

(FILECREATED "25-FEB-75 13:35:47" <DEUTSCH>VLOAD.;27.878

changes to: VFILES

previous date: "24-AUG-74 22:11:23" <DEUTSCH>VLOAD.;25)

(LISPXPRT (QUOTE VLOADCOMS)  
T T)  
(RPAQQ VLOADCOMS ((FNS VLOAD)  
(VARS VFILES MF VML)))  
(DEFINEQ

(VLOAD  
[LAMBDA (TYPES HISTFLAG)  
(CLOAD VFILES TYPES HISTFLAG)  
[MAPC VML (FUNCTION (LAMBDA (X)  
(APPLY (QUOTE RECADVISE)  
X])  
(MINFS 8000)  
(MINFS 4000 1])

)  
(RPAQQ VFILES ~~VPATCH~~  
(VHASH VCTX VEU VCA VCM VIA VFAC VREL VBOOL VQC VDS VPROC VIN VEP VPF  
VCG VQM VTP VX VTL VPP VTS VTG VSS VXL))  
(RPAQQ MF <DEUTSCH>MESSAGE.LISP;1)  
(RPAQQ VML ((SYSOUT AFTER NIL (LIST !VALUE)  
MF)  
(LOADX AFTER NIL (LIST (INFILEP FILE))  
MF)  
(DUMPX AFTER NIL (LIST (INFILEP FILE))  
MF)  
(MAKEFILE AFTER NIL (LIST !VALUE)  
MF)))  
(DECLARE: DONTCOPY  
(FILEMAP (NIL (264 468 (VLOAD 276 . 465))))))  
STOP

✓ ✓

VSEQ  
VOUT  
VSEP

XEROX

XEROX

E A R S

Filename: <DEUTSCH>KEY.STATS;2

Creation Date: WEDNESDAY, 19 MAR 1975 14:13-PDT

Printed by: DEUTSCH

XEROX

XEROX

```
(
(CTXEV# . "Calls on fast deduction")
(CEVT# . "Calls which found the argument in THCLAUSEP")
(CEVNIL# . "Calls which found the negation of the argument")
(CEVX# . "Other calls which found a truth value")
T
(CDISP# . "Calls to update context display")
(CDISP#C . "Total display entries deleted")
(CDISP#S . "Total display entries added")
(CDISP#E . "Display entries which were OK already")
T
(BPCTX# . "Calls to setup up backpaths")
T
(GETX# . "Retrievals from data bases")
(GETX#NIL1 . "Retrievals which found no entries at all for key")
(GETX1# . "Retrievals which did find an entry")
(GETX#DEPTH . "Sum of depths of CTX argument in context tree")
(GETX#LEN . "Sum of number of nodes of reduced tree")
(GETX#E . "Retrievals where leaf of backpath was desired node")
(GETX#S . "Number of backpath links followed for other cases")
T
(PUTX# . "Insertions into data bases")
(PUTX#NIL1 . "Insertions which found no entries")
T
(HCNIL# . "HCONSEs with 2nd arg NIL")
(HCNILNEW# . "Number of cells actually created for 2nd arg NIL")
(HC# . "HCONSEs with 2nd arg not NIL")
(HCNEW# . "Number of chains created for 2nd arg not NIL")
(HCADD# . "Number of additions to existing chains")
(HNL# . "Calls to unique-ize non-lists (big numbers, strings)")
(HNLNEW# . "Number of different non-lists")
(HCOP# . "Number of calls to HCOPI")
(HCOPNL# . "HCOPI calls for non-lists")
(HCOPNIL# . "HCOPI calls for lists with CDR=NIL, already HCONSEd")
(HCOPN# . "HCOPI calls for lists with CDR#NIL, already HCONSEd")
T
(FASTPLUS# . "Simple additions")
(SLOWPLUS# . "Complicated additions")
(CANCEL# . "Cancellations of like terms")
(COMBINE# . "Combinations of like terms")
(FASTTIMES# . "Simple multiplications")
(SLOWTIMES# . "Complicated multiplications")
)
```

#STATS							
FASTPLUS#	181	SLOWPLUS#	224	CANCEL#	15	COMBINE#	50
FASTTIMES#	81	SLOWTIMES#	10	FACMUL#	12	LSCMUL#	89
CTXEV#	1155	CEVX#	108	CEVREL#	391	UTEST#	122
CEVRR#	78	LBCONS#	232	TESTRR#	12		
CDISP#	88	CDISP#C	32	CDISP#S	76	CDISP#E	398
BPCTX#	6239	BPCTX#C	2333	BPCTX#T	4704	BPCTX#F	1841
BPCTX#CONS	351	BPCTX#S	504	GETX#	9437	GETX#NIL1	3145
GETX#E	8123	GETX#S	0	GETX#NIL2	0	GETX#NILS	0
PUTX#	1585	PUTX#NIL1	292	PUTX#R	119	PUTX#NIL3	112
PUTX#DD	0	PUTX#DA	0	PUTX#NIL2	368	PUTX#UP	1017
PUTX#D	0	PUTX#S	780	/PUTX#	0		
HCNIL#	1592	HCNILNEW#	287	HC#	4147	HCNEW#	969
HCADD#	341	HNL#	0	HNLNEW#	0	HCOP#	5966
HCOPNL#	1761	HCOPNIL#	153	HCOPN#	2098		

##STATS							
CDISP#	1404	CDISP#C	195	CDISP#S	1397	CDISP#E	1455
BPCTX#	32755	BPCTX#C	7966	BPCTX#T	21073	BPCTX#F	6920
BPCTX#CONS	982	BPCTX#S	1202	GETX#	46965	GETX#NIL1	19620
GETX#E	34680	GETX#S	0	GETX#NIL2	0	GETX#NILS	0
PUTX#	3906	PUTX#NIL1	1042	PUTX#R	399	PUTX#NIL3	276
PUTX#DD	0	PUTX#DA	0	PUTX#NIL2	1285	PUTX#UP	3751
PUTX#D	0	PUTX#S	3066	/PUTX#	0		
FASTPLUS#	227	SLOWPLUS#	791	CANCEL#	87	COMBINE#	50
FASTTIMES#	436	SLOWTIMES#	70	FACMUL#	60	LSCMUL#	242
CTXEV#	4869	CEVX#	579	CEVREL#	1326	UTEST#	506
CEVRR#	346	LBCONS#	254	TESTRR#	16		
HCNIL#	7136	HCNILNEW#	1001	HC#	16884	HCNEW#	3035
HCADD#	964	HNL#	0	HNLNEW#	0	HCOP#	25646
HCOPNL#	7103	HCOPNIL#	314	HCOPN#	9597		

\$STATS							
FASTPLUS#	173	SLOWPLUS#	364	CANCEL#	30	COMBINE#	23
FASTTIMES#	192	SLOWTIMES#	12	FACMUL#	54	LSCMUL#	124
CTXEV#	2128	CEVX#	328	CEVREL#	865	UTEST#	401
CEVRR#	176	LBCONS#	180	TESTRR#	12		
CDISP#	218	CDISP#C	114	CDISP#S	191	CDISP#E	892
BPCTX#	22514	BPCTX#C	8052	BPCTX#T	30141	BPCTX#F	6484
BPCTX#CONS	690	BPCTX#S	874	GETX#	33451	GETX#NIL1	19370
GETX#E	24929	GETX#S	0	GETX#NIL2	0	GETX#NILS	0
PUTX#	2593	PUTX#NIL1	688	PUTX#R	187	PUTX#NIL3	231
PUTX#DD	0	PUTX#DA	0	PUTX#NIL2	731	PUTX#UP	3524
PUTX#D	0	PUTX#S	2856	/PUTX#	0		
HCNIL#	3383	HCNILNEW#	532	HC#	9084	HCNEW#	1901
HCADD#	703	HNL#	0	HNLNEW#	0	HCOP#	14628
HCOPNL#	3925	HCOPNIL#	154	HCOPN#	5342		

##STATS							
CTXEV#	6469	CEVT#	400	CEVNIL#	421	CEVX#	863
CEVREL#	2185	UTEST#	1175	CEVRR#	210	LBCONS#	522
TESTRR#	50						
CDISP#	282	CDISP#C	111	CDISP#S	275	CDISP#E	1742
BPCTX#	49885	BPCTX#C	16355	BPCTX#T	45274	BPCTX#F	14360
BPCTX#CONS	1544	BPCTX#S	1838	GETX#	68810	GETX#NIL1	43872
GETX1#	58359	GETX#DEPTH	515016	GETX#LEN	261832	GETX#E	44224
GETX#S	0	GETX#NIL2	0	GETX#NILS	0	PUTX#	4602
PUTX#NIL1	777	PUTX#R	225	PUTX#NIL3	514	PUTX#DD	0
PUTX#DA	0	PUTX#NIL2	1131	PUTX#UP	2346	PUTX#D	0
PUTX#S	2295	/PUTX#	0				
HCNIL#	7166	HCNILNEW#	397	HC#	20499	HCNEW#	2027
HCADD#	953	HNL#	0	HNLNEW#	0	HCOP#	32619
HCOPNL#	9761	HCOPNIL#	208	HCOPN#	11061		
FASTPLUS#	289	SLOWPLUS#	1147	CANCEL#	81	COMBINE#	114
FASTTIMES#	559	SLOWTIMES#	139	FACMUL#	208	LSCMUL#	450

#OK



```

#STATS
FASTPLUS# 26 SLOWPLUS# 489 CANCEL# 42 COMBINE# 0
FASTTIMES# 230 SLOWTIMES# 0 FACMUL# 12 LSCMUL# 79
CTXEV# 2602 CEVT# 154 CEVNIL# 191 CEVX# 294
CEVREL# 990 UTEST# 112 CEVRR# 18 LBCONS# 2032
TESTRR# 165
CDISP# 511 CDISP#C 415 CDISP#S 527 CDISP#E 1984
BPCTX# 34919 BPCTX#C 9167 BPCTX#T 47934 BPCTX#F 7242
BPCTX#CONS 1444 BPCTX#S 1962 GETX# 36449 GETX#NIL1 16360
GETX1# 38934 GETX#DEPTH
                234540 GETX#LEN 332354 GETX#E 32070
GETX#S 0 GETX#NIL2 0 GETX#NILS 0 PUTX# 5627
PUTX#NIL1 623 PUTX#R 378 PUTX#NIL3 466 PUTX#DD 0
PUTX#DA 0 PUTX#NIL2 1536 PUTX#UP 5476 PUTX#D 0
PUTX#S 4202 /PUTX# 0
HCNIL# 2982 HCNILNEW# 294 HC# 7911 HCNEW# 1615
HCADD# 679 HNL# 0 HNLNEW# 0 HCOP# 12323
HCOPNL# 3821 HCOPNIL# 41 HCOPN# 4252

```

```
#STATS
FASTPLUS# 385
SLOWPLUS# 774
CANCEL# 205
COMBINE# 25
FASTTIMES# 268
SLOWTIMES# 13
FACMUL# 47
LSCMUL# 444
CDISP# 847
CDISP#C 566
CDISP#S 854
CDISP#E 4503
GETX# 16583
GETX#S 44932
GETX#E 5973
GETX#ES 63778
GETX#NILS 18990
GETX#NIL1 2697
GETX#NIL2 2867
PUTX# 6051
PUTX#S 35011
PUTX#A 1508
PUTX#AS 24179
PUTX#D 332
PUTX#DS 3877
PUTX#NILS 2904
PUTX#NIL1 1020
PUTX#NIL2 950
/PUTX# 0
HCNIL# 4149
HCNILNEW# 441
HC# 11067
HCNEW# 2074
HCADD# 992
HCOPNIL# 176
HCOP# 4543
```

```
#STATS
```

FASTPLUS#	700	SLOWPLUS#	1623	CANCEL#	640	COMBINE#	126
FASTTIMES#	367	SLOWTIMES#	118	FACMUL#	15	LSCMUL#	673
CTXEV#	10055	CEVX#	1132	CEVREL#	2703	UTEST#	1840
CEVRR#	710	LBCONS#	306	TESTRR#	68		
CDISP#	1480	CDISP#C	834	CDISP#S	1609	CDISP#E	9027
GETX#	65963	GETX#S	60300	GETX#E	12566	GETX#ES	65118
GETX#NILS	13801	GETX#NIL1	33045	GETX#NIL2	3416	PUTX#	6671
PUTX#S	20051	PUTX#A	1530	PUTX#AS	9976	PUTX#D	289
PUTX#DS	1517	PUTX#NILS	3162	PUTX#NIL1	1613	PUTX#NIL2	1089
/PUTX#	0						
HCNIL#	10051	HCNILNEW#	793	HC#	26392	HCNEW#	3064
HCADD#	1361	HCOPNIL#	246	HCOP#	13917		

#STATS							
CTXEV#	2245	CEVX#	169	CEVREL#	517	UTEST#	158
CEVRR#	123	LBCONS#	108	TESTRR#	0		
FASTPLUS#	87	SLOWPLUS#	171	CANCEL#	1	COMBINE#	0
FASTTIMES#	194	SLOWTIMES#	2	FACMUL#	66	LSCMUL#	47
CDISP#	545	CDISP#C	51	CDISP#S	534	CDISP#E	331
GETX#	6469	GETX#S	4069	GETX#E	1797	GETX#ES	3002
GETX#NILS	9018	GETX#NIL1	2440	GETX#NIL2	3497	PUTX#	1463
PUTX#S	831	PUTX#A	171	PUTX#AS	952	PUTX#D	56
PUTX#DS	417	PUTX#NILS	1814	PUTX#NIL1	540	PUTX#NIL2	474
/PUTX#	0						
HCNIL#	1895	HCNILNEW#	610	HC#	4812	HCNEW#	2009
HCADD#	596	HCOPNIL#	36	HCOP#	2604		

11-AUG-74 05:26:00,856  
Date: 11 AUG 1974 0526-PDT  
From: GERMANY

\$STATS							
#470334	#13						
FASTPLUS#	78	SLOWPLUS#	1125	CANCEL#	698	COMBINE#	351
FASTTIMES#	38	SLOWTIMES#	30	FACMUL#	6	LSCMUL#	375
CTXEV#	1093	CEVX#	166	CEVREL#	1093	UTEST#	756
CEVRR#	166	LBCONS#	0	TESTRR#	0		
CDISP#	273	CDISP#C	145	CDISP#S	168	CDISP#E	227
GETX#	8543	GETX#S	7231	GETX#E	5579	GETX#ES	44962
GETX#NILS	12513	GETX#NIL1	1252	GETX#NIL2	2130	PUTX#	3642
PUTX#S	23731	PUTX#A	484	PUTX#AS	3828	PUTX#D	510
PUTX#DS	3964	PUTX#NILS	716	PUTX#NIL1	159	PUTX#NIL2	184
/PUTX#	0						
HCNIL#	3501	HCNILNEW#	77	HC#	10638	HCNEW#	588
HCADD#	369	HCOPNIL#	68	HCOP#	3137		

\$STATS ]

-----  
 15-AUG-74 00:24:44,892  
 Date: 15 AUG 1974 0024-PDT  
 From: GERMAN  
 Subject: PIVOT STATS (FROM WI ANALYSIS OF K9)

```
##STATS
#470166      #17
FASTPLUS#  1339  SLOWPLUS#  3434  CANCEL#    1875  COMBINE#   1043
FASTTIMES#  516  SLOWTIMES#  0     FACMUL#    6     LSCMUL#   1790
CTXEV#     2297  CEVX#      937  CEVREL#   2094  UTEST#    2605
CEVRR#     937  LBCONS#    0     TESTRR#    0
CDISP#     85   CDISP#C    39   CDISP#S    101  CDISP#E   133
GETX#     13409  GETX#S    10794  GETX#E    8526  GETX#ES   47480
GETX#NILS  8824  GETX#NIL1  2763  GETX#NIL2  3303  PUTX#     5364
PUTX#S    26592  PUTX#A    672  PUTX#AS   3142  PUTX#D    327
PUTX#DS   1533  PUTX#NILS  590  PUTX#NIL1  451  PUTX#NIL2  337
/PUTX#     0
HCNIL#    14054  HCNILNEW#  245  HC#        41037  HCNEW#    2545
HCADD#    1760  HCOPNIL#   532  HCOP#     14424
```

-----  
 23-AUG-74 22:00:56,871  
 Date: 23 AUG 1974 2200-PDT  
 From: GERMAN  
 Subject: PIVOT STATISTICS

```
#STATS
#403030      #21
FASTPLUS#  1887  SLOWPLUS#  4588  CANCEL#    2989  COMBINE#   1239
FASTTIMES#  200  SLOWTIMES#  0     FACMUL#    6     LSCMUL#   2588
CTXEV#     2981  CEVX#     1091  CEVREL#   2942  UTEST#    2752
CEVRR#     1091  LBCONS#    0     TESTRR#    0
CDISP#     203  CDISP#C    49   CDISP#S    304  CDISP#E   162
GETX#     36156  GETX#S    58329  GETX#E   23263  GETX#ES  149383
GETX#NILS  64722  GETX#NIL1  4410  GETX#NIL2 11397  PUTX#     9182
PUTX#S    74498  PUTX#A    1015  PUTX#AS   10476  PUTX#D   2282
PUTX#DS   22499  PUTX#NILS  2348  PUTX#NIL1  611  PUTX#NIL2  526
/PUTX#     0
HCNIL#    17776  HCNILNEW#  188  HC#        54905  HCNEW#    2442
HCADD#    1739  HCOPNIL#   412  HCOP#     18576
```

```

#STATS
#364031      #0
FASTPLUS#   224
SLOWPLUS#   1159
CANCEL#     132
COMBINE#    97
FASTTIMES#  924
SLOWTIMES#  0
FACMUL#     5
LSCMUL#    460
CDISP#     1492
GETX#      22991
GETX#S     96058
GETX#E     5622
GETX#ES    104625
GETX#NILS  100729
GETX#NIL1  6605
GETX#NIL2  5398
PUTX#      7559
PUTX#S     51803
PUTX#NILS  71895
PUTX#NIL1  717
PUTX#NIL2  4567
HCNIL#     5093
HCNILNEW#  290
HC#        15369
HCNEW#     1513
HCADD#     723
HCOPNIL#   601
HCOP#      7653

```

## #JUSTIFY

```

(2/1) for procedure SIFTUP
(7/2) for paths beginning [101]-303...399
(13/7) assuming TRUE in 400
(40/13) assuming TRUE in 500
(67/40) assuming TRUE in 610
(76/67) assuming TRUE in 700
(92/76) for goal NIL
  Proved
(148/92) for proof
  Proved
(150/148) to prove remaining disjuncts of 7.3
  Proved
(158/150) to prove remaining disjuncts of 150.7
  Proved
(162/158) to prove remaining disjuncts of 158.11
  Proved
(161/158) to prove one disjunct of 158.11
  Proved
(164/161) to prove remaining disjuncts of 161.2
  Proved
(163/161) to prove one disjunct of 161.2
  Proved
(157/150) to prove one disjunct of 150.7
  Proved
(160/157) to prove remaining disjuncts of 157.10
  Proved
(159/157) to prove one disjunct of 157.10
  Proved
(149/148) to prove one disjunct of 7.3
  Proved
(152/149) to prove remaining disjuncts of 149.7
  Proved
(156/152) to prove remaining disjuncts of 152.7
  Proved
(155/152) to prove one disjunct of 152.7
  Proved
(151/149) to prove one disjunct of 149.7
  Proved
(154/151) to prove remaining disjuncts of 151.5
  Proved
(153/151) to prove one disjunct of 151.5
  Proved

