

142 /SYMS/ GAGLI

DATA IDENT 3/15/68

SYMDEF EQU -1 CAUSES (OR NOT) EXTRA SYMBOLS IN SYMMAC.
 2PRINT EQU -1 CAUSES DEBUGGING PRINTOUT (OR NOT)
 \$MAXSYM EQU 200D
 MAXRL EQU 150D
 NUM EQU MAXSYM-1
 \$MVECSZ EQU 1+NUM/24D MAX BIT VECTOR SIZE IN WORDS
 \$2MVCSZ EQU 2*MVECSZ
 TEMSTO EQU 10D

TYPE OPD 17500000B,1,1

*STORAGE MACRO FOR TABLES. IF SYMDEF IS TRUE, THEN EXTRA SYMBOLS
 * WILL BE DEFINED. THESE SYMBOLS ARE NOT NEEDED, BUT ARE USEFUL
 * FOR DEBUGGING. IT TAKES A WHILE TO GENERATE THEM, AND THE
 * ARPAS SYMBOL TABLE WILL OVERFLOW IF MAXSYM &/OR MAXRUL IS BIG.

SYMMAC MACRO ;* 5/4/68

3MVCSZ EQU 3*MVECSZ
 LOCAT EQU *

IF SYMDEF

NUM EQU 0
 RPT MAXSYM
 NUM EQU NUM+1
 \$L.(\$NUM) BSS MVECSZ
 \$R.(\$NUM) BSS MVECSZ
 ENDR
 \$S0 EQU *-1
 NUM EQU 0
 RPT MAXSYM
 NUM EQU NUM+1
 \$S.(\$NUM) DATA LOCAT POINTS TO L(NUM)
 LOCAT EQU LOCAT+2MVCSZ
 ENDR

ELSE

\$S0 BSS MAXSYM*2MVCSZ
 EQU *-1
 \$S1 EQU *
 RPT NUM+1
 DATA LOCAT POINTS TO L SETS
 LOCAT EQU LOCAT+2MVCSZ
 ENDR

ENDIF

NUM EQU 0
 RPT TEMSTO
 NUM EQU NUM+1
 \$T.(\$NUM) ZRO
 ENDR

```

LOCAT      EQU      *
           IF      SYMDEF
NUM        EQU      MAXSYM
           RPT     MAXSYM
SEQ.(SNUM) BSS     MVECSZ
SLT.(SNUM) BSS     MVECSZ
SGT.(SNUM) BSS     MVECSZ
NUM        EQU      NUM-1
           ENDR
SEQ0      EQU      *
NUM        EQU      MAXSYM
           RPT     MAXSYM+1
SLOC.(SNUM) DATA  LOCAT   POINTS TO EQ(NUM)
LOCAT     EQU      LOCAT+3MVCSZ
NUM        EQU      NUM-1
           ENDR

           ELSE
SEQ0      BSS     MAXSYM*3MVCSZ
           EQU     *
           RPT     MAXSYM+1
LOCAT     DATA  LOCAT   POINTS TO EQ SETS
           EQU     LOCAT+3MVCSZ
           ENDR
SLOC0     EQU     *-1

           ENDF

           FRGT   LOCAT,3MVCSZ,NUM
           ENDM

```

*NOTE: THE LOCATION OF S1 & EQ0 ARE USED IN LRSETS TO ZERO CORE!!
* BE CAREFUL REARRANGING THIS STORAGE.

