

PROGRAMA -> DECL PROGRAMA | PROCEDIMENT PROGRAMA | MAIN.

DECL -> DECL\_TUPLA ';' | DECL\_TIPUS ';' .

MODIF -> MODIF\_TIPUS ';' .

SENT -> SENTENCIES SENT | .

SENTENCIES -> DECL | MODIF | MENTRE | IF | SWITCH | FMENTRE | CRIDA\_PROC | FOR | OP\_CONDICIONAL.

PROCEDIMENT -> metode ID '(' ARGS ')' '{' SENT '}' | metode TIPUS ID '(' ARGS ')' '{' SENT RETURN '}' .

RETURN -> retorna E.

CRIDA\_PROC -> R.

MAIN -> metode main '(' cad arguments '[' ']' ')' '{' SENT '}' .

ARGS -> ARGS\_1 | .

ARGS\_1 -> ARG ',' ARGS\_1 | ARG.

ARG -> TIPUS ID.

MENTRE -> mentre '(' E ')' fer '{' SENT '}' .

IF -> si '(' E ')' '{' SENT '}' IF\_1.

IF\_1 -> sino si '(' E ')' '{' SENT '}' IF\_1 | sino '{' SENT '}' | .

FMENTRE -> fer '(' SENT ')' mentre '(' E ')' ';' .

FOR -> per '(' TIPUS A ';' FOR\_1 ';' FOR\_1 ')' '{' SENT '}' | per '(' ';' FOR\_1 ';' FOR\_1 ')' '{' SENT '}' .

FOR\_1 -> E | .

OP\_CONDICIONAL -> '{' E '}' '?' OP\_CONDICIONAL\_1 ':' OP\_CONDICIONAL\_1.

OP\_CONDICIONAL\_1 -> OP\_CONDICIONAL | E.

SWITCH -> seleccio '(' E ')' '{' SWITCH\_1 DEF '}' .

SWITCH\_1 -> cas E ':' SENT acaba ';' SWITCH\_1 | .

DEF -> pdefecte ':' SENT acaba ';' .

ENTRADAS -> entradaS '(' TIPUS ',' R ')' .

SORTIDAS -> sortidaS '(' E ')' .

DECL\_CONST -> const TIPUS A '=' E.

DECL\_TIPUS -> TIPUS A '=' E | TIPUS A '=' MODIF\_TIPUS | TIPUS A .

MODIF\_TIPUS -> R I\_1 | R '=' MODIF\_TIPUS.

TIPUS -> ent | logic | decimal | cadena | car .

A -> ID ',' A | ID.

I\_1 -> '++' | '--' | '+=' E | '-=' E | '=' E .

R -> ID | ID '(' VALORS')' | ID '.' R.

VALORS -> E ',' VALORS | E.

VALS\_TUPLA -> E ',' VALS\_TUPLA | E ':' ID ',' VALS\_TUPLA | E | E ':' ID.

DECL\_TUPLA -> tupla ID '(' ')' '=' '(' VALS\_TUPLA ')' | tupla ID '(' ')' '=' R.

E -> T E\_1.

E\_1 -> OR T E\_1 | AND T E\_1 | .

T -> N T\_1.

T\_1 -> '=' N T\_1 | .

N -> M N\_1.

N\_1 -> '+' M N\_1 | .

M -> F M\_1.

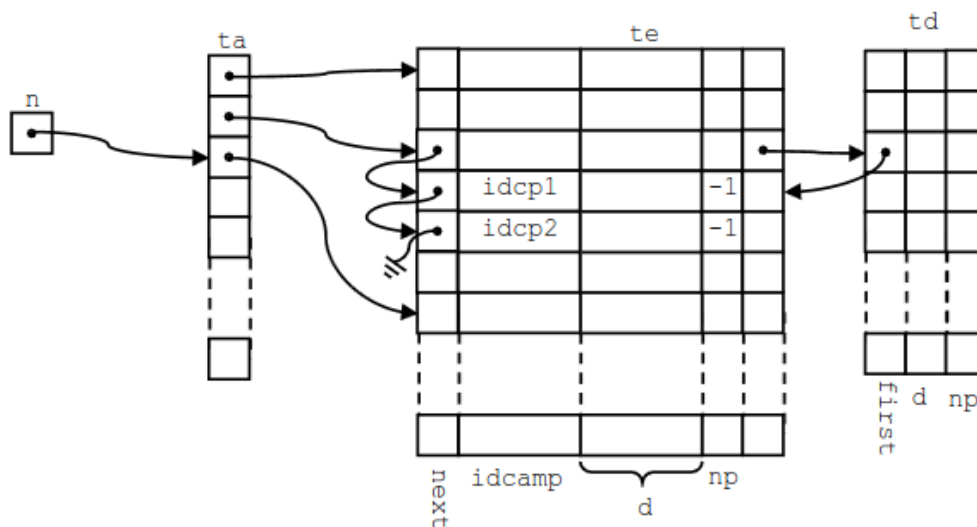
M\_1 -> '\*' F M\_1 | .

F -> '(' E ')' | '-' '(' E ')' | venter | vdecimal | vcharacter | '-' venter | '+' venter | '-' vdecimal | '+' vdecimal | '-' vcharacter | '+' vcharacter | vcadena | boolea | R | '-' R | '+' R.

#### Taula de Simbols

valor	atribut	tipus	descripció
<b>dnul·la</b>	-		no té atribut
<b>dvar</b>	nv	enter	identificador únic
	tipus	string	identificador del tipus de la variable
<b>dconst</b>	valor	...	el valor depèn del tipus
	tipus	string	identificador del tipus de la variable
<b>dtipus</b>	dt	definició_tipus	descripció del tipus
<b>dproc</b>	np	enter	identificador únic
<b>dcamp</b>	tipus	string	identificador del tipus del camp
	despl	enter	desplaçament relatiu respecte de l'inici de la tupla
<b>dindex</b>	tipus	string	el tipus subrang <sup>2</sup>
<b>darg_in</b>	nv	enter	identificador únic
	tipus	string	identificador del tipus del paràmetre
<b>darg</b>	tipus	string	identificador del tipus del paràmetre
	mode	tipus_acces	segons l'accés, pot tenir dos valors: mode in, mode in out
	nom	string	nom de l'argument
...	...	...	...

tipus subjacent	atribut	tipus	descripció
ts_boolea ts_caracter ts_enter	limit_inf limit_sup	enter	Representació binària d'un tipus escalar. Es pot plantejar com un enter amb signe d'una certa quantitat de bits
ts_array	tipuselemental	string	El tipus de l'element individual
ts_record ts_nul	-	-	No requereixen més atributs



te: array list -> Dada2(next,iidcamp,descripcio,np)

ta: array list

td: hash table -> Dada1(first,np,descripcio)