```

(77/76) for goal NIL  
Proved  
(147/77) for proof  
Proved  
(68/67) assuming FALSE in 700  
(75/68) for goal NIL  
Proved  
(146/75) for proof  
Proved  
(74/68) for goal NIL  
Proved  
(137/74) for proof  
Proved  
(139/137) to prove remaining disjuncts of 7.3  
Proved  
(138/137) to prove one disjunct of 7.3  
Proved  
(141/138) to prove remaining disjuncts of 138.4  
Proved  
(145/141) to prove remaining disjuncts of 141.7  
Proved  
(144/141) to prove one disjunct of 141.7  
Proved  
(140/138) to prove one disjunct of 138.4  
Proved  
(143/140) to prove remaining disjuncts of 140.5  
Proved  
(142/140) to prove one disjunct of 140.5  
Proved  
(41/40) assuming FALSE in 610  
(50/41) assuming TRUE in 700  
(66/50) for goal NIL  
Proved  
(165/66) for proof  
Proved  
(167/165) to prove remaining disjuncts of 7.3  
Proved  
(11/167) to prove remaining disjuncts of 167.7  
Proved  
(18/11) to prove remaining disjuncts of 11.11  
Proved  
(17/11) to prove one disjunct of 11.11  
Proved  
(20/17) to prove remaining disjuncts of 17.2  
Proved  
(19/17) to prove one disjunct of 17.2  
Proved  
(9/167) to prove one disjunct of 167.7  
Proved  
(16/9) to prove remaining disjuncts of 9.10  
Proved  
(12/9) to prove one disjunct of 9.10  
Proved  
(166/165) to prove one disjunct of 7.3  
Proved  
(169/166) to prove remaining disjuncts of 166.7  
Proved  
(8/169) to prove remaining disjuncts of 169.7  
Proved  
(6/169) to prove one disjunct of 169.7  
Proved  
(168/166) to prove one disjunct of 166.7  
Proved  
(5/168) to prove remaining disjuncts of 168.5  
Proved  
(4/168) to prove one disjunct of 168.5  
Proved  
(51/50) for goal NIL  
Proved  
(118/51) for proof  
Proved

(42/41) assuming FALSE in 700  
  (49/42) for goal NIL  
    Proved  
  (117/49) for proof  
    Proved  
  (48/42) for goal NIL  
    Proved  
  (108/48) for proof  
    Proved  
  (110/108) to prove remaining disjuncts of 7.3  
    Proved  
  (109/108) to prove one disjunct of 7.3  
    Proved  
  (112/109) to prove remaining disjuncts of 109.4  
    Proved  
  (116/112) to prove remaining disjuncts of 112.7  
    Proved  
  (115/112) to prove one disjunct of 112.7  
    Proved  
  (111/109) to prove one disjunct of 109.4  
    Proved  
  (114/111) to prove remaining disjuncts of 111.5  
    Proved  
  (113/111) to prove one disjunct of 111.5  
    Proved  
(14/13) assuming FALSE in 500  
(23/14) assuming TRUE in 700  
(39/23) for goal NIL  
  Proved  
(25/39) for proof  
  Proved  
(27/25) to prove remaining disjuncts of 7.3  
  Proved  
(29/27) to prove remaining disjuncts of 27.7  
  Proved  
(33/29) to prove remaining disjuncts of 29.8  
  Proved  
(32/29) to prove one disjunct of 29.8  
  Proved  
(35/32) to prove remaining disjuncts of 32.2  
  Proved  
(34/32) to prove one disjunct of 32.2  
  Proved  
(28/27) to prove one disjunct of 27.7  
  Proved  
(31/28) to prove remaining disjuncts of 28.7  
  Proved  
(30/28) to prove one disjunct of 28.7  
  Proved  
(26/25) to prove one disjunct of 7.3  
  Proved  
(24/23) for goal NIL  
  Proved  
(97/24) for proof  
  Proved  
(15/14) assuming FALSE in 700  
(22/15) for goal NIL  
  Proved  
(96/22) for proof  
  Proved  
(21/15) for goal NIL  
  Proved  
(93/21) for proof  
  Proved  
(95/93) to prove remaining disjuncts of 7.3  
  Proved  
(94/93) to prove one disjunct of 7.3  
  Proved  
(10/7) assuming FALSE in 400  
(3/2) for paths beginning 1...300-[101]-303...307

(FILECREATED "26-JUL-75 22:27:31" <DEUTSCH>VSU.;166 36594

changes to: VSUVARS RECOMP EDITSINCE DATEGTP DAYNO ESVAL VSUFNS CK  
FIXVARTYPES VSUSET RNFILE1 FILEVARP CHECKNEWVARS MAKENEWFILE ADVISEQ APPLMAPC  
APPLMAPC FIXENDUMP VSUCOMS VSUADVISE ADDTO DEFVARTYPE SYSVARS

previous date: "19-MAR-75 16:30:15" <DEUTSCH>VSU.;163)

(PRETTYCOMPRINT VSUCOMS)

[RPAQQ VSUCOMS ((FNS \* VSUFNS)

(VARS \* VSUVARS)  
(GLOBALVARS \* VSUGLOBALS)  
(BLOCKS \* VSUBLOCKS)  
(ADVISE: \* VSUADVISE)

~~(DECLARE: DOEVAL@COMPILE (PROP MACRO CONSTANT))~~

(ADDVARS (DIRECTORIES)

(PRINTYPELST)

~~(TYPEMANELOT)~~

~~(NEWVARDCLSY)~~

[PRETTYDEFMACROS

[ENDUMP (X)

(E (FIXENDUMP (QUOTE X)

[ADVISE: X (P (APPLMAPC (QUOTE X)

[NEW/FNS X (P (MAPC (QUOTE X)

[GLOBALVARS X (DECLARE: EVAL@COMPILE (ADDVARS

(ADDPROPS X (P \* (ADDPROPFOR (CAR (QUOTE X))

(LISPMACROS (CON (SUBEXEC))

(DET (DETACH))

(QUIT (LOGOUT))

(LISPMACROS (FRPLACA . /RPLACA)

(FRPLACD . /RPLACD))

(BREAKMACROS (EP EVAL VALUE))

(LISPMACROS (DET QUIT EP)

(NILCOMS (VARS \* SYSVARS))

(USERMACROS CK U FIXF/ SUBFN/ADDCOM)

(DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILEVARS

(ADDVARS (NLAMA)

(NLAML LISPXPRTQ LISPXPRTQ])

(RPAQQ VSUFNS

(COUNTFILES COUNTFILE COUNTCOM COUNTPROPS COUNTUP ESVAL DUMPFIX  
COMPROP BLOCKCHECK ASSOCA LCHECK ADDPROPFOR ADDLIST ADDTO  
ENDLOAD DOPRETTYCOMS DOEVAL RPA DOPDEC MAKELOAD NAMEPART  
FNAME DEFSAVEX DEFSAVE DEFTAIL1 DEFGETP /DEFPUT DEFPUT  
EDITED EDITDATE EDITFIX CK VSUSET ADDA DEFVARTYPE MOVFN  
UNFILE MOVPROP MOVDEFS UNDEFINE RNFILE MERGEFILE RNFILE1  
RNFN MAKENEWFILE FNFILE MOVEFRONT CHANGEDFILE UNCHANGE  
FIXENDUMP CHECKNEWVARS FILEVARP DEFINEX CONSTANT  
DWIMUSERFN FPROPSAVES CORECOMPILE SUBEXEC DETACH RECOR  
RECADVISE ERRORTYPE PRINTYPE ADDFORM LISPXPRT1  
LISPXPRTQ LISPXPRTQ NPRINT NPRINT2 NPRINT1 NPRINTX NPRINT  
UPRIN))

(DEFINEQ

(COUNTFILES

[LAMBDA (L)

(PROG [(R (MAPCAR (COND  
((NULL L)  
FILELST)  
(T (LISTP L)))  
(FUNCTION (LAMBDA (X)  
(CONS X (COUNTFILE X])  
(RETURN (CONS (COUNTUP R (FUNCTION CDR))  
R])

(\* edited (10-AUG-74 . 2105))

★ to FIRST

(MACROS X  
(DECLARE: EVAL@COMPILE  
(PROP MACRO . X))

★

(P (ADDTOPROP . X))

★ VSUMACROS =  
(CONSTANT)



```

(COUNTFILE
  [LAMBDA (X)                                     (* edited (8-JAN-75 . 1858))
    (IPLUS (COUNT (FCOMS X))
      (COUNTUP (FCOMS X)
        (FUNCTION COUNTCOM))
      (COUNTPROPS X FILEPROPS])

(COUNTCOM
  [LAMBDA (Y)                                     (* edited (8-JAN-75 . 1816))
    (COND
      ((LISTP Y)
        (SELECTQ (CAR Y)
          [VARS (COUNTUP (DOPEVAL (CDR Y))
            (FUNCTION (LAMBDA (Z)
              (COUNT (COND
                ((LITATOM Z)
                  (CAR Z))
                (T (CAAR Z))
              )
            )
          [FNS (COUNTUP (DOPEVAL (CDR Y))
            (FUNCTION (LAMBDA (Z)
              (COUNT (OR (GETP Z (QUOTE EXPR))
                (GETD Z))
            )
          [(PROP IFPROP COMPROP)
            (PROG [(PL (OR (LISTP (CADR Y))
              (LIST (CADR Y))
            )
              (COUNTUP (DOPEVAL (CDDR Y))
                (FUNCTION (LAMBDA (Z)
                  (COUNTPROPS Z PL])
            ]
          [ADDVARS (COUNTUP (DOPEVAL (CDR Y))
            (FUNCTION (LAMBDA (Z)
              (LENGTH (CDR Z))
            )
          [DECLARE: (PROG ((LOAD T)
            (COPY T))
            (RETURN (COUNTUP
              (CDR Y)
              (FUNCTION (LAMBDA (Z)
                (SELECTQ Z
                  ((EVAL@LOAD DOEVAL@LOAD)
                    (SETQ LOAD T)
                    NIL)
                  (DONTEVAL@LOAD
                    (SETQ LOAD NIL))
                  ((COPY DOCOPY)
                    (SETQ COPY T)
                    NIL)
                  (DONTCOPY (SETQ COPY
                    NIL))
                  (COUNTCOM Z]
            )
          (COMS (COUNTUP (DOPEVAL (CDR Y))
            (FUNCTION COUNTCOM)))
        ([LAMBDA (D)
          (COND
            (D (COUNTUP [COND
              ((AND (LITATOM (CAR D))
                (CAR D))
              (SUBST (DOPEVAL (CDR Y))
                (CAR D)
                (CDR D)))
              (T (SUBPAIR (CAR D)
                (DOPEVAL (CDR Y))
                (CDR D])
              (FUNCTION COUNTCOM]
            )
          (CDR (FASSOC (CAR Y)
            PRETTYDEFMACROS])

(COUNTPROPS
  [LAMBDA (Z PL)                                 (* edited (6-JAN-75 . 1801))
    (COUNTUP PL (FUNCTION (LAMBDA (P)
      (COND
        ((SETQ P (FGETP Z P))
          (IPLUS 2 (COUNT P])
    )

```

```
(COUNTUP
 [LAMBDA (MAPL MAPFN) (* edited (10-AUG-74 . 2110))
 (PROG ((MAPSUM 0))
 [MAPC MAPL (FUNCTION (LAMBDA (MAPX)
 (COND
 ((SETQ MAPX (APPLY* MAPFN MAPX))
 (SETQ MAPSUM (IPLUS MAPSUM MAPX]
 (RETURN MAPSUM])
```

```
(ESVAL
 [LAMBDA (ESVALX) (* edited (26-JUL-75 . 1847))
 (COND
 ((NEQ (GETTOPVAL ESVALX)
 (QUOTE NOBIND))
 (GETTOPVAL ESVALX])
```

```
(DUMPFIX
 [LAMBDA (FILE) (* edited NIL)
 (PROG [(D (FNAME FILE (QUOTE DUMP]
 (COND
 ((GETD D)
 (EVAL (LISPXPRINT (LIST D)
 T])
```

```
(COMPROP
 [LAMBDA (PROP L ONLY) (* edited (6-JAN-75 . 1805))
 (SUBST (CONS PROP L)
 (QUOTE X)
 (COND
 [ONLY (QUOTE (DECLARE: DOEVAL@COMPILE DONTCOPY (PROP . X]
 (T (QUOTE (DECLARE: DOEVAL@COMPILE (PROP . X])
```

```
(BLOCKCHECK
 [LAMBDA (FILE) (* edited (5-AUG-74 . 1614))
 (PROG ((BLOCKS (FBLOCKS FILE))
 (FNS (FFNS FILE))
 NAMES ENTRIES)
 (OR BLOCKS (RETURN))
 [SETQ NAMES (MAPCONC BLOCKS (FUNCTION (LAMBDA (X)
 (AND (CAR X)
 (LIST (CAR X]
 [SETQ ENTRIES
 (MAPCONC BLOCKS
 (FUNCTION (LAMBDA (X)
 (AND (CAR X)
 (OR (APPEND (CDR (ASSOCA (QUOTE ENTRIES)
 X)))
 (LIST (CAR X]
 [MAPC
 BLOCKS
 (FUNCTION (LAMBDA (X)
 (COND
 ([OR (NLISTP X)
 (NOT (LITATOM (CAR X]
 (LISPXPRINT (APPEND (QUOTE (Bad BLOCKS entry))
 (LIST X))
 T)
 (SETQ X NIL))
 ((AND (FMEMB (CAR X)
 (CDR X))
 (OR (ASSOCA (QUOTE ENTRIES)
 X)
 (ASSOCA (QUOTE BLKAPPLYFNS)
 X)))
 (LISPXPRINT (CONS (CAR X)
 (QUOTE (- must use different block name)))
 T)))
 (MAPC
 (CDR X)
```



```
(ASSOCA
 [LAMBDA (X L)
 (CAR (SOME L (FUNCTION (LAMBDA (Y)
 (AND (LISTP Y)
 (EQ (CAR Y)
 X]))
 (* edited NIL)
```

```
(LCHECK
 [LAMBDA (X A)
 (OR (LISTP (CDR A))
 (LISPXPRT (LIST (QUOTE BAD:)
 A)
 T))
 (MAPC (CDR A)
 (FUNCTION (LAMBDA (Y)
 (COND
 ((NOT (FMEMB Y (CDR X)))
 (LISPXPRT (CONS Y (CONS (LIST (QUOTE on)
 (CAR A))
 (APPEND (QUOTE (is not in))
 (LIST (CAR X))
```

```
(ADDPROPFORM
 [LAMBDA (PROP L)
 (LIST (LIST (QUOTE ADDLIST)
 (KWOTE L)
 (KWOTE PROP]))
 (* edited (3-JUN-74 . 59))
```

(ADDTOPROP  
[X X  
(ADDLIST (CDR X)  
(CAR X])

```
(ADDLIST
 [LAMBDA (L PROP)
 [MAPC L (FUNCTION (LAMBDA (Y)
 (/PUT (CAR Y)
 PROP
 (APPEND (CDR Y)
 (GETP (CAR Y)
 PROP]
 (* edited (3-JUN-74 . 56))
```

```
(ADDTO
 [LAMBDA (X L FLG)
 (PROG [(V (LISTP (GETTOPVAL X)
 (RETURN (SAVESET X
 (COND
 [FLG (/NCONC V (SUBSET L
 (FUNCTION (LAMBDA (Y)
 (NOT (MEMBER Y V))
 (T (UNION L V)))
 T])
 (* edited (26-JUL-75 . 2155))
```

```
(ENDLOAD
 [LAMBDA (FILE ALL)
 (DOPRETTYCOMS (F' FILE)
 (* edited (17-JUL-74 . 1341))
```

(MAPPRETTYCOMS (FCOMS FILE) (F' DOPRETTYCOM))

(MAPPRETTYCOMS  
[X (MAPX MAPFN1 MAPFN2 MAPZ)  
(OR MAPFN2 (SETQ MAPFN2 (F' NIL)))  
(MAPCOMSI MAPX MAPZ])

(DOPRETTYCOM) (C R)  
[LAMBDA (L ALL)]

(\* edited (9-JUL-74 . 1427))

~~(MAPC  
L  
(FUNCTION (LAMBDA (X)  
(COND  
((LISTP X)  
(PROG ((R (CDR X)))  
(SELECTQ  
(CAR X)  
[(P DO E)  
(AND ALL (MAPC (DOPEVAL R)  
(FUNCTION EVAL]  
[VARS (MAPC (DOPEVAL R)  
(FUNCTION (LAMBDA (X)  
(COND  
((LISTP X)  
(RPA (CAR X)  
(EVAL (CADR X])  
(COMS (DOPRETTYCOMS) (DOPEVAL R)  
ALL))  
[ADDDVARS (MAPC (DOPEVAL R)  
(FUNCTION (LAMBDA (X)  
(RPA (CAR X)  
(UNION (CDR X)  
(LISTP (CAAR X])  
(DECLARE: (DOPRETTYCOMS (DOPDEC R T)  
ALL))  
(COND  
((SETQ X (FASOC (CAR X)  
PRETTYMACROS))  
(DOPRETTYCOMS (SUBPAIR (CADR X)  
R  
(CDDR X))  
ALL.]~~

(SELECTQ  
(CAR X)  
[(P DO E)  
(AND ALL (MAPC (DOPEVAL R)  
(FUNCTION EVAL]  
[VARS (MAPC (DOPEVAL R)  
(FUNCTION (LAMBDA (X)  
(COND  
((LISTP X)  
(RPA (CAR X)  
(EVAL (CADR X])  
(COMS (DOPRETTYCOMS) (DOPEVAL R)  
ALL))  
[ADDDVARS (MAPC (DOPEVAL R)  
(FUNCTION (LAMBDA (X)  
(RPA (CAR X)  
(UNION (CDR X)  
(LISTP (CAAR X])

(ADDDO (CAR X) (CDR X))

(DECLARE: (DOPRETTYCOMS (DOPDEC R T)  
ALL))  
(COND  
((SETQ X (FASOC (CAR X)  
PRETTYMACROS))  
(DOPRETTYCOMS (SUBPAIR (CADR X)  
R  
(CDDR X))  
ALL.]