*THE STORAGE.
SYMMAC

```

$FFUNC BES    MAXSYM
$GFUNC BES    MAXSYM
$ASAVE ZRO
$BSAVE ZRO
$XSAVE ZRO
$1STSYM      ZRO
$ARROW DATA 3
$FLAG ZRO    0
$$FLAG ZRO   0
$DFLAG ZRO   0
$TTYFLG ZRO  0
$WASTE ZRO   0
$RLNUM ZRO   0
$RLNMS ZRO   0
$SYMMUM ZRO  0

```

```

THE SIZE OF THE ARROW IN THE RULES
USED IN LEX, AND SOON ELSEWHERE
SYMBOL FLAG IN SYNNER, SET -.
&D FLAG IN INPUT ROUTINES, SET +
O/I IS WITH TTY, SET +
WHAT A GARBAGE COLLECTION COULD CLAIM
THE NUMBER OF THE LAST RULE
THE LAST SYMBOL DEFINED VIA HASHING

```

\$SYMNMS ZRO	0	TO SAVE THE ABOVE IN APPEND
\$SYMM1 ZRO		
\$MSYMNM ZRO		
\$AVECM1 ZRO	0	ACT. SIZE OF BIT VEC. - 1
\$SYMP1 ZRO	0	1 + NUMBER OF SYMBOLS.
\$FILEO ZRO	0	THE FILE OPENED FOR OUTPUT.
\$FILEI ZRO	0	& FOR INPUT. BOTH TTY AT FIRST.
\$OFIL ZRO	0	WORKING FILE NUMBERS. SET TO THE
\$IFIL ZRO	0	FILES ABOVE OR TTY.
IF	2PRINT	
PFLAG DATA	-1	
ENDF		

*THE STRING POINTERS

\$OLDLNS BSS	2	
\$STRPTS ZRO		
\$LEXLIN BSS	2	
\$OLDLIN BSS	2	
\$NEWLIN ZRO	0	ALSO USED BY COMPLR'S TABLES.
\$STRPTR DATA	(R)SS-1	
\$STREND DATA	(R)ESS-1	
TYPE	STRMSG	
BRU	SNR5	
STRMSG ASC	'\$STR. STO. EXCEEDED\$/'	

*THE CONTROL TBL.

\$CTL DATA	HASHT, EHASHT, 0	
NUM EQU	MAXRL+MAXSYM	
NUM EQU	NUM*5	
NUM EQU	NUM/3	
NUM EQU	NUM*3	
\$HASHT BSS	NUM	
\$EHASHT ZRO		
SS BSS	4*MAXSYM+4*MAXRL	
ESS ZRO		
\$PTBL BSS	2*MAXRL+2	PTR'S TO THE PDN'S.
\$RULEB BSS	MAXRL+1	THESE WILL POINT TO THE RULES IN SS.
\$RULEE BSS	MAXRL+1	

*THE TABLES FOR THE COMPILER

\$FFCN DATA	0, 3, 3, 1, 1, 1, 3, 1, 0
\$GFCN DATA	0, 1, 1, 2, 1, 2, 1, 1, 0
\$PRTBL DATA	0, P1-1, P2-1, 0, 0, 0, P6-1, P7-1, 0
P1 DATA	-2, P11-1, P11A, -2, P12-1, P12A, -2, P13-1, P13A
P1 DATA	-2, P14-1, P14A, 0
P2 DATA	-2, P21-1, P21A, 0
P6 DATA	-2, P61-1, P61A, -2, P62-1, P62A, -1, P63-1, P63A, 0
P7 DATA	-1, P71-1, P71A, 0
P11 DATA	F, T
P12 DATA	S, T
P13 DATA	LT, T
P14 DATA	B, T

P21	DATA	B,GT
P61	DATA	F,N
P62	DATA	S,N
P63	DATA	N
P71	DATA	S
S	EQU	7
N	EQU	6
B	EQU	5
F	EQU	4
LT	EQU	3
GT	EQU	2
T	EQU	1

*THE ACTIONS.

