = {num\_inst, fin};

RESTO = {num\_inst, fin};

EXP\_LOG = {num\_inst};

OPER = {op\_arit, op\_log,num\_inst,fin};