NIL

(DOPEVAL  
[LAMBDA (L)  
(COND  
((AND (LISTP L)  
(EQ (CAR L)  
(QUOTE \*)))  
(EVAL (CADR L)))  
(T L])

(\* edited (9-AUG-74 . 2255))

to FIRST

(RPA  
[LAMBDA (X Y)  
(SAVESET X Y T])

(\* edited NIL)

(DOPDEC  
[LAMBDA (L FLAG)  
(SUBSET L (FUNCTION (LAMBDA (X)  
(SELECTQ X  
((DOEVAL@LOAD EVAL@LOAD)  
(SETQ FLAG T)  
NIL)  
(DONT@EVAL@LOAD (SETQ FLAG NIL))  
(AND FLAG (LISTP X]))

(\* edited (9-JUL-74 . 1426))

(MAPCOMSI

(L VAL FLAG)

(COND ((NULL L) VAL)

((NLISTP (CAR L)) (APPLY\* MAPFN2

(APPLY\* MAPFN1 'VARS (FRPLACA '(NIL) (CAR L))  
(CAR L))

(MAPCOMSI (CDR L) VAL))

(T (SELECTR (CAAR L) [COMS

(MAPCOMSI (DOPEVAL (CDAR L)) (MAPCOMSI (CDR L) VAL]

[DECLARE: (MAPCOMSI (CDAR L)

((L (D) (COND (D (MAPCOMSI (SUBPAIR

T]

(MAKELOAD  
[LAMBDA (GROUP LIS) (\* edited (7-AUG-74 . 846))

(PROG [(FILE (FNAME GROUP (QUOTE LOAD)))  
(FILES (FNAME GROUP (QUOTE FILES)  
(SETFCOMS FILE (LIST (LIST (QUOTE FNS)  
FILE)  
(LIST (QUOTE VARS)  
FILES)))]

(SET FILES LIS)  
[DEFINEX  
(LIST (CONS FILE

(CONS (QUOTE (TYPES HISTFLAG))  
(SUBST FILES (QUOTE FILES)  
(QUOTE ((COND  
(NOT (FMEMB (QUOTE VFU)  
FILELST))  
(LAMBDA (LISPXHIST)  
(LOAD (QUOTE VFU)  
NIL)))  
(LOAD FILES TYPES HISTFLAG]

'(LOADFIRSTFILES)  
<'LOAD FILES !' (TYPES  
HISTFLAG) >

(RETURN (MAKENEWFILE FILE])

(NAMEPART  
[LAMBDA (FILE SUFFIXFLG) (\* edited NIL)

(PROG [(P (GETP FILE (QUOTE NAMEFIELD))  
[COND  
(NULL P)  
(PUT FILE (QUOTE NAMEFIELD)  
(SETQ P (CONS (NAMEFIELD FILE NIL)  
(NAMEFIELD FILE T])  
(RETURN (COND  
(SUFFIXFLG (CDR P))  
(T (CAR P])

EDITFILE

(FNAME  
[LAMBDA (FILE VAR) (\* edited NIL)

([LAMBDA (W)  
(FRPLACA (CDR W)  
VAR)  
(PACK (FRPLACA W (NAMEPART FILE])  
(QUOTE (VSU FNS])

(DEFSAVEX  
[LAMBDA (L) (\* edited (3-JUL-74 . 1137))

(AND (NOT (STKPOS (QUOTE DEFINEQ)))  
(NOT (STKPOS (QUOTE LOAD)))  
(NOT (STKPOS (QUOTE LOADFNS)))  
(DEFSAVE L])

(DEFSAVE  
[LAMBDA (L) (\* edited (9-JUL-74 . 1427))

(MAPC L (FUNCTION EDITED))  
(MAPC L (FUNCTION (LAMBDA (X)  
(PROG ((Y (CK X)))  
(COND  
(Y (LISXPXPRIN1 (QUOTE \*\*\*)  
T)  
(LISXPXPRIN2 (CONS X Y)  
T)  
(LISXPXPRINT (QUOTE \*\*\*)  
T])

(DEFTAIL1  
[LAMBDA (FN) (\* edited (6-JAN-75 . 1807))

(LISTP (CDR (LISTP (CDR (LISTP (VIRGINFN FN NIL]))

(DEFGETP  
[LAMBDA (FN PROP) (\* edited (6-JAN-75 . 1807))

(PROG ((D (DEFTAIL1 FN)))  
(RETURN (AND D (EQ [CAR (SETQ D (LISTP (CAR D)  
(QUOTE \*])  
(CADR (FMEMB PROP (CDR D])

```

(/DEFPUT
[LAMBDA (FN PROP VAL)                                (* edited (6-JAN-75 . 1807))
 (DEFPUT FN PROP VAL T])

(DEFPUT
[LAMBDA (FN PROP VAL UNDOFLG)                        (* edited (8-JAN-75 . 1826))
 (PROG ((D (DEFTAIL1 FN))
        D1)
        [COND
          ((NULL D)
           (ERROR FN (QUOTE "HAS NO EXPR IN CORE")))
          TOP (COND
                [(AND (EQ [CAR (SETQ D1 (LISTP (CAR D)
              (QUOTE *))
              (LISTP (CDR D1))
              (EQ (CADR D1)
              (QUOTE edited)))
                (COND
                 [VAL (COND
                       (UNDOFLG (/PUTL (CDR D1)
                                       PROP VAL))
                       (T (PUTL (CDR D1)
                               PROP VAL]
                 [(EQ PROP (QUOTE edited))
                  (COND
                   [(CDDDR D1)
                    (COND
                     (UNDOFLG (/RPLACA (CDDR D1)
                                       NIL))
                     (T (RPLACA (CDDR D1)
                               NIL]
                   (T (COND
                       (UNDOFLG (/RPLNODE2 D (CDR D)))
                       (T (RPLNODE2 D (CDR D]
                    (T (PROG ((L (CDDR D1)))
                        LP (COND
                            ((NULL (CDR L))
                             (RETURN))
                            [(EQ (CADR L)
                                PROP)
                             (RETURN (COND
                                     (UNDOFLG (/RPLACD L (CDDDR L)))
                                     (T (RPLACD L (CDDDR L]
                            (T (SETQ L (CDR L))
                                (GO LP]
                    (VAL (SETQ D1 (LIST (QUOTE *)
                                      (QUOTE edited)
                                      NIL))
                  (COND
                   (UNDOFLG (/ATTACH D1 D))
                   (T (ATTACH D1 D)))
                  (GO TOP)))
                (RETURN VAL])

(EDITED
[LAMBDA (X)                                           (* edited (6-JAN-75 . 1811))
 [AND (LITATOM X)
      X
      (/DEFPUT X (QUOTE edited)
                (CONS (MKATOM (DATE 67371008))
                     (MKATOM (DATE -34317533184]
      X])

(EDITDATE
[LAMBDA (X)                                           (* edited (6-JAN-75 . 1812))
 (DEFGETP X (QUOTE edited])

```

```
(EDITFIX
[LAMBDA (FILE) (* edited (6-JAN-75 . 1813))
(MAPC (FFNS FILE)
(FUNCTION (LAMBDA (FN)
(COND
((NULL (EDITDATE FN))
(DEFPUT FN (QUOTE edited)
(GETP FN (QUOTE EDITDATE]))
```

```
(CK
[LAMBDA (FN) (* edited (26-JUL-75 . 1850))
(AND
(FGETD (QUOTE CALLS))
(PROG ((Z (CALLS FN))
Y)
(RETURN (NCONC [AND [SETQ Y (SUBSET (CAR Z)
(FUNCTION (LAMBDA (X)
(NOT (FGETD X])
(LIST (CONS (QUOTE calls:)
(SORT Y (FUNCTION ALPHORDER]
(AND (SETQ Y (SUBSET (CADDR Z)
(FUNCTION LITATOM)))
(LIST (CONS (QUOTE uses:)
(SORT Y (FUNCTION ALPHORDER]))
```

```
(VSUSET
[LAMBDA NIL (* edited (26-JUL-75 . 1857))
```

```

(SETQ DIMUSERFN 1)
(SETQ EOL CHARACTER 31)
(SETQ ALT CHARACTER 27)
(MINES 1200 30)
(EDITV LISPXMARCS (ORR ((E (EXEC --))
(1)
NIL))

```

→ to VARS

```
[DEFINEX (QUOTE ((FILEP (X)
(GETP X (QUOTE FILE)))
(SETFILEP (X V)
(/PUT X (QUOTE FILE)
V))
(REMFILEP (X)
(/REMPROP X (QUOTE FILE)
```

```
(SETQ CCARRAY (ARRAY 32))
[PROG ((J 1)
(MAPC CCLIST (FUNCTION (LAMBDA (X)
(SETA CCARRAY J X)
(ADD1VAR J]
```

→ to VPROP

```
(SETQ VARTYPELIST NIL)
(DEFVARTYPE (QUOTE COMS)
NIL T)
(APPLYMAPC INITVARTYPES (FUNCTION DEFWARTYPE))
(ERRORTYPE 23 (QUOTE (SEARCHDIRS (CADR ERRORMESS)
DIRECTORIES]))
```

```
(ADDA
[LAMBDA (L X V) (* edited NIL)
([LAMBDA (D)
(COND
(D (NCONC1 D V)
L)
(T (CONS (LIST X V)
L])
(FASSOC X L])
```



```

(DEFVARTYPE
[LAMBDA (X COM TYPES) (*'edited (26-JUL-75 . 2152))
[/PUT X (QUOTE NEWPROP)
(DEFINEX (SUBPAIR (QUOTE (Q F SETF))
(LIST (KWOTE X)
(PACK (LIST (QUOTE F)
X))
(PACK (LIST (QUOTE SETF)
X)))
(QUOTE ((F (X)
(ESVAL (FNAME X Q)))
(SETF (X V)
(SAVESET (FNAME X Q)
V T (QUOTE NOPRINT]

[COND
(COM (/PUT COM (QUOTE PRETTYTYPE)
(COND
((NULL TYPES)
(QUOTE NIL))
((LISTP TYPES)
(LIST (QUOTE LAMBDA)
(QUOTE (COM TYPE NAME))
(LIST (QUOTE SELECTQ)
(QUOTE TYPE)
(LIST (COND
((CDR TYPES)
TYPES)
(T (CAR TYPES)))
(QUOTE (CDR COM)))
NIL)))
((NEQ TYPES T)
TYPES]
(ADDTO (QUOTE VARTYPELST)
(LIST (NLIST X COM TYPES)
T)
X])

```

```

(MOVFN
[LAMBDA (FNS NEW) (* edited (3-JUL-74 . 1159))
(PROG ([FL (COND
((LISTP FNS)
(APPEND FNS))
(FNS (LIST FNS]
FN L AFTER M (N NEW))
(UPDATEFILES)
LP (OR FL (RETURN FNS))
(SETQ FN (FNCHECK (CAR FL)))
[COND
((FMEMB FN CHANGEDFNSLST)
(SETQ CHANGEDFNSLST (/DREMOVE FN CHANGEDFNSLST)))
((FMEMB FN L)
(UNFILE FN N)
(SETQ L NIL))
(T (UNFILE FN (FNFILE FN]
(COND
(NEW (COND
((LISTP L))
([OR (FMEMB N FILELST)
(COND
((SETQ N (FNFILE (SETQ AFTER N)))
(ADDSPELL FN 0)
(PRINT (LIST N)
T]
(SETQ L (FFNS N)))
(T (ERROR NEW (QUOTE "not found")
T)))
[COND
((MEMB FN L))
([AND AFTER (SETQ M (MEMB AFTER L))]
(/RPLACD M (CONS FN (CDR M]
(T (SETFFNS N (SETQ L (CONS FN L]
(CHANGEDFILE N FN)
(SETQ AFTER FN)))
(SETQ FL (CDR FL))
(GO LP])

```

```
(UNFILE
 [LAMBDA (FN OLD) (* edited NIL)
 (COND
 (OLD (SETFFNS OLD (/DREMOVE FN (FFNS OLD)))
 (CHANGEDFILE OLD FN) OLD])
```

```
(MOVPROP
 [LAMBDA (PROP OLD NEW) (* edited NIL)
 (COND
 ((GETP OLD PROP)
 (/PUT NEW PROP (GETP OLD PROP))
 (/REMPROP OLD PROP])
```

```
(MOVDEFS
 [LAMBDA (L) (* edited NIL)
 (MAPC L (FUNCTION (LAMBDA (X)
 (MOVD (CAR X)
 (CDR X))
```

```
(UNDEFINE
 [LAMBDA (FN) (* edited (3-JUL-74 . 1200))
 (COND
 ((LISTP FN)
 (MAPCAR FN (FUNCTION UNDEFINE)))
 (T (PROG1 (FNFILE FN)
 (MOVFN FN NIL)
 (/PUTD FN NIL)
 [MAPC FNPROPS (FUNCTION (LAMBDA (X)
 (/REMPROP FN X])
 (/SET (QUOTE CHANGEDFNLSLST)
 (/DREMOVE FN CHANGEDFNLSLST])
```

```
(RNFILE
 [LAMBDA (OLD NEW) (* edited (3-JUL-74 . 1200))
 (MAKENEWFILE NEW)
 (RNFILE1 OLD NEW])
```

```
(MERGEFILE
 [LAMBDA (FROM TO) (* edited (3-JUL-74 . 1200))
 (RNFILE1 FROM TO])
```

```
(RNFILE1
 [LAMBDA (OLD NEW) (* edited (26-JUL-75 . 1857))
 (COND
```

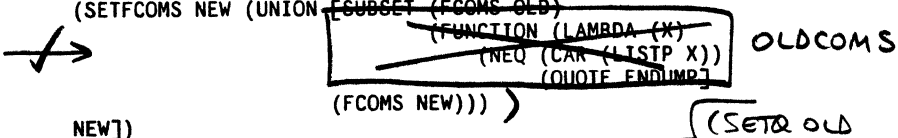
```

  ((LISTP (FILEP OLD))
   (UPDATEFILES)
   (/NCONC (FILEP NEW)
            (CDR (FILEP OLD))))
  (/REMPROP OLD (QUOTE FILE))
  (SETQ FILELST (/DREMOVE OLD FILELST))
  (/DSUBST NEW OLD NOTLISTEDFILES)
  (/DSUBST NEW OLD NOTCOMPILEDFILES)
  [MAPC FILEVARTYPES
   (FUNCTION (LAMBDA (X)
    (SETQ X (CAR X))
    (PROG ((A (PACK (LIST OLD X)))
           (B (PACK (LIST NEW X)))
           L)
          (COND
           ((NEQ A A)
            (QUOTE NOBIND))
           [SETTOPVAL B
            (/NCONC (SETQ L (ESVAL B))
                    (SUBSET
                     (CAR A)
                     (FUNCTION (LAMBDA (X)
                               (NOT (MEMBER X L)
                               (/DSUBST B A OLD COMS))
            (QUOTE NOBIND))
            (SETTOPVAL A (QUOTE NOBIND))
            (PROG ((OLDCOMS (FCOMS OLD)))
                  (VARNAMESLST
```

```

[MAPC (QUOTE (SET DUMP))
  (FUNCTION (LAMBDA (X)
    (PROG [(A (PACK (LIST OLD X)))
      (B (PACK (LIST NEW X)
        (COND
          ((GETD A)
            (COND
              [(GETD B)
                (/NCONC (GETD B)
                  (CDDR (GETD A)
                    (T (RNFN A B)
                      (SETFCOMS NEW (UNION (FSUBST (FCOMS OLD)
                        (FUNCTION (LAMBDA (X)
                          (NEQ (CAR (LISTP X))
                            (QUOTE ENDUMP)
                              (FCOMS NEW))) )
                          (SETQ OLD

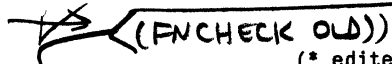
```



```

(RNFN
[LAMBDA (OLD NEW FIX)
  (PROG ((FILE (FNFILE OLD)))
    (RETURN (COND
      (GETD OLD)
        (/MOVD OLD NEW)
        (/PUTD OLD NIL)
        [MAPC FNPROPS (FUNCTION (LAMBDA (X)
          (MOVPROP X OLD NEW)
            (/REMPROP OLD (QUOTE EDITDATE))
            (EDITED NEW)
            (AND FILE (/DSUBST NEW OLD (FFNS FILE))
              (OR (FMEMB NEW (/DSUBST NEW OLD CHANGEDFNLSST))
                (CHANGEDFILE FILE NEW)))
            (ADDSPELL NEW 0)
            [AND FIX (MAPC (COND
              ((EQ FIX T)
                (FFNS FILE))
              (T FIX))
                (FUNCTION (LAMBDA (X)
                  (CHANGENAME X OLD NEW)
                    (CONS NEW (MAPCAR (CDR OLD)
                      (FUNCTION [LAMBDA (X)
                        X])
                        (FUNCTION CDDR)
                          (T (ERROR OLD (QUOTE "not defined")
                            ))

```



```

(* edited (3-JUL-74 . 1735))
      (RETURN (COND
        (GETD OLD)
          (/MOVD OLD NEW)
          (/PUTD OLD NIL)
          [MAPC FNPROPS (FUNCTION (LAMBDA (X)
            (MOVPROP X OLD NEW)
              (/REMPROP OLD (QUOTE EDITDATE))
              (EDITED NEW)
              (AND FILE (/DSUBST NEW OLD (FFNS FILE))
                (OR (FMEMB NEW (/DSUBST NEW OLD CHANGEDFNLSST))
                  (CHANGEDFILE FILE NEW)))
              (ADDSPELL NEW 0)
              [AND FIX (MAPC (COND
                ((EQ FIX T)
                  (FFNS FILE))
                (T FIX))
                  (FUNCTION (LAMBDA (X)
                    (CHANGENAME X OLD NEW)
                      (CONS NEW (MAPCAR (CDR OLD)
                        (FUNCTION [LAMBDA (X)
                          X])
                          (FUNCTION CDDR)
                            (T (ERROR OLD (QUOTE "not defined")
                              ))

```

```

(MAKENEWFILE
[LAMBDA (FILE)
  [COND
    ((NULL (FCOMS FILE))
      (SETFCOMS FILE (LIST (LIST (QUOTE FNS)
        (QUOTE *)
          (FNAME FILE (QUOTE FNS)
            (MAKEFILE FILE (QUOTE NEW]))

```

```

(FNFILE
[LAMBDA (FN ALL)
  (PROG [(FILE (CAR (SOME FILELST (FUNCTION (LAMBDA (X)
    (FMEMB FN (FFNS X)
      (RETURN
        (COND
          (FILE (MOVEFRONT FILE FILELST)
            FILE)
          (ALL
            (CAR (SOME FILELST
              (FUNCTION (LAMBDA (X)
                (OR [SOME (FPROPS X)
                  (FUNCTION (LAMBDA (Y)
                    (FMEMB FN (FGETP (CAR Y)
                      (QUOTE NEWPROP)
                        (FASSOC FN (FMEMOS X])

```

```
(MOVEFRONT
[LAMBDA (X L) (* edited NIL)
  (OR (EQ X (CAR L))
    (AND (FMEMB X L)
      (ATTACH X (DREMOVE X L])))

(CHANGEDFILE
[LAMBDA (FILE FLAG) (* edited (3-JUL-74 . 1739))
  (COND
    ((SETQ FILE (LISTP (FILEP FILE)))
      [/NCONC FILE (OR (LISTP FLAG)
        (NLIST FLAG)
        (LIST (SETQ FLAG T]
      FLAG]))
```

```
(UNCHANGE
[LAMBDA (FILE) (* edited (9-JUL-74 . 1428))
  (COND
    ((NULL FILE)
      (PROG1 (LIST CHANGEDFNLSLST CHANGEDVARSLST)
        (SETQ CHANGEDFNLSLST NIL)
        (SETQ CHANGEDVARSLST NIL)))
    (T (PROG ((L (FILEP FILE)))
      (RETURN (COND
        ((LISTP L)
          (UPDATEFILES)
          (RPLACD L NIL)
          L]))
```

```
(FIXENDUMP
[LAMBDA (FILE) (* edited (26-JUL-75 . 2151))
  (PROG [(COMS (MAPCONC VARTYPELST
    (FUNCTION (LAMBDA (X)
      (PROG [(N (FNAME FILE (CAR X)
        (RETURN (COND
          ((ESVAL N)
            (LIST (LIST (CADR X)
              (QUOTE *)
              N]
          (APPLY (QUOTE EDITV)
            (LIST (FNAME FILE (QUOTE COMS))
              (CONS -1 (APPEND COMS))
              (QUOTE (F (ENDUMP &)))
              (QUOTE DELETE)))
          (RETURN COMS])
```



```
(NEWVAR
[LAMBDA (X) (* edited (26-JUL-75 . 1940))
  (PROG ((D (FILEVARP X))
    E C)
    (COND
      ([AND D (CADR (SETQ E (FASSOC (CDR D)
        VARTYPELST]
        (SETQ C (LIST (CADR E)
          (QUOTE *)
          X))
      (COND
        ([NOT (MEMBER C (FCOMS (CAR D]
          (APPLY (QUOTE EDITV)
            (LIST (FNAME (CAR D)
              (QUOTE COMS))
              (LIST (QUOTE ADDCOM)
                C]))
```

```

(FILEVARP
[LAMBDA (X)
  (PROG (VL FL)
    (RETURN (COND
      ([SETQ VL
        (SOME VARTYPELST
          (FUNCTION (LAMBDA (Y)
            (AND (SETQ Y
              (STRPOS (CAR Y)
                X
              (IMINUS (NCHARS (CAR Y)))
                NIL T))
            (SETQ FL
              (SOME FILELST
                (FUNCTION (LAMBDA (Z)
                  (EQ (STRPOS Z X NIL NIL T T)
                    Y]
                (CONS (CAR FL)
                  (CAAR VL]))