P11A	EAX*	HBEG	HASH THE TERMINAL SYMBOL
	EAX	3,2	
	LDA	0,2	
	BRM	HASHER	
	LDA	=S	
	BRU	ACT	
P12A	BRU	P11A	
P13A	LDA	=B	THE LT POINTS TO THE BEG. OF NON-T
	BRU	ACT	
P14A	BRU	ACT1	
P21A	EAX*	HBEG	HASH THE NON-T.
	EAX	1,2	
	LDA	0,2	
	EAX	2,2	
	LDB	0,2	
	BRM	HASHNT	A,B POINT TO THE NON-T.
	EAX*	HBEG	
	EAX	1,2	
	STA	0,2	THE SYMNUM
	LDA	=N	
	BRU	ACT	
P61A	LDA	=S	
	BRU	ACT	
P62A	BRU	ACT1	
P63A	EAX*	HBEG	SET UP THE 1ST SYMBOL
	EAX	1,2	
	LDA	0,2	
	STA	1STSYM	
	SKR	STRPTR	DON'T WANT 1ST SYM IN PDN.
	LDA	=F	
	BRU	ACT	
P71A	LDA	RLNUM	FINISHED. PUT POINTERS IN PTBL
	LSH	1	
	CAX		
	LDP	NEWLIN	
	STP	PTBL,2	THEN HASH THE PDN & EXIT.
	MUL	=3	
	LSH	23	
	SUB	=1	

	STA	NEWLIN	
	LDA	STRPTR	
	STA	FLAG	JUST BORROWING THIS LOC.
	MUL	=3	
	LSH	23	
	ADD	=2	
	STA	STRPTR	
	CAB		
	LDA	NEWLIN	
	EAX	CTL	
	BRM	BRS5	
	BRM	BRS6	
	SKE	=0	THE NEW PDN IS HASHED
	SBRM	CERROR	IT WASN'T NEW!
	MIN	FLAG	
	STB*	FLAG	PUT PTR TO HASHT AFTER PDN
	LDA	=3	
	ADM	STRPTR	SET THIS UP FOR APPEND
	RCH	22B	CBX + CLB
	EAX	2,2	
	LDA	1STSYM	
	LSH	12D	
	MRG	RLNUM	
	STA	0,2	PUT FISRT SYM. & RLNUM IN HASHT
	MIN	COMPLR	
	BRR	COMPLR	
ACT	STA*	HBEG	
	IF	2PRINT	
ACT1	SKN	PFLAG	
	BRU	PPRINT	
	BRU	REFLEC	
	ELSF	1	
ACT1	BRU	REFLEC	
	ENDF		
HASHER	ZRO		
	LDX	=-1	
	STX	T1	A FLAG USED BELOW
	CAB		
	SUB	=1	SET UP FOR TERMINAL
	BRU	HSR1	
HASHNT	ZRO		
	LDX	=+1	
	STX	T1	SAME FLAG
	LDX	HASHNT	
	STX	HASHER	
HSR1	EAX	CTL	
	BRM	BRS5	
	BRM	BRS6	
	SKE	=0	
	BRU	HSR2	AN OLD SYMBOL
	MIN	SYMNUM	
	LDA	SYMNUM	
	CBX		
	EAX	2,2	

HSR2	STA	0,2	I.E., IN THE HASH TBL
	MIN	STRPTR	
	STA*	STRPTR	APPENDS THIS SYM TO PDN
	SKN	T1	AHA! THE FLAG.
	BRR	HASHER	
	CAX		
	LDA	=20000000B	
	MRG	S0,2	
	STA	S0,2	
	BRR	HASHER	
CERRX	ZRO		
CERROR	ZRO	CERRX	
	TCO	=152B	
	ETR	=7777B	
	LDB	=10D	AN ERROR (OBVIOUSLY)
	LDX	=1	
	BRS	36D	
	TYPE	DUPMSG	
DUPMSG	BRR	CERRX	GO BACK AS IF IT NEVER HAPPENED (MAY 21)
	ASC	' HAS SAME RHSS/'	
	DATA	1,2,3,4,5,6,7	
\$NCODE	BSS	0	
	DATA	64B, 36B, 34B, 46B, 42B, 56B, 63B	
\$ASCODE	BSS	0	
\$NUMSYM	DATA	NCODE-ASCODE-1	
\$ENDSYM	DATA	ASCODE-NCODE+1	
	END		