

## 1 Declaracions

PROC	→	<b>procedure</b> C_PROC <b>is</b> DECLS <b>begin</b> SENTS <b>end procedure;</b>
DECLS	→ 	DECLS DECL λ
DECL	→     	PROC DECL_VAR DECL_T
DECL_RVARS	→ 	DECL_RVARS DECL_VAR DECL_VAR
DECL_VAR	→ 	L_ID:ID; L_ID: <b>const</b> ID:=EXPR;
DECL_T	→	<b>type</b> ID <b>is</b> DECL_T_CONT
DECL_T_CONT	→     	<b>new</b> ID <b>range</b> IDX..IDX; <b>record</b> DECL_RVARS <b>end record;</b> <b>array</b> (L_ID) <b>of</b> ID;

### 1.1 Decls Auxiliars

C_PROC	→	ID(ARGS)
		ID
ARGS	→	ARGS;ARG
		ARG
ARG	→	L_ID:MODE ID
MODE	→	<b>in</b>
		<b>out</b>
		<b>in out</b>
IDXS	→	IDXS,IDX
		IDX
IDX	→	EXPR
L_ID	→	L_ID,ID
		ID

## 2 Sentencies

SENTS	→	SENTS.CONT
		<b>null;</b>
SENTS.CONT	→	SENT
		SENT
SENT	→	S_ITER
		S_COND
		S_CRIDA
		S_ASSIGN
S_ITER	→	<b>while</b> EXPR <b>loop</b>
		SENTS
		<b>end loop;</b>
S_COND	→	<b>if</b> EXPR <b>then</b>
		SENTS
		<b>end if;</b>
		<b>if</b> EXPR <b>then</b>
		SENTS
		<b>else;</b>
		SENTS
		<b>end if;</b>
S_CRIDA	→	C_PROC;
S_ASSIGN	→	REF:=EXPR;

### 2.1 Sents Auxiliars

REF	→	REF.CAMP
		CAMP
CAMP	→	ID
		ID(IDXS)

### 3 Expressions

EXPR	→	E0   E1   E2
E0	→	E0 <b>and</b> E2
		E2 <b>and</b> E2
E1	→	E1 <b>or</b> E2
		E2 <b>or</b> E2
E2	→	E2 OP_REL E3
		E3
E3	→	E3 OP_B.C E4
		E4
E4	→	E4 OP_B.E E5
		E5
E5	→	OP_N E6
		E6 OP_E E6
		E6
E6	→	REF
		(EXPR)
		LIT
		-LIT
		TUPLA

#### 3.1 Exprs Auxiliars

TUPLA	→	(L_EXPR)
L_EXPR		L_EXPR, EXPR
		EXPR
OP_N	→	<b>not</b>
OP_E	→	<b>**</b>
OP_REL		
OP_B.C		
OP_B.E		