```

```

(DEFINEX
[LAMBDA (L)
  [RESETVAR CHANGEDFNLSLST NIL
  (APPLY
    (QUOTE DEFINEQ)
    (SUBSET
      L
      (FUNCTION (LAMBDA (X)
        (PROG [(FN (CAR X))
          (FORM (COND
            ((CDDR X)
              (CONS (QUOTE LAMBDA)
                (CDR X)))
            (T (CADR X]
          (RETURN
            (NOT
              (COND
                ((EQUAL (GETD FN)
                  FORM))
                ((EXPRP FN)
                  NIL)
                ((FGETD FN)
                  (OR (EQUAL (GETP FN (QUOTE EXPR))
                    FORM)
                    (COND
                      ((OR (AND (EQ (CAR FORM)
                        (QUOTE LAMBDA))
                          (EQUAL (GETP FN (QUOTE MACRO))
                            (CDR FORM)))
                        (EQUAL (GETP FN (QUOTE BLKLIBRARYDEF))
                          FORM))
                    (/PUT FN (QUOTE EXPR)
                      FORM]
                (/PUT FN (QUOTE EXPR)
                  FORM]

```

(MAPCAR L (FUNCTION CAR])

```

(CONSTANT
[LAMBDA (X)
  X])
(* edited NIL)

```

```

(DWIMUSERFN
[LAMBDA NIL
  (PROG (FAULTFN)
    (RETURN (COND
      ((AND (NOT FAULTAPPLYFLG)
        (LISTP FAULTX)
        (STRPOS (QUOTE F/)
          (SETQ FAULTFN (CAR FAULTX))
            1 NIL T))
        (/RPLACA FAULTX (QUOTE FUNCTION))
      [COND
        ((NEQ FAULTFN (QUOTE F/))
          (/RPLACD FAULTX
            (LIST (CONS (QUOTE LAMBDA)
              (CONS (CDDR (UNPACK FAULTFN))
                (CDR FAULTX]
          FAULTX])

```

```
(FPROPSAVES
[LAMBDA (FILE PROP) (* edited (26-JUL-74 . 1642))
  (MAPCONC (FCOMS FILE)
    (FUNCTION (LAMBDA (X)
      (COND
        ([AND (LISTP X)
          (FMEMB (CAR X)
            (QUOTE (PROP IFPROP COMPROP COMPROP* ADDPROPS))
          (PROG [(P (CADR X))
            (F (EQ (CAR X)
              (QUOTE IFPROP)))
          (Y (COND
            ((EQ (CADDR X)
              (QUOTE *))
            (EVAL (CADDRR X)))
          (T (CDDR X]
        (RETURN (COND
          ((COND
            ((EQ P PROP))
            ((LISTP P)
              (FMEMB PROP P))
            ((EQ P (QUOTE ALL))
              (SETQ F T)))
          (COND
            [F (SUBSET Y (FUNCTION (LAMBDA (Z)
              (GETP Z PROP)
            ((EQ (CAR X)
              (QUOTE ADDPROPS))
            (MAPCAR Y (FUNCTION CAR)))
            (T (APPEND Y])
```

```
(CORECOMPILE
[LAMBDA (L D) (* edited (3-JUL-74 . 1740))
  ([LAMBDA (LAPFLG STRF SVFLG LCFIL LSTFIL)
    (MAPC (OR (LISTP L)
      (LIST L))
    (FUNCTION (LAMBDA (X)
      (COND
        (D (COMPILE2 X D))
        ((EXPRP X)
          (COMPILE2 X (GETD X)))
        (T (LISPXPRT (CONS X (QUOTE (not compilable)))
          T]
    NIL T T NIL NIL)
  L])
```

```
(SUBEXEC
[LAMBDA NIL (* edited NIL)
  (COND
    ((OR (EQ (CAR (QUOTE EXEC))
      (QUOTE NOBIND))
    (EQ (CAR (RFST8 EXEC))
      -1))
    (SETQ EXEC (SUBSYS)))
  (T (SUBSYS EXEC])
```

```
(DETACH
[LAMBDA NIL (* edited (9-NOV-74 . 124))
  (LISPXPRT " Detaching job # " T)
  (RESETFORM (RADIX 10)
    (LISPXPRT1 (JSYS 11 NIL NIL NIL 3)
      T))
  (LISPXPRT " at " T)
  (LISPXPRT1 (DATE)
    T)
  (JSYS 77) (* Following printing is held up
    until job is re-attached.)
  (LISPXPRT " Re-attaching at " T)
  (LISPXPRT1 (DATE)
    T)
  T])
```

```

(RECOR
  [LAMBDA (ITEM FILE)                                (* edited (5-AUG-74 . 1401))
    (PROG ((DATA (LIST (MKATOM (USERNAME))
                        (DATE)
                        (GETAB (QUOTE LHOSTN)
                              0)
                        ITEM))
          OUTPUT)
      (RETURN (COND
                ((NLSETQ (SETQ OUTPUT (APPENDFILE FILE)))
                 (PRINT DATA)
                 (CLOSEF (OUTPUT OUTPUT))
                 DATA]))

```

```

(RECADVISE
  [LAMBDA (FN WHEN WHERE EXP FILE)                  (* edited NIL)
    (COND
      ((FNTYP FN)
       (ADVISE FN WHEN WHERE (LIST (QUOTE RECOR)
                                   (LIST (QUOTE CONS)
                                         (KWOTE FN)
                                         EXP)
                                   FILE)))
      (T (LISPXPRT (CONS FN (QUOTE (not defined)))
                  T]))

```

```

(ERRORTYPE
  [LAMBDA (N FORM)                                  (* edited (7-AUG-74 . 1023))
    (/SET (QUOTE ERRORTYPELST)
          (ADDFORM ERRORTYPELST N FORM))]

```

```

(PRINTYPE
  [LAMBDA (N FORM)                                  (* edited (7-AUG-74 . 1023))
    (/SET (QUOTE PRINTYPELST)
          (ADDFORM PRINTYPELST N FORM))]

```

```

(ADDFORM
  [LAMBDA (L N FORM)                                (* edited (7-AUG-74 . 830))
    (PROG ((D (FASSOC N L)))
          (RETURN (COND
                    ((NULL D)
                     (CONS (LIST N FORM)
                             L))
                    ((MEMBER FORM (CDR D))
                     L)
                    (T (/NCONC1 D FORM)
                       L))

```

```

(LISPXPRT1
  [LAMBDA (X FILE)                                  (* edited NIL)
    (LISPXPRT1 X FILE)
    (LISPXTERPRI FILE)]

```

```

(LISPXPRTINQ
  [NLAMBDA (PRINX PRINF)                            (* edited NIL)
    (LISPXPRTINQ PRINX (EVAL PRINF))]

```

```

(LISPXPRTINTQ
  [NLAMBDA (PRINX PRINF)                            (* edited NIL)
    (LISPXPRTINTQ PRINX (EVAL PRINF))]

```

```

(NPRINT
  [LAMBDA (X FILE PTBL)                              (* edited (6-JAN-75 . 1814))
    (PROG1 (NPRINT2 X FILE PTBL)
           (TERPRI FILE))

```

```

(NPRINT2
  [LAMBDA (X FILE PTBL)                              (* edited (6-JAN-75 . 1814))
    (NPRINT2 X FILE PTBL (FUNCTION PRIN2))

```

(NPRIN1  
[LAMBDA (X FILE PTBL) (\* edited (6-JAN-75 . 1815)) |  
(NPRINX X FILE PTBL (FUNCTION PRIN1])

(NPRINX  
[LAMBDA (X PFILE PTBL PFN) (\* edited (19-MAR-75 . 1625))

(\* Must be outside NPRIN block because of possible recursive calls of NPRIN from print functions.)

(PROG ((PLEV (AND (EQ (OR PFILE (OUTPUT))  
T)  
0)))  
(NPRIN X)  
(RETURN X])

(NPRIN  
[LAMBDA (X) (\* edited (19-MAR-75 . 1626))

(PROG ((TYP (NTYP X))  
(PLEV PLEV))  
(COND  
((AND (EQ TYP 8)  
PLEV  
(IGREATERP (SETQ PLEV (ADD1 PLEV))  
(PRINTLEVEL)))  
(PRIN1 (QUOTE &  
PFILE)  
(RETURN)))  
([LAMBDA (FN)  
(COND  
([AND FN (SOME FN (FUNCTION (LAMBDA (XFN)  
(APPLY\* XFN X PFILE PTBL (EVQ PFN)  
(RETURN]  
(CDR (FASSOC TYP PRINTYPELST)))  
[SELECTQ TYP  
(8 (GO LIS))  
(RETURN) (COND  
((ILESSP TYP 31)  
(APPLY\* (EVQ PFN)  
X PFILE PTBL))  
(T (NPRIN X TYP))

LIS (PRIN1 (QUOTE %(  
PFILE)

LL (NPRIN (CAR X))

(COND  
((AND PLEV (IGREATERP PLEV (PRINTLEVEL)))  
(PRIN1 (QUOTE --)  
PFILE))  
((LISTP (SETQ X (CDR X)))  
(SPACES 1 PFILE)  
(GO LL))  
(X (PRIN1 (QUOTE " . ")  
PFILE)  
(NPRIN X)))  
(PRIN1 (QUOTE %)  
PFILE])

(PRIN2 TYP PFILE PTBL)

(PRIN1 X PFILE)



```

(UPRIN
[LAMBDA (X TYP (* edited (7-AUG-74 . 834))
(PRIN1 (QUOTE (
PFILE)
(PRIN2 [CDR (OR (FASSOC TYP TYPENAMELST)
(CAR (SETQ TYPENAMELST
(CONS (CONS TYP (PACK (LIST T TYP)))
TYPENAMELST]
PFILE)
[PROG ((N (GETNPTRS TYP))
(J -1))
LP (COND
((ILESSP (SETQ J (ADD1 J))
N)
(SPACES 1 PFILE)
[NPRIN ([LAMBDA (P)
(COND
((ZEROP (LOGAND J 1))
(CAR P))
(T (CDR P)
(VAG (IPLUS (LRSH J 1)
(LOC X)
(GO LP)))
(SETQ N (GETNWRDS TYP))
(SETQ J (LRSH J 1))
LW (COND
((ILESSP (SETQ J (ADD1 J))
N)
(SPACES 1 PFILE)
[NPRIN (OPENR (IPLUS J (LOC X)
(GO LW)
(PRIN1 (QUOTE (
PFILE))
)

```

(RPAQQ VSUVARS (INITVARTYPES CCLIST FNPROPS FILEPROPS (SYSVARS)))

```

(RPAQQ INITVARTYPES ((FNS FNS (FNS))
(VARS VARS (VARS))
(GLOBALS GLOBALVARS)
(ADVICE ADVISE:)
(BLOCKS BLOCKS)
(/FNS NEW/FNS)))

```

(MACROS MACROS)

```

(RPAQQ CCLIST
(I CAR CDR CAAR CDAR CADR CDDR CAAAR CDAAR CADAR CDDAR CAADR CDADR
CADDR CDDDR CAAAAR CAAAAR CADAAR CDDAAR CAADAR CDADAR CADDAR CDDDR
CAAADR CDAADR CADADR CDDADR CAADDR CDADDR CADDDR CDDDDR NIL))

```

```

(RPAQQ FNPROPS (CODE 'EXPR))
(RPAQQ FILEPROPS (FILE FILECHANGES FILEDATES FILEMAP FILEVARNAMES))

```

```

(RPAQQ SYSVARS (GLOBALVARS NLAMA NLAML LINKFNS NOLINKFNS BLKLIBRARY
CHANGEDFNSLST CHANGEDVARSLSLST NEWVARSLSLST VARTYPELST))

```

```

(RPAQQ VSUGLOBALS (PRINTYPELST TYPENAMELST))
[DECLARE: EVAL@COMPILE
(ADDOVAR GLOBALVARS PRINTYPELST TYPENAMELST)
]

```

```

[RPAQQ VSUBLOCKS ((NPRIN NPRIN UPRIN (LOCALFREEVARS PFN)
(SPECVARS PLEV)
[DECLARE: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY
(BLOCK: NPRIN NPRIN UPRIN (LOCALFREEVARS PFN)
(SPECVARS PLEV))
]

```

```

[RPAQQ VSUADVICE ((DEFINE AFTER NIL (DEFSAVEX !VALUE))
(UPDATEFILES BEFORE NIL (PROGN (CHECKNEWVARS NEWVARSLSLST)
(SETQ NEWVARSLSLST NIL))))
(SAVESET AROUND (PROG ((OLDV (GETTOPVAL NAME)))

```

```

(SELECTQ OLDV
((NIL NOBIND)
(LISTP (GETTOPVAL
NAME))))
NIL)

```

```

[COND ((AND (EQ OLDV (QUOTE NOBIND))
(NEQ (GETTOPVAL NAME)
(QUOTE NOBIND))))

```

```

(ADDTQ (QUOTE NEWVARSLSLST) (NEWVAR NAME)
(LIST NAME)

```

(RETURN VALUE)))

```

(MAKEFILE BEFORE NIL (DUMPFIX FILE))
(FILES? AFTER NIL (RETURN CHANGEDVARSLSLST))
(EDITF AFTER NIL (EDITED !VALUE))

```

```

(APPLYMAPC [QUOTE ((DEFINE AFTER NIL (DEFSAVEX !VALUE))
(UPDATEFILES BEFORE NIL (PROGN (CHECKNEWVARS NEWVARSLSLST)
(SETQ NEWVARSLSLST NIL)))
(SAVESET AROUND (PROG ((OLDV (GETTOPVAL NAME)))
*
[COND ((AND (EQ OLDV (QUOTE NOBIND))
(NEQ (GETTOPVAL NAME)
(QUOTE NOBIND)))
(ADDTO (QUOTE NEWVARSLSLST)
(LIST NAME]
(RETURN VALUE)))
(MAKEFILE BEFORE NIL (DUMPFIX FILE))
(FILES? AFTER NIL (RETURN CHANGEDVARSLSLST))
(EDITF AFTER NIL (EDITED !VALUE]
(FUNCTION ADVISE))
[DECLARE: DOEVAL@COMPILE
(DEFLIST(QUOTE(
[CONSTANT (OPENX (PROG [(VAL (EVAL (CAR OPENX)
(RETURN (COND ((OR (LITATOM VAL)
(LISTP VAL))
(KWOTE VAL))
(T VAL]
)))(QUOTE MACRO))
]
(ADDTOVAR DIRECTORIES)
(ADDTOVAR PRINTYPELST)
(ADDTOVAR TYPENAMELST)
(ADDTOVAR NEWVARSLSLST)
[ADDTOVAR PRETTYDEFMACROS [ENDUMP (X)
(E (FIXENDUMP (QUOTE X)
[ADVISE: X (P (APPLYMAPC (QUOTE X)
(FUNCTION ADVISE]
[NEW/FNS X (P (MAPC (QUOTE X)
(FUNCTION NEW/FN]
[GLOBALVARS X (DECLARE: EVAL@COMPILE (ADDVARS (GLOBALVARS . X]
(ADDPROPS X (P * (ADDPROPFORM (CAR (QUOTE X))
(CDR (QUOTE X]
(ADDTOVAR LISPXMACHOS (CON (SUBEXEC)
(DET (DETACH))
(QUIT (LOGOUT)))
(ADDTOVAR LISPXFNS (FRPLACA . /RPLACA)
(FRPLACD . /RPLACD))
(ADDTOVAR BREAKMACROS (EP EVAL VALUE))
(ADDTOVAR LISPXCOMS CON DET QUIT EP)
(ADDTOVAR NILCOMS (VARS * SYSVARS))
[ADDTOVAR USERMACROS (ADDCOM C MARK (LPQ (IF (EQ (## 2 2)
(QUOTE *)))
(NTH 2))
(ORR ((-2 . C))
((N . C)))
←)
[CK NIL (E (CK (## +]
(U NIL UP)
[FIXF/ NIL (F (F/+[ --)
N)
(IF (EQ (## 1)
(QUOTE F/))
((1 FUNCTION))
([IF (EQ (## 1)
(QUOTE F/L))
((1 LAMBDA))
((I 1 (QUOTE LAMBDA)
(CDDR (UNPACK (## 1]
(EMBED IN FUNCTION]

```

```

(SUBFN X (IF (SELECTQ (## 1)
                    ([LAMBDA NLAMBDA]
                     T)
                    NIL)
  [[E (DEFINE (LIST (LIST (CAR (QUOTE X))
                          (COPY (##)
                                UP
                                (I 1 (CAR (QUOTE X]
  ([E (DEFINE (LIST (LIST (CAR (QUOTE X))
                          (CADR (QUOTE X))
                          (COPY (##)
                                UP
                                (I 1 (CONS (CAR (QUOTE X))
                                              (CADR (QUOTE X]

(ADDTOVAR EDITCOMSA FIXF/ U CK)
(ADDTOVAR EDITCOMSL SUBFN ADDCOM)
[DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVERS
  (ADDTOVAR NLAMA)
  (ADDTOVAR NLAML LISPXPRINTQ LISPXPRINQ)
]
[DECLARE: DONTCOPY
  (FILEMAP (NIL (2188 32519 (COUNTFILES 2200 . 2486) (COUNTFILE 2490 . 2745)
    (COUNTCOM 2749 . 4278) (COUNTPROPS 4282 . 4476) (COUNTUP 4480 . 4758) (ESVAL
    4762 . 4934) (DUMPFIX 4938 . 5127) (COMPROP 5131 . 5478) (BLOCKCHECK 5482
    . 8116) (ASSOCA 8120 . 8280) (LCHECK 8284 . 8718) (ADDPROPFORM 8722 . 8873)
    (ADDLIST 8877 . 9100) (ADDTO 9104 . 9398) (ENDLOAD 9402 . 9532) (DOPRETTYCOMS
    9536 . 10401) (DOPEVAL 10405 . 10591) (RPA 10595 . 10685) (DOPDEC 10689 .
    10951) (MAKELOAD 10955 . 11622) (NAMEPART 11626 . 11949) (FNAME 11953 . 12129)
    (DEFSAVEX 12133 . 12342) (DEFSAVE 12346 . 12672) (DEFTAIL1 12676 . 12818)
    (DEFGETP 12822 . 13054) (/DEFPUT 13058 . 13173) (DEFPUT 13177 . 14969) (EDITED
    14973 . 15245) (EDITDATE 15249 . 15368) (EDITFIX 15372 . 15629) (CK 15633
    . 16106) (VSUSET 16110 . 16907) (ADDA 16911 . 17090) (DEFVARTYPE 17094 . 17979)
    (MOVFN 17983 . 18988) (UNFILE 18992 . 19157) (MOVPROP 19161 . 19325) (MOVDEFS
    19329 . 19474) (UNDEFINE 19478 . 19825) (RNFILE 19829 . 19961) (MERGEFILE
    19965 . 20077) (RNFILE1 20081 . 21214) (RNFN 21218 . 22039) (MAKENEWFILE 22043
    . 22297) (FNFILE 22301 . 22774) (MOVEFRONT 22778 . 22927) (CHANGEDFILE 22931
    . 23157) (UNCHANGE 23161 . 23523) (FIXENDUMP 23527 . 24021) (CHECKNEWVARS
    24025 . 24514) (FILEVARP 24518 . 24999) (DEFINEX 25003 . 25799) (CONSTANT
    25803 . 25885) (DWIMUSERFN 25889 . 26372) (FPROPSAVES 26376 . 27155) (
    CORECOMPILE 27159 . 27550) (SUBEXEC 27554 . 27788) (DETACH 27792 . 28257)
    (RECOR 28261 . 28627) (RECADVISE 28631 . 28922) (ERRORTYPE 28926 . 29081)
    (PRINTYPE 29085 . 29237) (ADDFORM 29241 . 29522) (LISPXPRINT1 29526 . 29652)
    (LISPXPRINQ 29656 . 29768) (LISPXPRINTQ 29772 . 29886) (NPRINT 29890 . 30064)
    (NPRIN2 30068 . 30214) (NPRIN1 30218 . 30364) (NPRINX 30368 . 30688) (NPRIN
    30692 . 31701) (UPRIN 31705 . 32516))))))
STOP

```







```

(PROPPFORMS
 [LAMBDA (P L MODE A M2) (* edited (27-JUL-75 . 257))
 (COND
 ((LISTP P)
 (FORMPROP P L MODE A))
 (T
 [COND
 ((LISTP A)
 (SETQ A (PREVAL A])
 (PROG [M (N (OR A P))
 (NOSET (MEMB (QUOTE NOSET)
 L))
 (UNDO (MEMB (QUOTE UNDO)
 L))
 (UNDOABLE (AND WANTDEFS (MEMB (QUOTE UNDOABLE)
 L)))
 FORMS Q D D1 D2 BL UL S1 (S (PACK (LIST (QUOTE SET)
 P])
 (COND
 ([NOT (LITATOM (OR MODE (SETQQ MODE PROP]
 (LISPXPRT (CONS MODE (QUOTE (- ILLEGAL MODE)))
 T)
 (RETURN))
 ([NULL (SETQ FORMS (GETP MODE (QUOTE NEWPROPPFORMS]
 (LISPXPRT (CONS MODE (QUOTE (- NO SUCH MODE)))
 T)
 (RETURN)))
 [SETQ D
 (COND
 [WANTDEFS (SETQ M (LIST (SETQ Q (KWOTE N))
 P S N))
 (CONS (SUBPLY (QUOTE (Q P S N))
 M
 (CAAR FORMS))
 (AND (NOT NOSET)
 (LIST (SUBPLY (QUOTE (Q P S N))
 M
 (CADAR FORMS]
 (T (CONS P (AND (NOT NOSET)
 (LIST S]
 (COND
 (UNDOABLE (SETQ D (SUBLIS LISPXFNS D)))
 (UNDO [NCONCI D] (CONS (SETQ S1 (PACK (LIST (QUOTE /)
 S)))
 (SUBLIS LISPXFNS (CDADR D]
 [SETQ UL (CONS (CONS S S1)
 LISPXFNS] (T S1]
 [SETQ D
 (APPEND
 (SETQ D1 D)
 (SETQ D2
 (MAPCONC
 L
 (FUNCTION (LAMBDA (X)
 (SELECTQ
 X
 ((NOSET UNDO UNDOABLE BLKFN)
 NIL)
 (MACRO (PRPUT P (QUOTE MACRO)
 (CDAR D))
 NIL)
 (SETMACRO (AND (NOT NOSET)
 (PRPUT S (QUOTE MACRO)
 (CDADR D)))
 NIL)

```

(COND [WANTDEFS
 (T S1]

```

(PROG [(DO (GETPROPFORM X FORMS))
DL D1 (P1 (PACK (LIST X P]
(RETURN
(COND
(DO
[SETQ DL
(LIST
(COND
(WANTDEFS
(SUBPLY (QUOTE (FN Q P S N))
(CONS P1 M)
DO))
(T P1]
[COND
(UNDOABLE (SETQ DL (SUBLIS LISPXFNS
DL)))
([AND
UNDO
(NEQ
[SETQ DO
(COND
(WANTDEFS (CDAR DL))
(T
(SUBPLY
(QUOTE (FN Q P S N))
[CONS
P1
(OR
M
(SETQ M
(LIST
(KWOTE N)
P S N]
DO]
(SETQ D1 (SUBLIS UL DO]
(SETQ P1
(PACK (LIST (QUOTE /)
P1)))
(NCONC1 DL
(COND
(WANTDEFS (CONS P1 D1))
(T P1]
DL)
(T (LISPXPRINT
(CONS (CAR M1)
(QUOTE (- can't define)))
T)
NIL]
[COND
(WANTDEFS (PRPUT P (QUOTE NEWPROP)
(MAPCAR D (FUNCTION CAR)))
[COND
[[SETQ BL
(AND (MEMB (QUOTE BLKFN)
L)
(APPEND (AND (NOT (MEMB (QUOTE MACRO)
L))
(LIST (CAAR D1)))
(AND (NOT (MEMB (QUOTE SETMACRO)
L))
(NLIST (CAADR D1)))
(NLIST S1)
(MAPCAR D2 (FUNCTION CAR)
(PRPUT P (QUOTE BLKFN)
BL)
(MAPC BL
(FUNCTION (LAMBDA (X)
(/PUT X (QUOTE BLKLIBRARYDEF)
(CONS (QUOTE LAMBDA)
(CDR (FASSOC X D]
(T (/REMPROP P (QUOTE BLKFN]

```



```

      [COND
        (CLISPROPFLG (PRAPPLY (QUOTE ACCESSDEF)
                              (NLIST P P
                                       (AND (NOT NOSET)
                                             S)
                                       S1]
          (COND
            (S1 (PRAPPLY (QUOTE NEW/FN)
                        (LIST S]
      (RETURN D])

(GETPROPFORM
 [LAMBDA (OP FORMS)                                (* edited (18-JAN-75 . 1425))
  (PROG [(D (FASSOC OP (CDR FORMS)
                    (RETURN (COND
                              (D (CDR D))
                              (T (GETP OP (QUOTE NEWPROPDEFAULT]))

(SUBPLY
 [LAMBDA (OLD NEW FORM)                            (* edited NIL)
  (COND
    ((LITATOM FORM)
     (APPLY FORM NEW))
    (T (SUBPAIR OLD NEW FORM]))

(ACCESSDEF
 [LAMBDA (PROP FN SETFN /SETFN)                   (* edited (23-OCT-74 . 1523))
  [COND
    (PROP (PRPUT PROP (QUOTE ACCESSFN)
                  FN)
          (COND
            ((NEQ FN PROP)
             (PRPUT FN (QUOTE ACCESSFN)
                     (LIST PROP]
          [COND
            (SETFN (PRPUT FN (QUOTE SETFN)
                          SETFN)
                  (PRPUT SETFN (QUOTE SETFN)
                              (LIST FN))
          (COND
            (/SETFN (PRPUT SETFN (QUOTE CLISPCCLASS)
                                SETFN)
                   (PRPUT SETFN (QUOTE CLISPCCLASSDEF)
                               (LIST (QUOTE ACCESS)
                                     SETFN /SETFN))
                   (PRPUT SETFN (QUOTE LISPFN)
                              SETFN)
                   (PRPUT /SETFN (QUOTE CLISPCCLASS)
                              SETFN)
                   (SAVESET (QUOTE DECLWORDS)
                            (UNION (LIST SETFN /SETFN)
                                   DECLWORDS)
                            NIL
                            (QUOTE NOSAVE]
      PROP])

(PRPUT
 [LAMBDA (X P V)                                  (* edited (27-JUL-75 . 301))
  [COND
    (WANTDEFS (COND
              ((NEQUAL V (GETP X P))
               (/PUT X P V)))
    (COND
      ((NEQ PROPSAVES T)
       (SETQ PROPSAVES (ADDA PROPSAVES P X]
  V])

```

```

(PRAPPLY
[LAMBDA (FN L)                                     (* edited (27-JUL-75 . 302))
(COND
(WANTDEFS (PROG1 (APPLY FN L)
(COND
((NEQ COMSAVES T)
(SETQ COMSAVES
(ADDA COMSAVES (QUOTE P)
(CONS FN (MAPCAR L (FUNCTION KWOTEX]))

(PREVAL
[LAMBDA (EXP)                                     (* edited (27-JUL-75 . 304))
(COND
((NLISTP EXP)
EXP)
[WANTDEFS (PROG1 (EVAL EXP)
(COND
((NEQ COMSAVES T)
(SETQ COMSAVES (ADDA COMSAVES (QUOTE P)
EXP]

((EQ (CAR EXP)
(QUOTE SETHASHQ))
(CADR EXP))
(T (EVAL EXP]))

(KWOTEX
[LAMBDA (X)                                       (* edited (23-OCT-74 . 1825))
(COND
((OR (NULL X)
(EQ X T)
(NUMBERP X))
X)
(T (KWOTE X]))

(CCGETF
[LAMBDA (Q P S N)                                 (* edited NIL)
(LIST P (QUOTE (X))
(CCGET (QUOTE X)
N]))

(CCSETF
[LAMBDA (Q P S N)                                 (* edited NIL)
(LIST S (QUOTE (X V))
(CCSET (QUOTE X)
N
(QUOTE V]))

(FORMPROP
[LAMBDA (FORM OPTIONS NAME VARS)                 (* edited (27-JUL-75 . 303))
(NCONC (FORMAT1 FORM 1)
(COND
(NAME (LIST (COND
(WANTDEFS (LIST NAME VARS (FORMAT2 FORM VARS)))
(T NAME]))

(FORMAT1
[LAMBDA (FORM K)                                 (* edited (23-OCT-74 . 1533))
(COND
((FORMCONSTP FORM)
NIL)
((NLISTP FORM)
(PROPFOMS FORM OPTIONS (QUOTE CC)
K))
(T (NCONC (FORMAT1 (CAR FORM)
(LSH K 1))
(FORMAT1 (CDR FORM)
(ADD1 (LSH K 1]))

```

```
(FORMAT2
[LAMBDA (FORM VARS)                                (* edited NIL)
(COND
  ((FORMCONSTP FORM)
   FORM)
  ((NLISTP FORM)
   (COND
    ((FMEMB FORM VARS)
     FORM)
    ((LITATOM FORM)
     (KWOTE FORM))
    (T FORM)))
  (T (FORMCONS (FORMAT2 (CAR FORM)
                       VARS)
               (FORMAT2 (CDR FORM)
                       VARS)
        T]))
```

```
(FORMCONS
[LAMBDA (X Y FLAG)                                (* edited NIL)
(COND
  [(FORMCONSTP Y)
   (COND
    [(AND (NULL FLAG)
          (FORMCONSTP X))
     (KWOTE (CONS (FORMCONSTVAL X)
                  (FORMCONSTVAL Y))
            ((NULL Y)
             (LIST (QUOTE LIST)
                   X))
            (T (LIST (QUOTE CONS)
                    X Y]
    [(AND (LISTP Y)
          (EQ (CAR Y)
              (QUOTE LIST)))]
     (CONS (QUOTE LIST)
           (CONS X (CDR Y))
            (T (LIST (QUOTE CONS)
                    X Y]))]
```

```
(FORMCONSTP
[LAMBDA (X)                                        (* edited NIL)
(OR (NULL X)
    (EQ X T)
    (COND
     ((LISTP X)
      (EQ (CAR X)
          (QUOTE QUOTE)))
     ((NOT (LITATOM X))
```

```
(FORMCONSTVAL
[LAMBDA (X)                                        (* edited NIL)
(COND
  ((LISTP X)
   (CADR X))
  (T X])
```

```
(CNTH
[LAMBDA (X N)                                    (* edited NIL)
(OR (FNTH X N)
    (HELP (QUOTE CNTH)
          N])
```

```
(CCGET
[LAMBDA (X N)                                    (* edited NIL)
(COND
  ((ILESSP N 32)
   (LIST (ELT CCARRAY N)
         X))
  (T (LIST (ELT CCARRAY (IPLUS 16 (LOGAND N 15)))
          (CCGET X (RSH N 4]))
```

```

(CCSET
  [LAMBDA (X N Y)
    (LIST (SELECTQ (LOGAND N 1)
            (1 (QUOTE FRPLACD))
            (QUOTE FRPLACA))
          [COND
            ((ILESSP N 4)
             X)
            (T (CCGET X (RSH N 1]
              Y]))
    (* edited NIL)

(CCNTH
  [LAMBDA (X N)
    (COND
      [(ILESSP N 6)
       (CCGET X (SUB1 (LSH 1 N]
        (T (LIST (QUOTE NTH)
                 X N]))
    (* edited NIL)

(ACCFN
  [LAMBDA (NAME OP FLAG)
    (PROG ((L (QUOTE (T T)))
           (FN)
           (FRPLACA L OP)
           (FRPLACA (CDR L)
                     NAME)
           (SETQ FN (PACK L))
           (RETURN (COND
                    ((FNTYP FN)
                     FN)
                    ((NULL FLAG)
                     (ERROR (QUOTE "Undefined access function")
                              FN))))
    (* edited (9-APR-75 . 1402))

(AGETP
  [LAMBDA (X P)
    (APPLY* P X)]
  (* edited (9-APR-75 . 1403))

(APUT
  [LAMBDA (X P V)
    (APPLY* (ACCFN P (QUOTE SET))
            X V)]
  (* edited (9-APR-75 . 1403))

(/APUT
  [LAMBDA (X P V)
    (UNDOSAVE (LIST (QUOTE /APUT)
                    X P (AGETP X P)))
    (APUT X P V)]
  (* edited (9-APR-75 . 1403))

(AREMPROP
  [LAMBDA (X P)
    ([LAMBDA (R)
     (COND
       (R (APPLY* R X))
       ((AGETP X P)
        (APUT X P NIL)
        P]
     (ACCFN P (QUOTE REM)
              T))]
  (* edited (9-APR-75 . 1406))

(/AREMPROP
  [LAMBDA (X P)
    (UNDOSAVE (LIST (QUOTE /APUT)
                    X P (AGETP X P)))
    (AREMPROP X P)]
  (* edited (9-APR-75 . 1405))

(ADEFLIST
  [LAMBDA (L P)
    (PROG [(S (ACCFN P (QUOTE SET))
              [MAPC L (FUNCTION (LAMBDA (X)
                                (APPLY* S (CAR X)
                                           (CADR X)
                                           (RETURN P]))
    (* edited (9-APR-75 . 1403))

```



(OPENFN

[LAMBDA (FN L FORM)  
(PROG ((D (GETD FN))

(\* edited (23-OCT-74 . 1833))

~~(K 2)  
L0 L1 S S1 M)~~

(COND  
((EXPRP FN))  
[(SETQ D (GETP FN (QUOTE EXPR))  
(T (RETURN NIL)))]

[COND  
((EQ L T))  
(SETQ L (CADR D))

[SETQ M  
(COND  
(NULL L)  
D)  
((EQUAL L (CADR D))  
(CDR D))

[PROG (VARS ARGS SUBS)

~~(T~~ [MAPC (CADR D)  
(FUNCTION (LAMBDA (X)

[COND  
((FMEMB X L)  
(SETQ L0 (CONS X L0))  
(SETQ S (CONS (CGGET (QUOTE OPENX))

~~S1))  
(T (SETQ L1 (CONS X L1))  
(SETQ S1 (CONS (CGGET (QUOTE OPENX))~~

(SETQ VARS (CONS X VARS))

(CAR (SETQ ARGS (CONS  
(PACK <'S1 X>) ARGS)))

~~(SETQ K (1+TIMES 2 (ADD1 K))~~

~~(LIST  
(QUOTE OPENX)  
(CONS (QUOTE LIST)  
(CONS [LIST (QUOTE SUBPAIR)  
(KWOTE L0)  
(CONS (QUOTE LIST)  
(KWOTE (CONS (CAR D)  
(CONS (DREVERSE L1)  
(OR (NLIST FORM)  
(CDDR D])~~

(RETURN <SUBS

<< (CAR D) (DREVERSE VARS)

!(OR (NLIST FORM)

(CDDR D))>>

!(DREVERSE ARGS)>>]

~~(DREVERSE S1]  
(OR (EQUAL M (GETP FN (QUOTE MACRO)))  
(/PUT FN (QUOTE MACRO)  
M))  
(RETURN FN])~~

(DUMPOPEN

[LAMBDA (L)  
(MAPCAR L (FUNCTION (LAMBDA (X)  
(APPLY (QUOTE OPENFN)  
X)  
(CAR X]))

(\* edited (27-JUL-75 . 306))

(DUMPBLKFNS

[LAMBDA (L)  
(MAPC L (FUNCTION (LAMBDA (X)  
(COND  
((EXPRP X)  
(PUT X (QUOTE BLKLIBRARYDEF)  
(GETD X)))  
((GETP X (QUOTE BLKLIBRARYDEF)))  
(T (LISPXPRT (CONS X (QUOTE (has no EXPR definition)))  
T])

(\* edited (27-JUL-75 . 307))

L])

)

```

(RPAQQ VPROPVAR (NEWPROPOPS (PROPSAVES T)
                             (COMSAVES T)
                             (CLISPROPFLG NIL)))
(RPAQQ NEWPROPOPS (APP ADD NTH DEL REM MEMB INC))
(RPAQ PROPSAVES T)
(RPAQ COMSAVES T)
(RPAQ CLISPROPFLG NIL)
(RPAQQ VPROPGLOBALS (HASHNAMEARRAY PROPSAVES COMSAVES WANTDEFS CLISPROPFLG))
[DECLARE: EVAL@COMPILE
 (ADDOVAR GLOBALVARS HASHNAMEARRAY PROPSAVES COMSAVES WANTDEFS CLISPROPFLG)
]
[RPAQQ VPROPADVICE ([ (SUBST IN HELPDLBLOCK)
                     BEFORE NIL (COND ((EQ X Y)
                                         (RETURN Z]
                               (SAVESET AFTER NIL (SVCHECK NAME VALUE]
 (APPLYMAPC [QUOTE ([ (SUBST IN HELPDLBLOCK)
                     BEFORE NIL (COND ((EQ X Y)
                                         (RETURN Z]
                               (SAVESET AFTER NIL (SVCHECK NAME VALUE]
                               (FUNCTION ADVISE))
 (APPLYMAPC (QUOTE ((MEMOS MEMOS T)
                    (OPENFNS OPENFNS)
                    (BLKFNS BLKFNS)
                    (PROPS NEWPROPS T)))
            (FUNCTION DEFVARTYPE))
 (ADDOVAR PRETTYDEFMACROS [MEMOS X (FNS * (DUMPMEMOS (QUOTE X)
                                                    (DUMPOPEN (QUOTE X)
                                                                (OPENFNS X (DECLARE: DOEVAL@COMPILE (PROP MACRO *
                                                                                               (PROP BLKLIBRARY . X))
                                                                                               (DUMPBLKFNS (QUOTE X)
                                                                                               (BLKFNS * (PROPBLKFNSO (QUOTE X)))
                                                                                               (NONCOMS * (DUMPPROPS (QUOTE X)
                                                                                               (NONCOMS X (COMS . X)))
 (DEFLIST(QUOTE(
 (MEMOS NIL)
 (PROPS NIL)
 (NONCOMS NIL)
 ))(QUOTE PRETTYTYPE))
 (DEFLIST(QUOTE(
 [PROP (((P (X)
            (FGETP X Q))
        (S (X V)
            (PUTP X Q V)))
 (REM FN (X)
            (REMPROP X Q]
 (CC ((CCGETF CCSETF)))
 [HASH (((P (X)
            (GETHASH X N))
        (S (X V)
            (PUTHASH X V N)))
 (CLR FN NIL (CLRHASH N))
 (MAP FN (MAPFN)
            (MAPHASH N (FUNCTION (LAMBDA (MAPY MAPX)
                                (APPLY* MAPFN MAPX MAPY]
 [ELT (((P (X)
            (ELT N X))
        (S (X V)
            (SETA N X V]
 [ELTD (((P (X)
            (ELTD N X))
        (S (X V)
            (SETD N X V]

```





(FILECREATED " 6-JAN-75 19:07:46" &lt;DEUTSCH&gt;VU.;20 16265

previous date: "24-AUG-74 16:23:45" &lt;DEUTSCH&gt;VU.;19)

```

(LISPXPRINT (QUOTE VUCOMS)
  T T)
[RPAQQ VUCOMS ((DECLARE: DOEVAL@COMPILE
  (PROP MACRO PRINTQ PRINQ RTFRM FRPTQ FGETP DREV)
  (ADDVARS (NOLINKFNS TRY)))
  (ENDUMP VU)
  (DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVERS
  (ADDVARS (NLAMA FRPTQ)
  (NLAML TRY RTFRM PRINTQ PRINQ XETQ)]
[DECLARE: DOEVAL@COMPILE
(DEFLIST(QUOTE(
  [PRINTQ (X (NLIST (QUOTE PRINT1)
    (KWOTE (CAR X))
    (CADR X])
  [PRINQ (X (NLIST (QUOTE PRIN1)
    (KWOTE (CAR X))
    (CADR X])
  (RTFRM ((X Y Z)
    (RETFROM (STKPOS (QUOTE X)
      Z)
      Y)))
  [FRPTQ (FRPTZ (SUBPAIR (QUOTE (FRPTN FRPTF))
    (LIST (CAR FRPTZ)
      (CONS (QUOTE PROGN)
        (CDR FRPTZ)))
    (QUOTE (ASSEMBLE NIL (CQ (VAG FRPTN))
      (JUMPLE 1 , FRPTQDONE)
      (PUSHN)
      FRPTQLOOP
      (CQ FRPTF)
      (NREF (SOSLE 0))
      (JUMPA FRPTQLOOP)
      (POPNN 1)
      FRPTQDONE])
  (FGETP ((ATM PROP)
    (ASSEMBLE NIL (CQ ATM)
      (CQ2 PROP)
      FGETPLOOP
      (HLRZ 1 , 0 (1))
      (CAMN 1 , ' NIL)
      (JUMPA FGETPDONE)
      (MOVE 3 , 0 (1))
      (HLRZ 1 , 3)
      (CAIE 2 , 0 (3))
      (JUMPA FGETPLOOP)
      (HRRZ 1 , 0 (1))
      FGETPDONE)))
  [DREV ((X Y)
    (ASSEMBLE NIL (CQ X)
      [C (COND [(NULL (QUOTE Y))
        (QUOTE (ASSEMBLE NIL (HRL 1 , ' NIL]
        (T (QUOTE (ASSEMBLE NIL (CQ2 Y)
          (HRLI 1 , 0 (2))
          DREV1
          (HRRZI 2 , 0 (1))
          (CAMN 2 , ' NIL)
          (JUMPA DREV2)
          (HLR 1 , 0 (2))
          (HLLM 1 , 0 (2))
          (HRLI 1 , 0 (2))
          (JUMPA DREV1)
          DREV2
          (HLRZ 1 , 1]
    ))(QUOTE MACRO))

```

```

(ADDOVAR NOLINKFNS TRY)
]
(RPAQQ VUOPENFNS ((EQCAR T)
  (NEQCAR T)
  (NEQUAL T)
  (ILEQ T)
  (IGEQ T)
  (LEQ T)
  (GEQ T)
  (FTAILP T)))
(RPAQQ VUBLKFNS (PUTP))
(RPAQQ VUFNS
  (PRINW NREML DREMV MASSOC INDEX LEQL REML TRUTH OCCURS TABMIN RPLSEG
  STRSUBST NSTROCC FILLARRAY DELASSOC NTHOCC NTHC1 ADDASSOC DREML
  RPLASSOC /RPLASSOC DREMASSOC /DREMASSOC REMCAR GENLIST XETQ XAR
  XDR ICCAR ERASE EQCAR NEQCAR EQLIST NEQUAL ADDELMT ADDELMT
  PRINT1 PRINQ PRINTQ MOD SIGN GCD LCM ILEQ IGEQ LEQ GEQ IMIN
  IMAX MIN MAX MERGECOPY SORTCOPY I HEADP RTFRM CONSN NCONS
  NCONSN TRY FAIL SLESS REMV1 INTERSECTP APP2 LDONC PUTP /PUTP
  FGTEP FTAILP DREV FRPTQ))
(RPAQQ VUVARS NIL)
(RPAQQ VU/FNS (RPLASSOC DREMASSOC PUTP))
[DECLARE: DOEVAL@COMPILE
(DEFLIST(QUOTE(
  (EQCAR ((X Y)
    (* edited NIL)
    (EQ (CAR (LISTP X))
      Y)))
  (NEQCAR ((X Y)
    (* edited NIL)
    (NEQ (CAR (LISTP X))
      Y)))
  [NEQUAL ((X Y)
    (* edited NIL)
    (NOT (EQUAL X Y)
      Y))
  [ILEQ ((X Y)
    (* edited (12-JUL-74 . 931))
    (NOT (IGREATERP X Y)
      Y))
  [IGEQ ((X Y)
    (* edited (12-JUL-74 . 931))
    (NOT (ILESSP X Y)
      Y))
  [LEQ ((X Y)
    (* edited NIL)
    (NOT (GREATERP X Y)
      Y))
  [GEQ ((X Y)
    (* edited NIL)
    (NOT (LESSP X Y)
      Y))
  [FTAILP ((MACROX MACROY)
    (* edited (12-JUL-74 . 930))
    (ASSEMBLE NIL (CQ MACROX)
      (CQ2 MACROY)
      FTAILPLOOP
      (CAIE 2 , 0 (1))
      (CAMN 2 , ' NIL)
      (JUMPA FTAILPDONE)
      (HLRZ 2 , 0 (2))
      (JUMPA FTAILPLOOP)
      FTAILPDONE
      (MOVE 1 , 2]
    ))(QUOTE MACRO))
]
[DECLARE: DOEVAL@COMPILE
(ADDOVAR BLKLIBRARY PUTP)
]
[DECLARE: DOEVAL@COMPILE
(DEFLIST(QUOTE(
  [PUTP (LAMBDA (X P V)
    (* edited NIL)
    (COND (V (PUT X P V))
      (T (REMPROP X P]
    ))(QUOTE BLKLIBRARYDEF))
]
(DEFINEQ

```

```
(PRINW
[LAMBDA (X K FILE) (* edited NIL)
(COND
  ((IGREATERP (SETQ K (IDIFFERENCE K (NCHARS X)))
    0)
  (SPACES K FILE)))
(PRIN2 X FILE])
```

(HARRAYP  
[λ (A)  
(AND  
(ARRAYP A)  
(MINUSP (OPENR  
(LOC A]))

```
(NREML
[LAMBDA (X L) (* edited NIL)
(COND
  ((NULL L)
  NIL)
  ((FMEMB (CAR L)
    X)
  (NREML X (CDR L)))
  (T (CONSN L (CAR L)
    (NREML X (CDR L])))
```

```
(DREMV
[LAMBDA (X L) (* edited NIL)
(COND
  ((AND L (EQ X (CAR L)))
  (CDR L))
  (T (DREMOVE X L]))
```

```
(MASSOC
[LAMBDA (X L) (* edited NIL)
  (AND (SETQ X (FASSOC X L))
  (FMEMB X L))
```

```
(INDEX
[LAMBDA (X L) (* edited NIL)
  (PROG ((L1 (MEMB X L)))
  (RETURN (AND L1 (DIFFERENCE (LENGTH L)
    (LENGTH (CDR L1))))
```

```
(LEQL
[LAMBDA (X Y L) (* edited NIL)
  (MEMB Y (MEMB X L]))
```

```
(REML
[LAMBDA (X L) (* edited NIL)
(COND
  ((NULL L)
  NIL)
  ((FMEMB (CAR L)
    X)
  (REML X (CDR L)))
  (T (CONS (CAR L)
    (REML X (CDR L))))
```

```
(TRUTH
[LAMBDA (X) (* edited NIL)
  (NEQ X NIL)]
```

```
(OCCURS
[LAMBDA (X Y) (* edited NIL)
  (NTHOCC X Y 1)]
```

```
(TABMIN
[LAMBDA (X Y) (* edited NIL)
  (SPACES (MAX (DIFFERENCE X (POSITION))
    Y))
```

```

(RPLSEG
[LAMBDA (X M N Y)                                (* edited NIL)
 (PROG ((K (ADD1 (IDIFFERENCE N M)))
        (NX (NCHARS X))
        (NY (NCHARS Y)))
        (RETURN (COND
                ((EQ K NY)
                 (RPLSTRING X M Y))
                [(AND (EQ N NX)
                     (NOT (IGREATERP NY K)))
                 (SUBSTRING (RPLSTRING X M Y)
                             1
                             (IPLUS M (SUB1 NY)
                                     (AND (EQ M 1)
                                           (NOT (IGREATERP NY N)))
                                           (SETQ K (ADD1 (IDIFFERENCE N NY)))
                                           (SUBSTRING (RPLSTRING X K Y)
                                                     K -1))
                                     (T (CONCAT (SUBSTRING X 1 (SUB1 M)
                                               Y)
                                               (SUBSTRING X (ADD1 N])
                ]))
        ]))
        ]))
(STRSUBST
[LAMBDA (X Y Z)                                    (* edited NIL)
 (PROG ((L1 (NCHARS X))
        (L2 (NCHARS Y))
        (J 1)
        K)
        LP (COND
            ((SETQ K (STRPOS Y Z J))
             (SETQ Z (RPLSEG Z K (IPLUS K L2 -1)
                             X))
             (SETQ J (IPLUS K L1))
             (GO LP))
            (T (RETURN Z]))
        ]))
(NSTROCC
[LAMBDA (X Y POS)                                  (* edited NIL)
 (PROG ((N 0))
        LP (COND
            ((SETQ POS (STRPOS X Y POS NIL NIL T))
             (SETQ N (ADD1 N))
             (GO LP))
            (T (RETURN N]))
        ]))
(FILLARRAY
[LAMBDA (A N L)                                    (* edited NIL)
 (PROG NIL
        L1 (COND
            ((NLISTP L)
             (RETURN N)))
            (SETA A N (CAR L))
            (SETQ N (ADD1 N))
            (SETQ L (CDR L))
            (GO L1]))
        ]))
(DELASSOC
[LAMBDA (X L)                                      (* edited NIL)
 (AND L (COND
        ((EQUAL X (CAAR L))
         (CDR L))
        (T (CONS (CAR L)
                  (DELASSOC X (CDR L))
                  ]))
        ]))
(NTHOCC
[LAMBDA (X E N)                                    (* edited NIL)
 (EQ (NTHC1 X E N)
     T])

```

```
(NTHC1
[LAMBDA (X E N)
(COND
  [(EQUAL E X)
   (COND
    ((EQ N 1)
     (RTFRM NTHOCC T))
    (T (SUB1 N]
     ((NLISTP E)
      NIL)
     (T (NTHC1 X (CDR E)
              (OR (NTHC1 X (CAR E)
                    N)
                  N]))
      N]))
  (* edited NIL)
```

```
(ADDASSOC
[LAMBDA (X L V)
(COND
  ((NULL L)
   (LIST (CONS X V)))
  ((EQUAL X (CAAR L))
   (CONSN L (CONSN (CAR L)
                   X V)
            (CDR L)))
  (T (CONSN L (CAR L)
              (ADDASSOC X (CDR L)
                        V]))
  (* edited NIL)
```

```
(DREML
[LAMBDA (X L)
(COND
  ((NULL L)
   NIL)
  ((MEMB (CAR L)
         X)
   (DREML X (CDR L)))
  (T (RPLACD L (DREML X (CDR L]))
  (* edited NIL)
```

```
(RPLASSOC
[LAMBDA (X L Y)
(PROG ((A (FASSOC X L)))
  (RETURN (COND
    (A (RPLACD A Y)
        L)
    (T (CONS (CONS X Y)
             L]))
  (* edited NIL)
```

```
(/RPLASSOC
[LAMBDA (X L Y)
(PROG ((A (FASSOC X L)))
  (RETURN (COND
    (A (/RPLACD A Y)
        L)
    (T (CONS (CONS X Y)
             L]))
  (* edited NIL)
```

```
(DREMASSOC
[LAMBDA (X L)
(COND
  ((EQ X (CAAR L))
   (CDR L))
  (T (PROG ((M L))
        LP (COND
            ((NULL M)
             (RETURN NIL))
            ((EQ X (CAADR M))
             (RPLACD M (CDDR M))
             (RETURN L))
            (T (SETQ M (CDR M))
               (GO LP]))
  (* edited NIL)
```

```

(/DREMASSOC
[LAMBDA (X L)                                (* edited NIL)
(COND
  ((EQ X (CAAR L))
   (CDR L))
  (T (PROG ((M L))
        LP (COND
              ((NULL M)
               (RETURN NIL))
              ((EQ X (CAADR M))
               (/RPLACD M (CDDR M))
               (RETURN L))
              (T (SETQ M (CDR M))
                  (GO LP]))))

(REMCAR
[LAMBDA (X)                                (* edited NIL)
(RPLACA X (CADR X))
(RPLACD X (CDDR X))

(GENLIST
[LAMBDA (N I J)                            (* edited NIL)
(OR J (SETQ J 1))
(OR I (SETQ I 0))
(PROG (L)
  LP (COND
      ((MINUSP (SETQ N (SUB1 N)))
       (RETURN L)))
      (SETQ L (CONS (IPLUS I (ITIMES N J))
                    L))
      (GO LP]))

(XETQ
[NLAMBDA (XETX XETY)                       (* edited NIL)
(PROG1 (EVAL XETX)
      (SET XETX (EVAL XETY]))

(XAR
[LAMBDA (X Y)                              (* edited NIL)
(PROG1 (CAR X)
      (RPLACA X Y))

(XDR
[LAMBDA (X Y)                              (* edited NIL)
(PROG1 (CDR X)
      (RPLACD X Y))

(TCCAR
[LAMBDA (X A)                              (* edited NIL)
(RPLACA (RPLACD X (CAR X))
      A)]

(ERASE
[LAMBDA (A)                                (* edited NIL)
(/RPLACD A NIL)
(/RPLACA A (QUOTE NOBIND))
(/PUTD A NIL)]

(EQCAR
[LAMBDA (X Y)                              (* edited NIL)
(EQ (CAR (LISTP X))
   Y)]

(NEQCAR
[LAMBDA (X Y)                              (* edited NIL)
(NEQ (CAR (LISTP X))
   Y)]

```

```
(EQLIST
[LAMBDA (A L) (* edited NIL)
 (AND (LISTP L)
 (EQUAL A (CAR L))
 (NULL (CDR L]))

(NEQUAL
[LAMBDA (X Y) (* edited NIL)
 (NOT (EQUAL X Y))]

(ADELTA
[LAMBDA (X L) (* edited NIL)
 (COND
 ((FMEMB X L)
 L)
 (T (CONS X L]))

(ADDELMT
[LAMBDA (X L) (* edited NIL)
 (COND
 ((MEMBER X L)
 L)
 (T (CONS X L]))

(PRINT1
[LAMBDA (X FILE) (* edited NIL)
 (PRIN1 X FILE)
 (TERPRI FILE)]

(PRINQ
[NLAMBDA (PRINX PRINF) (* edited NIL)
 (PRIN1 PRINX (EVAL PRINF))]

(PRINTQ
[NLAMBDA (PRINX PRINF) (* edited NIL)
 (PRINT1 PRINX (EVAL PRINF))]

(MOD
[LAMBDA (X Y) (* edited NIL)
 (DIFFERENCE X (TIMES (QUOTIENT X Y)
 Y))]

(SIGN
[LAMBDA (X) (* edited NIL)
 (COND
 ((LESSP X 0)
 -1)
 (T 1))]

(GCD
[LAMBDA (X Y) (* edited NIL)
 (COND
 ((EQ X 0)
 Y)
 (T (GCD (REMAINDER Y X)
 X))]

(LCM
[LAMBDA (X Y) (* edited NIL)
 (IQUOTIENT (ITIMES X Y)
 (GCD X Y))]

(ILEQ
[LAMBDA (X Y) (* edited (12-JUL-74 . 931))
 (NOT (IGREATERP X Y))
```

(IGEQ  
[LAMBDA (X Y) (\* edited (12-JUL-74 . 931))  
(NOT (ILESSP X Y])

(LEQ  
[LAMBDA (X Y) (\* edited NIL)  
(NOT (GREATERP X Y])

(GEO  
[LAMBDA (X Y) (\* edited NIL)  
(NOT (LESSP X Y])

(IMIN  
[LAMBDA (X Y) (\* edited (12-JUL-74 . 931))  
(COND  
((IGREATERP X Y)  
Y)  
(T X])

(IMAX  
[LAMBDA (X Y) (\* edited (12-JUL-74 . 931))  
(COND  
((IGREATERP X Y)  
X)  
(T Y])

(MIN  
[LAMBDA (X Y) (\* edited NIL)  
(COND  
((GREATERP X Y)  
Y)  
(T X])

(MAX  
[LAMBDA (X Y) (\* edited NIL)  
(COND  
((GREATERP X Y)  
X)  
(T Y])

(MERGECOPY  
[LAMBDA (A B FN) (\* edited NIL)  
(MERGE (APPEND A)  
(APPEND B)  
FN])

(SORTCOPY  
[LAMBDA (L FN) (\* edited NIL)  
(COND  
((SORTED L FN)  
L)  
(T (SORT (APPEND L)  
FN])

(I  
[LAMBDA (X) (\* edited NIL)  
X])

(HEADP  
[LAMBDA (X Y) (\* edited NIL)  
(OR (NULL X)  
(AND Y (EQUAL (CAR X)  
(CAR Y))  
(HEADP (CDR X)  
(CDR Y])



```

(RTFRM
  [NLAMBDA (RTFN RTVAL RTN)
    (RETFROM (STKPOS RTFN (EVAL RTN))
      (EVAL RTVAL))]
  (* edited NIL)

(CONSN
  [LAMBDA (M X Y)
    (COND
      ((AND (LISTP M)
        (EQ X (CAR M))
        (EQ Y (CDR M)))
        M)
      (T (CONS X Y))]
  (* edited NIL)

(NCONS
  [LAMBDA (X Y)
    (COND
      (X (CONS X Y))
      (T Y))]
  (* edited NIL)

(NCONSN
  [LAMBDA (M X Y)
    (COND
      (X (CONSN M X Y))
      (T Y))]
  (* edited NIL)

(TRY
  [NLAMBDA (TRYEXP)
    (EVAL TRYEXP)
    T]]
  (* edited NIL)
  (* FOR EFFECT ONLY)

(FAIL
  [LAMBDA NIL
    (RTFRM TRY NIL))]
  (* edited NIL)

(SLESS
  [LAMBDA (X Y)
    (COND
      [(NLISTP X)
        (OR (LISTP Y)
          (NOT (ALPHORDER Y X))
          )
        ]
      ((LISTP Y)
        (OR (SLESS (CAR X)
          (CAR Y))
          (AND (EQUAL (CAR X)
            (CAR Y))
            (SLESS (CDR X)
              (CDR Y))
          )
        )
      )
  (* edited NIL)
  (* ALPHORDER (X X) = T !)

(REMV1
  [LAMBDA (X L)
    (LDCONC L (SETQ X (FMEMB X L))
      (CDR X))]
  (* edited NIL)

(INTERSECTP
  [LAMBDA (X Y)
    (PROG ((Z X))
      LP (COND
        ((NLISTP Z)
          (RETURN NIL))
        ((FMEMB (CAR Z)
          Y)
          (RETURN Z)))
        (SETQ Z (CDR Z))
        (GO LP))]
  (* edited (4-AUG-74 . 2331))

```

```

(APP2
[LAMBDA (X Y)                                (* edited NIL)
(COND
  ((IGREATERP (FLENGTH X)
              (FLENGTH Y))
   (APPEND Y X))
  (T (APPEND X Y]))

(LDCONC
[LAMBDA (X Y Z)                              (* edited NIL)
(COND
  (Y (NCONC (LDIFF X Y)
            Z))
  (T (APPEND X Z]))

(PUTP
[LAMBDA (X P V)                              (* edited NIL)
(COND
  (V (PUT X P V))
  (T (REMPROP X P]))

(/PUTP
[LAMBDA (X P V)                              (* edited NIL)
(COND
  (V (/PUT X P V))
  (T (/REMPROP X P]))

(FGETP
[LAMBDA (ATM PROP)                          (* edited (12-JUL-74 . 929))
(ASSEMBLE NIL
  (CQ ATM)
  (CQ2 PROP)
  FGETPLOOP
  (HLRZ 1 , 0 (1))
  (CAMN 1 , ' NIL)
  (JUMPA FGETPDONE)
  (MOVE 3 , 0 (1))
  (HLRZ 1 , 3)
  (CAIE 2 , 0 (3))
  (JUMPA FGETPLOOP)
  (HRRZ 1 , 0 (1))
  FGETPDONE])

(FTAILP
[LAMBDA (MACROX MACROY)                    (* edited (12-JUL-74 . 930))
(ASSEMBLE NIL
  (CQ MACROX)
  (CQ2 MACROY)
  FTAILPLOOP
  (CAIE 2 , 0 (1))
  (CAMN 2 , ' NIL)
  (JUMPA FTAILPDONE)
  (HLRZ 2 , 0 (2))
  (JUMPA FTAILPLOOP)
  FTAILPDONE
  (MOVE 1 , 2]))

(DREV
[LAMBDA (X Y)                              (* edited (24-AUG-74 . 1611))
(ASSEMBLE NIL
  (CQ X)
  (CQ2 Y)
  (HRLI 1 , 0 (2))
  DREV1
  (HRRZI 2 , 0 (1))
  (CAMN 2 , ' NIL)
  (JUMPA DREV2)
  (HLR 1 , 0 (2))
  (HLLM 1 , 0 (2))
  (HRLI 1 , 0 (2))
  (JUMPA DREV1)
  DREV2
  (HLRZ 1 , 1]))

```

```

(FRPTQ
[NLAMBDA FRPTZ
(ASSEMBLE NIL
[CQ (VAG (EVAL (CAR FRPTZ]
(JUMPLE 1 , FRPTQDONE)
(PUSHN)
FRPTQLOOP
(CQ (APPLY (QUOTE PROGN)
(CDR FRPTZ)))
(NREF (SOSLE 0))
(JUMPA FRPTQLOOP)
(POPNN 1)
FRPTQDONE])
)
(MAPC VU/FNS (FUNCTION NEW/FN))
[DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS
(ADDOVAR NLAMA FRPTQ)
(ADDOVAR NLAML TRY RTFRM PRINTQ PRINQ XETQ)
]
[DECLARE: DONTCOPY
(FILEMAP (NIL (3557 16066 (PRINW 3569 . 3754) (NREML 3758 . 3966) (DREMV 3970
. 4118) (MASSOC 4122 . 4241) (INDEX 4245 . 4423) (LEQL 4427 . 4521) (REML
4525 . 4733) (TRUTH 4737 . 4825) (OCCURS 4829 . 4921) (TABMIN 4925 . 5048)
(RPLSEG 5052 . 5639) (STRSUBST 5643 . 5958) (NSTROCC 5962 . 6179) (FILLARRAY
6183 . 6435) (DELASSOC 6439 . 6621) (NTHOCC 6625 . 6725) (NTHC1 6729 . 6990)
(ADDASSOC 6994 . 7257) (DREML 7261 . 7457) (RPLASSOC 7461 . 7665) (/RPLASSOC
7669 . 7875) (DREMASSOC 7879 . 8178) (/DREMASSOC 8182 . 8483) (REMCAR 8487
. 8608) (GENLIST 8612 . 8893) (XETQ 8897 . 9018) (XAR 9022 . 9129) (XDR 9133
. 9240) (TCCAR 9244 . 9356) (ERASE 9360 . 9503) (EQCAR 9507 . 9608) (NEQCAR
9612 . 9716) (EQLIST 9720 . 9850) (NEQUAL 9854 . 9948) (ADDELT 9952 . 10081)
(ADDELMT 10085 . 10216) (PRINT1 10220 . 10331) (PRINQ 10335 . 10437) (PRINTQ
10441 . 10545) (MOD 10549 . 10667) (SIGN 10671 . 10791) (GCD 10795 . 10939)
(LCM 10943 . 11060) (ILEQ 11064 . 11174) (FGEQ 11178 . 11285) (LEQ 11289 .
11383) (GEQ 11387 . 11478) (IMIN 11482 . 11619) (IMAX 11623 . 11760) (MIN
11764 . 11885) (MAX 11889 . 12010) (MERGECOPY 12014 . 12137) (SORTCOPY 12141
. 12293) (I 12297 . 12372) (HEADP 12376 . 12544) (RTFRM 12548 . 12678) (CONSN
12682 . 12856) (NCONS 12860 . 12977) (NCONSN 12981 . 13102) (TRY 13106 . 13252)
(FAIL 13256 . 13347) (SLESS 13351 . 13666) (REMV1 13670 . 13791) (INTERSECTP
13795 . 14061) (APP2 14065 . 14234) (LDCONC 14238 . 14380) (PUTP 14384 . 14512)
(/PUTP 14516 . 14647) (FGETP 14651 . 15018) (FTAILP 15022 . 15348) (DREV 15352
. 15722) (FRPTQ 15726 . 16063))))))
STOP

```

(\* edited (24-AUG-74 . 1611))

(FILECREATED " 7-AUG-75 01:35:43" &lt;DEUTSCH&gt;VMAP.:34 14771

changes to: VMAPCOMS

previous date: " 1-JUN-75 22:35:23" &lt;DEUTSCH&gt;VMAP.:33)

```

(PRETTYCOMPRINT VMAPCOMS)
[RPAQQ VMAPCOMS ((FNS * VMAFFNS)
  (OPENFNS * VMAPOPENFNS)
  (BLKFNS * VMAPBLKFNS)
  (DECLARE: DOEVAL@COMPILE (PROP MACRO MAPPLY1 MAPPLY2))
  (P (MOVD (QUOTE APPLY*)
    (QUOTE MAPPLY1]
(RPAQQ VMAFFNS
  (MAPPLY2 FSORT XXPROD FIRSTV PERMMAP PERM1 PERMROT MAPINT DFILTER
    SORTED MAPPATHS MAPCARU NMAPLISTU MAPLISTU MAPU MAPRPC DMAP
    DISTRIB DDISTRIB CASSOC MAPREDC MAPR MAPCR MAPCARN MAPLISTN
    NMAPCAR NMAPCARN MAPRED MAPROD NMAPROD MAPHASH1))
(DEFINEQ
(MAPPLY2
[LAMBDA (APPLYFN APPLYX APPLYFNX)                (* edited NIL)
  (COND
    ((NULL (OR APPLYFN APPLYFNX))
      (CDR APPLYX))
    (T (APPLY* (OR APPLYFN APPLYFNX)
      APPLYX]))
(FSORT
[LAMBDA (DATA COMPAREFN)                (* edited (2-SEP-74 . 1011))
  [COND
    ((CDR DATA)
      (PROG ((MAPR (CDR DATA))
        (MAPS (FRPLACD DATA NIL))
        MAPA MAPW MAPM)
      L1 (SETQ MAPA (CAR (SETQ MAPW MAPR)))
        (SETQ MAPR (CDR MAPR))
        (SETQ MAPM MAPS)
      L2 (COND
        ((MAPPLY1 COMPAREFN MAPA (CAR MAPM))
          (FRPLNODE2 MAPW MAPM)
          (FRPLNODE MAPM MAPA MAPW))
        ((NULL (CDR MAPM))
          (FRPLACD MAPM (FRPLACD MAPW NIL)))
        (T (SETQ MAPM (CDR MAPM))
          (GO L2)))
      (COND
        (MAPR (GO L1]
    DATA])
(XXPROD
[LAMBDA (L FN)                (* edited NIL)
  (COND
    ((NULL L)
      (LIST NIL))
    (T (MAPROD (MAPPLY1 FN (CAR L))
      (XXPROD (CDR L)
        FN)
      (FUNCTION CONS]))
(FIRSTV
[LAMBDA (FIRSTX FIRSTFN1 FIRSTFN2)        (* edited NIL)
  (PROG (FIRSTY)
    LP (COND
      ((NULL FIRSTX)
        (RETURN NIL))
      ((SETQ FIRSTY (MAPPLY1 FIRSTFN1 (CAR FIRSTX)))
        (RETURN FIRSTY)))
    (SETQ FIRSTX (MAPPLY2 FIRSTFN2 FIRSTX))
    (GO LP])
(PERMMAP
[LAMBDA (PERML PERMFN)                (* edited NIL)
  (PROG ((PERMLAST (LAST PERML)))
    (PERM1 PERML (LENGTH PERML]))

```

```

(PERM1
 [LAMBDA (PERMX PERMN)                                (* edited NIL)
  (COND
   ((ILESSP PERMN 2)
    (MAPPLY1 PERMFN PERML))
   (T (PROG ((PERMJ PERMN))
            L1 (PERM1 (CDR PERMX)
                      (SUB1 PERMN))
            (PERMROT PERMX)
            (COND
             ((ZEROP (SETQ PERMJ (SUB1 PERMJ)))
              (RETURN))
             (T (GO L1])))))

(PERMROT
 [LAMBDA (L)                                           (* edited NIL)
  (PROG ((E (CDR L)))
        [RPLACA E (PROG1 (CAR L)
                          (RPLACA L (CAR E))
        (COND
         ((CDR E)
          (RPLACD L (CDR E))
          (RPLACD PERMLAST (SETQ PERMLAST (RPLACD E NIL]))

(MAPINT
 [LAMBDA (MAPJ MAPK MAPFN1 MAPFN2)                   (* edited NIL)
  (PROG NIL
   LP (COND
      ((IGREATERP MAPJ MAPK)
       (RETURN NIL)))
      (MAPPLY1 MAPFN1 MAPJ)
      (SETQ MAPJ (MAPPLY2 MAPFN2 MAPJ (FUNCTION ADD1)))
      (GO LP])

(DFILTER
 [LAMBDA (DFL DFFN)                                    (* edited NIL)
  (PROG (DFM DFP)
   LP [COND
      ((NULL DFL)
       (RETURN DFM))
      ((MAPPLY1 DFFN (CAR DFL))
       (SETQ DFL (CDR (SETQ DFP DFL)))
       (OR DFM (SETQ DFM DFP)))
      [DFP (FRPLACD DFP (SETQ DFL (CDR DFL))
        (T (SETQ DFL (CDR DFL))
         (GO LP])

(SORTED
 [LAMBDA (L FN)                                        (* edited NIL)
  (PROG NIL
   LP (COND
      ((NULL (CDR L))
       (RETURN T))
      ((APPLY* FN (CAR L)
               (CADR L))
       (SETQ L (CDR L))
       (GO LP))
      (T (RETURN NIL])

(MAPPATHS
 [LAMBDA (PATHX PATHFN1 PATHFN2 PATHL)              (* edited NIL)
  (SETQ PATHL (CONS PATHX PATHL))
  (COND
   ((FMEMB PATHX (CDR PATHL))
    (NLIST (APPLY* PATHFN2 PATHL T)))
   (T (PROG ((PATHZ (APPLY* PATHFN1 PATHX)))
            (RETURN (COND
                     [PATHZ (MAPCONC PATHZ (FUNCTION (LAMBDA (Z)
                                                             (MAPPATHS Z PATHFN1 PATHFN2 PATHL))
                     (T (NLIST (APPLY* PATHFN2 PATHL))

(MAPCARU
 [LAMBDA (MAPX MAPFN1 MAPFN2)                       (* edited NIL)
  (COND
   (MAPX (CONS (MAPPLY1 MAPFN1 (CAR MAPX))
               (MAPCARU (MAPPLY2 MAPFN2 MAPX)
                       MAPFN1 MAPFN2]))

```



```

(CASSOC
[LAMBDA (ASSOCX ASSOCL ASSOCFN1 ASSOCFN2) (* edited NIL)
  (PROG NIL
    LP (COND
      ((NULL ASSOCL)
        (RETURN NIL))
      ((EQUAL ASSOCX (MAPPLY1 ASSOCFN1 ASSOCL))
        (RETURN (CAR ASSOCL)))
      (T (SETQ ASSOCL (MAPPLY2 ASSOCFN2 ASSOCL))
        (GO LP]))

(MAPREDC
[LAMBDA (MAPL MAPFN MAPV) (* edited NIL)
  (COND
    (MAPL (APPLY* MAPFN (CAR MAPL)
      (MAPREDC (CDR MAPL)
        MAPFN MAPV)))
    (T MAPV])

(MAPR
[LAMBDA (MAPX MAPFN1 MAPFN2) (* edited NIL)
  (MAP MAPX MAPFN1 MAPFN2)
  MAPX])

(MAPCR
[LAMBDA (MAPX MAPFN1 MAPFN2) (* edited NIL)
  (MAPC MAPX MAPFN1 MAPFN2)
  MAPX])

(MAPCARN
[LAMBDA (MAPL MAPFN) (* edited NIL)
  (PROG (MAPV MAPW (MAPL1 MAPL)
    MAPL2
    (MAPM MAPL))
    LP (COND
      ((NULL MAPL)
        (RETURN MAPL1)))
      (SETQ MAPW (CAR MAPL))
      (SETQ MAPL (CDR MAPL))
      (COND
        ((EQ (SETQ MAPV (MAPPLY1 MAPFN MAPW))
          MAPW)
          (GO LP)))
      L2 [SETQ MAPW (CONS (CAR MAPM)
        (SETQ MAPM (CDR MAPM))]
        [COND
          ((NULL MAPL2)
            (SETQ MAPL1 (SETQ MAPL2 MAPW)))
          (T (FRPLACD MAPL2 (SETQ MAPL2 MAPW))]
          (COND
            ((NEQ MAPM MAPL)
              (GO L2)))
          (FRPLACA MAPL2 MAPV)
          (GO LP]))

(MAPLISTN
[LAMBDA (MAPL MAPFN) (* edited NIL)
  (PROG (MAPV MAPW (MAPL1 MAPL)
    MAPL2
    (MAPM MAPL))
    LP (COND
      ((NULL MAPL)
        (RETURN MAPL1)))
      (SETQ MAPL (CDR (SETQ MAPW MAPL)))
      (COND
        ((EQ (CAR MAPW)
          (SETQ MAPV (MAPPLY1 MAPFN MAPW)))
          (GO LP)))
      L2 [SETQ MAPW (CONS (CAR MAPM)
        (SETQ MAPM (CDR MAPM))]
        [COND
          ((NULL MAPL2)
            (SETQ MAPL1 (SETQ MAPL2 MAPW)))
          (T (FRPLACD MAPL2 (SETQ MAPL2 MAPW))]
          (COND
            ((NEQ MAPM MAPL)
              (GO L2)))
          (FRPLACA MAPL2 MAPV)
          (GO LP]))

```

```

(NMAPCAR
  [LAMBDA (MAPX MAPFN1 MAPFN2)                                (* edited NIL)
    (COND
      (MAPX (NCONS (MAPPLY1 MAPFN1 (CAR MAPX))
                    (NMAPCAR (MAPPLY2 MAPFN2 MAPX)
                              MAPFN1 MAPFN2]))
    )
)

(NMAPCARN
  [LAMBDA (MAPX MAPFN1 MAPFN2)                                (* edited NIL)
    (COND
      (MAPX (NCONSN MAPX (MAPPLY1 MAPFN1 (CAR MAPX))
                       (NMAPCARN (MAPPLY2 MAPFN2 MAPX)
                                  MAPFN1 MAPFN2]))
    )
)

(MAPRED
  [LAMBDA (MAPL MAPFN MAPV)                                    (* edited NIL)
    (COND
      (MAPL (MAPPLY1 MAPFN MAPL (MAPRED (CDR MAPL)
                                         MAPFN MAPV)))
      (T MAPV])
)

(MAPROD
  [LAMBDA (PRODX PRODY PRODFN)                                (* edited NIL)
    (MAPCONC PRODX (FUNCTION (LAMBDA (PRODU)
                                (MAPCAR PRODY (FUNCTION (LAMBDA (PRODV)
                                                            (APPLY* PRODFN PRODU PRODV)))
                                )
)

(NMAPROD
  [LAMBDA (PRODX PRODY PRODFN)                                (* edited NIL)
    (MAPCONC PRODX (FUNCTION (LAMBDA (PRODU)
                                (NMAPCAR PRODY (FUNCTION (LAMBDA (PRODV)
                                                            (APPLY* PRODFN PRODU PRODV)))
                                )
)

(MAPHASH1
  [LAMBDA (MAPB MAPFN)                                        (* edited (1-JUN-75 . 4272Q))
    [COND
      ((LISTP MAPB)
        (SETQ MAPB (CAR MAPB])
      (PROG (MAPX)
        (ASSEMBLE NIL
          (CQ (VAG (ARRAYSIZE MAPB)))
          (PUSHN)
          (JRST MAP2)
          MAP1(CQ MAPB)
          (NREF (ADD 1 , 0))
          (HRRZ 2 , 2 (1))
          (JUMPE 2 , MAP2)
          (ADDI 1 , 2)
          (SETQ MAPX)
          (CQ (MAPPLY1 MAPFN MAPX))
          MAP2(NREF (SOSL 0))
          (JRST MAP1)
          (POPN)
          (CQ NIL))
        (RETURN])
    )
  [RPAQQ VMAOPENFNS ((MAPINT (MAPK MAPFN1 MAPFN2))
    (DFILTER (DFFN))
    (MAPU (MAPFN1 MAPFN2))
    (MAPRPC (MAPFN1 MAPFN2))
    (DMAP (MAPFN1 MAPFN2))
    (MAPR (MAPFN1 MAPFN2))
    (MAPCR (MAPFN1 MAPFN2))
    (MAPCARN (MAPFN))
    (MAPHASH1 (MAPFN])
  [DECLARE: DOEVAL@COMPILE
  (DEFLIST(QUOTE(
    (MAPINT (($$MAPJ MAPK MAPFN1 MAPFN2)
      ([LAMBDA (MAPJ)
        (* edited NIL)
        (PROG NIL LP (COND ((IGREATERP MAPJ MAPK)
                           (RETURN NIL)))
          (MAPPLY1 MAPFN1 MAPJ)
          (SETQ MAPJ (MAPPLY2 MAPFN2 MAPJ
                              (FUNCTION ADD1)))
          (GO LP]
      $$MAPJ)))

```



```

(DFILTER (($DFL DFFN)
  ([LAMBDA (DFL)
    (* edited NIL)
    (PROG (DFM DFP)
      LP
      [COND ((NULL DFL)
        (RETURN DFM))
        ((MAPPLY1 DFFN (CAR DFL))
        (SETQ DFL (CDR (SETQ DFP DFL)))
        (OR DFM (SETQ DFM DFP)))
        [DFP (FRPLACD DFP (SETQ DFL (CDR DFL))
        (T (SETQ DFL (CDR DFL))
        (GO LP]
      ]
    ]
  ($DFL)))
(MAPU (($MAPX MAPFN1 MAPFN2)
  ([LAMBDA (MAPX)
    (* edited NIL)
    (PROG ((MAPY MAPX))
      LP
      (COND ((NULL MAPY)
        (RETURN MAPX)))
      (MAPPLY1 MAPFN1 MAPY)
      (SETQ MAPY (MAPPLY2 MAPFN2 MAPY))
      (GO LP]
    ($MAPX)))
(MAPRPC (($MAPX MAPFN1 MAPFN2)
  ([LAMBDA (MAPX)
    (* edited NIL)
    (PROG ((MAPY MAPX))
      LP
      (COND ((NLISTP MAPY)
        (RETURN MAPX)))
      (FRPLACA MAPY (MAPPLY1 MAPFN1 (CAR MAPY)))
      (SETQ MAPY (MAPPLY2 MAPFN2 MAPY))
      (GO LP]
    ($MAPX)))
(DMAP (($MAPX MAPFN1 MAPFN2)
  ([LAMBDA (MAPX)
    (* edited NIL)
    (PROG ((MAPY MAPX))
      LP
      (COND ((NULL MAPY)
        (RETURN MAPX)))
      (SETQ MAPY (PROG1 (MAPPLY2 MAPFN2 MAPY)
        (MAPPLY1 MAPFN1 MAPY)))
      (GO LP]
    ($MAPX)))
(MAPR (($MAPX MAPFN1 MAPFN2)
  ([LAMBDA (MAPX)
    (* edited NIL)
    (MAP MAPX MAPFN1 MAPFN2)
    MAPX]
  ($MAPX)))
(MAPCR (($MAPX MAPFN1 MAPFN2)
  ([LAMBDA (MAPX)
    (* edited NIL)
    (MAPC MAPX MAPFN1 MAPFN2)
    MAPX]
  ($MAPX)))

```

```

(MAPCARN (($MAPL MAPFN)
  ([LAMBDA (MAPL)
    (* edited NIL)
    (PROG (MAPV MAPW (MAPL1 MAPL)
      MAPL2
      (MAPM MAPL))
      LP
      (COND ((NULL MAPL)
        (RETURN MAPL1)))
      (SETQ MAPW (CAR MAPL))
      (SETQ MAPL (CDR MAPL))
      (COND ((EQ (SETQ MAPV (MAPPLY1 MAPFN MAPW))
        MAPW)
        (GO LP)))
      L2
      [SETQ MAPW (CONS (CAR MAPM)
        (SETQ MAPM (CDR MAPM)
          [COND ((NULL MAPL2)
            (SETQ MAPL1 (SETQ MAPL2 MAPW)))
            (T (FRPLACD MAPL2 (SETQ MAPL2 MAPW))
              (COND ((NEQ MAPM MAPL)
                (GO L2)))
              (FRPLACA MAPL2 MAPV)
              (GO LP]
          $MAPL)))
    (MAPHASH1 (($MAPB MAPFN)
      ([LAMBDA (MAPB)
        (* edited (1-JUN-75 . 2234))
        [COND ((LISTP MAPB)
          (SETQ MAPB (CAR MAPB)
            (PROG (MAPX)
              (ASSEMBLE NIL (CQ (VAG (ARRAYSIZE MAPB)))
                (PUSHN)
                (JRST MAP2)
                MAP1
                (CQ MAPB)
                (NREF (ADD 1 , 0))
                (HRRZ 2 , 2 (1))
                (JUMPE 2 , MAP2)
                (ADDI 1 , 2)
                (SETQ MAPX)
                (CQ (MAPPLY1 MAPFN MAPX))
                MAP2
                (NREF (SOSL 0))
                (JRST MAP1)
                (POPN)
                (CQ NIL))
              (RETURN]
            $MAPB)))
        ))(QUOTE MACRO))
      ]
      (RPAQQ VMAPBLKFNS (FSORT))
      [DECLARE: DOEVAL@COMPILE
        (ADDTTOVAR BLKLIBRARY FSORT)
      (DEFLIST(QUOTE(
        (FSORT [LAMBDA (DATA COMPAREFN)
          (* edited (2-SEP-74 . 1011))
          [COND ((CDR DATA)
            (PROG ((MAPR (CDR DATA))
              (MAPS (FRPLACD DATA NIL))
              MAPA MAPW MAPM)
              L1
              (SETQ MAPA (CAR (SETQ MAPW MAPR)))
              (SETQ MAPR (CDR MAPR))
              (SETQ MAPM MAPS)
              L2
              (COND ((MAPPLY1 COMPAREFN MAPA (CAR MAPM))
                (FRPLNODE2 MAPW MAPM)
                (FRPLNODE MAPM MAPA MAPW))
                ((NULL (CDR MAPM))
                (FRPLACD MAPM (FRPLACD MAPW NIL)))
                (T (SETQ MAPM (CDR MAPM))
                  (GO L2)))
              (COND (MAPR (GO L1]
          DATA]))
        ))(QUOTE BLKLIBRARYDEF))
      ]
      [DECLARE: DOEVAL@COMPILE

```

```

(DEFLIST(QUOTE(
  [MAPPLY1 (OPENX (COND ((AND (LISTP (CAR OPENX))
                              (EQ (CAAR OPENX)
                                   (QUOTE FUNCTION))
                              (NULL (CDDAR OPENX)))
                    (CONS (CADAR OPENX)
                          (CDR OPENX)))
    (T (CONS (QUOTE APPLY*)
            OPENX]
  [MAPPLY2 (OPENX (PROG [(FN (OR (CAR OPENX)
                                (CADDR OPENX)))
                        (V (CADR OPENX))
                        (FN2 (OR (CADDR OPENX)
                               (QUOTE (FUNCTION CDR]
                        (RETURN (COND ((NULL FN)
                                     (CONS (QUOTE CDR)
                                           (CDR OPENX)))
                                   ((AND (LISTP FN)
                                        (EQ (CAR FN)
                                             (QUOTE FUNCTION))
                                        (NULL (CDDR FN)))
                                    (LIST (CADR FN)
                                          V))
                                   [(LITATOM FN)
                                    (LIST (QUOTE COND)
                                          (LIST FN (LIST (QUOTE APPLY*)
                                                            FN V))
                                          (LIST T (LIST (CADR FN2)
                                                            V]
                                   (T (LIST (QUOTE APPLY*)
                                           (LIST (QUOTE OR)
                                                 FN FN2)
                                           V]
    ))(QUOTE MACRO))
  ]
  (MOVD (QUOTE APPLY*)
        (QUOTE MAPPLY1))
(DECLARE: DONTCOPY
  (FILEMAP (NIL (661 9775 (MAPPLY2 673 . 862) (FSORT 866 . 1429) (XXPROD 1433
. 1635) (FIRSTV 1639 . 1940) (PERMMAP 1944 . 2093) (PERM1 2097 . 2423) (PERMROT
2427 . 2735) (MAPINT 2739 . 3008) (DFILTER 3012 . 3360) (SORTED 3364 . 3618)
(MAPPATHS 3622 . 4019) (MAPCARU 4023 . 4212) (NMAPLISTU 4216 . 4408) (MAPLISTU
4412 . 4600) (MAPU 4604 . 4856) (MAPRPC 4860 . 5137) (DMAP 5141 . 5398) (DISTRIB
5402 . 5797) (DDISTRIB 5801 . 6261) (CASSOC 6265 . 6557) (MAPREDC 6561 . 6749)
(MAPR 6753 . 6864) (MAPCR 6868 . 6981) (MAPCARN 6985 . 7632) (MAPLISTN 7636
. 8268) (NMAPCAR 8272 . 8464) (NMAPCARN 8468 . 8671) (MAPRED 8675 . 8846)
(MAPROD 8850 . 9049) (NMAPROD 9053 . 9255) (MAPHASH1 9259 . 9772))))))
STOP

```

(FILECREATED " 2-NOV-74 13:40:45" VFA.;18 7674

changes to: VFACOMS SMFNLIST

previous date: " 1-NOV-74 11:37:10" VFA.;16)

(LISXPXPRINT (QUOTE VFACOMS)

T T)

(RPAQQ VFACOMS ((DECLARE: DOEVAL@COMPILE  
(PROP MACRO \* (MAPCAR SMFNLIST  
(FUNCTION CAR)))  
(PROP MACRO AEPS DISPATCH))

(PROP NEWPROFORMS ELTS ELTDS ELTNS)

~~(ENDUMP VFA)))~~

[DECLARE: DOEVAL@COMPILE

(DEFLIST(QUOTE(

[ELTS ((A I)

(CAR (AEPS A I]

[ELTDS ((A I)

(CDR (AEPS A I]

(SETAS ((A I X)

(FRPLACA (AEPS A I)

X)))

(SETDS ((A I X)

(FRPLACD (AEPS A I)

X)))

[ELTNS ((A I)

(OPENR (LOC (AEPS A I]

(SETNS ((A I X)

(CLOSER (LOC (AEPS A I))

X)))

(SMGREATERP (OPENX (ORDSM (QUOTE CAIG)

OPENX)))

(SMLESSP (OPENX (ORDSM (QUOTE CAIL)

OPENX)))

(SMGEQ (OPENX (ORDSM (QUOTE CAIGE)

OPENX)))

(SMLEQ (OPENX (ORDSM (QUOTE CAILE)

OPENX)))

(SMPLUS (OPENX (ASMPLUS OPENX)))

[SMINUS ((X)

(ASSEMBLE NIL (CQ X)

(MOVNI 1 , 0 (1))

(ADDI 1 , (ITIMES 2 (LOC 0)

(SMINUSP ((X)

(SMLESSP X 0)))

(SMADD1 ((X)

(SMPLUS X 1)))

(SMSUB1 ((X)

(SMPLUS X -1)))

(SMQUOTIENT (OPENX (ASMQUOTIENT OPENX)))

(SMDIFFERENCE (OPENX (ASMDIFFERENCE OPENX)))

(SMNTH ((X N)

(ASSEMBLE NIL (CQ X)

(CQ2 N)

(SUBI 2 , (LOC 1))

(JUMPLE 2 , FNTHOUT)

FNTHLOOP

(HLRZ 1 , 0 (1))

(SOJG 2 , FNTHLOOP)

FNTHOUT)))

))(QUOTE MACRO))

*fixed*

```

(DEFLIST(QUOTE(
  [AEPS ((A I)
    (ASSEMBLE NIL (CQ I)
      (CQ2 A)
      (ADDI 1 , (IDIFFERENCE 1 (LOC 0))
        (2]
  [DISPATCH (OPENX (COND ((FNTYP (QUOTE DISPCOMP))
    (DISPCOMP OPENX))
    (T (CONS (QUOTE PROG1)
      OPENX]
  ))(QUOTE MACRO))
]
(DEFLIST(QUOTE(
  [ELTS (((P (X)
    (ELTS N X))
    (S (X V)
    (SETAS N X V]
  [ELTDS (((P (X)
    (ELTDS N X))
    (S (X V)
    (SETDS N X V]
  [ELTNS (((P (X)
    (ELTNS N X))
    (S (X V)
    (SETNS N X V]
  ))(QUOTE NEWPROPFORMS))
  (RPAQQ VFAFNS (VFASET INCCONST ORDSM ASPLUS ASMDIFFERENCE ASMQUOTIENT
    DISPATCH DISPCOMP DISPADD))
  (RPAQQ VFAVARS (SMFNLIST))
  [RPAQQ VFABLOCKS ((DISPCOMP DISPCOMP DISPADD (LOCALFREEVARS MIN MAX DLIST DGO]
(DEFINEQ
(VFASET
  [LAMBDA NIL
    (MAPC SMFNLIST (FUNCTION (LAMBDA (X)
      (PUTD (CAR X)
        (GETD (CDR X]))
(INCCONST
  [LAMBDA (AC V)
    (COND
      ((OR (ILESSP V -262143)
        (IGREATERP V 262143))
        (LIST (QUOTE ADD)
          AC
          (QUOTE ,)
          (QUOTE =)
          V))
      ((MINUSP V)
        (LIST (QUOTE SUBI)
          AC
          (QUOTE ,)
          (IMINUS V)))
      (T (LIST (QUOTE ADDI)
        AC
        (QUOTE ,)
        V])

```

(ORDSM

```

[LAMBDA (OP Z)
(NCONC [SUBST (CAR Z)
          (QUOTE X)
          (QUOTE (ASSEMBLE NIL
                  (CQ X]
(COND
 [(SMALLP (CADR Z))
  (LIST (LIST OP 1 (QUOTE ,)
          (LOC (CADR Z]
  (T (SUBPAIR (QUOTE (OP Y))
          (CONS OP (CDR Z))
          [QUOTE ((CQ2 Y)
                  (OP 1 , 0 (2)
          T)))
  (LIST (QUOTE (CQ NIL])

```

(ASMPUS

```

[LAMBDA (OPENX)
(COND
 ((NULL OPENX)
  0)
 ((NULL (CDR OPENX))
  (CAR OPENX))
 (T (PROG ((L (CONS)
            (V 0))
          (TCONC L (QUOTE ASSEMBLE))
          (TCONC L NIL)
          [MAPC OPENX
            (FUNCTION (LAMBDA (X)
                      (COND
 ((NUMBERP X)
 (SETQ V (IPLUS V X)))
 [(CDDAR L)
 [SETQ V (IPLUS V (IMINUS (LOC 0)
                          (LCONC L (SUBST X (QUOTE X)
                                      (QUOTE ((CQ2 X)
                                              (ADD 1 , 2]
                          (T (TCONC L (LIST (QUOTE CQ)
                                          X]
                      (RETURN (COND
 [(CDDAR L)
 (COND
 ((ZEROP V)
 (CAR L))
 (T (CAR (TCONC L (INCONST 1 V)
                  (T V])

```

(ASMDIFFERENCE

```

[LAMBDA (OPENX)
(COND
 [(NUMBERP (CADR OPENX))
  (ASMPUS (LIST (CAR OPENX)
                (IMINUS (CADR OPENX]
  (T (SUBPAIR (QUOTE (X Y))
            OPENX
            (QUOTE (ASSEMBLE NIL
                    (CQ X)
                    (CQ2 Y)
                    (SUBI 1 , 0 (2))
                    (ADDI 1 , (LOC 0])

```

```

(ASMQUOTIENT
[LAMBDA (OPENX)
(NCONC [SUBST (CAR OPENX)
(QUOTE X)
(QUOTE (ASSEMBLE NIL
(CQ X)
(SUBI 1 , (LOC 0]
[SUBST (CADR OPENX)
(QUOTE Y)
(COND
[(AND (SMALLP (CADR OPENX))
(GEQ (CADR OPENX)
0))
(QUOTE ((IDIVI 1 , Y)
(T (QUOTE ((CQ2 Y)
(SUBI 2 , (LOC 0))
(IDIV 1 , 2]
(LIST (QUOTE (ADDI 1 , (LOC 0]))

(DISPATCH
[LAMBDA (X)
X])

(DISPCOMP
[LAMBDA (L)
(COND
[(AND (EQ (CAR (LISTP (CAR L)))
(QUOTE SELECTQ))
(NULL (CDR L)))
(PROG ((Z (CDAR L))
(EXIT (GENSYM))
(DEFAULT (GENSYM))
X R MIN MAX DLIST CLIST XLIST DGO EGO G)
(SETQ DGO (LIST (QUOTE JUMPA)
DEFAULT))
(SETQ EGO (LIST (QUOTE JUMPA)
EXIT))
LP (COND
((CDR (SETQ Z (CDR Z)))
(SETQ X (CAR Z))
(COND
[(SMALLP (CAR X))
(DISPADD (CAR X)
(SETQ G (GENSYM)
[(LISTP (CAR X))
(SETQ G NIL)
[COND
[[SETQ R
(SUBSET (CAR X)
(FUNCTION (LAMBDA (Y)
(NOT (COND
((SMALLP Y)
(DISPADD Y (OR G (SETQ G
(GENSYM)
(SETQ XLIST (TCONC XLIST (CONS R (CDR X)
(COND
((NULL G)
(GO LP]
(T (SETQ XLIST (TCONC XLIST X))
(GO LP)))
(COND
((CAR CLIST)
(TCONC CLIST EGO)))
[SETQ CLIST (LCONC CLIST (LIST G (CONS (QUOTE CQ)
(CDR X)
(GO LP)))

```





```
(DEFLIST(QUOTE(
(VFASET (1-NOV-74 . 1136))
))(QUOTE EDITDATE))
(RPAQQ SMFNLIST ((ELTS . ELT)
(ELTDS . ELTD)
(SETAS . SETA)
(SETDS . SETD)
(ELTNS . ELT)
(SETNS . SETA)
(SMGREATERP . IGREATERP)
(SMLESSP . ILESSP)
(SMGEQ . IGEQ)
(SMLEQ . ILEQ)
(SMPLUS . IPLUS)
(SMINUS . IMINUS)
(SMINUSP . MINUSP)
(SMADD1 . ADD1)
(SMSUB1 . SUB1)
(SMQUOTIENT . IQUOTIENT)
(SMDIFFERENCE . IDIFFERENCE)
(SMNTH . FNTH)))
[DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
(BLOCK: NIL VFASET INCONST ORDSM ASMPPLUS ASMDIFFERENCE ASMQUOTIENT DISPATCH)
(BLOCK: DISPCOMP DISPCOMP DISPADD (LOCALFREEVARS MIN MAX DLIST DGO))
]
(LISPXPRINT (QUOTE (VFASET))
T)
(VFASET)
(DECLARE: DONTCOPY
(FILEMAP (NIL (2250 6916 (VFASET 2262 . 2370) (INCONST 2374 . 2712) (ORDSM
2716 . 3060) (ASMPPLUS 3064 . 3723) (ASMDIFFERENCE 3727 . 4043) (ASMQUOTIENT
4047 . 4478) (DISPATCH 4482 . 4515) (DISPCOMP 4519 . 6483) (DISPADD 6487 .
6913))))))
STOP
```

(FILECREATED " 7-JAN-75 12:21:08" <DEUTSCH>VHASH.;24 8026

changes to: T

previous date: " 3-SEP-74 19:25:10" <DEUTSCH>VHASH.;23)

(LISPXPRINT (QUOTE VHASHCOMS)

T T)

(RPAQQ VHASHCOMS ((P (SETHASHQ HCARRAY 100 2.0)  
(SETHASHQ HCNILARRAY 100 2.0))  
(ENDUMP VHASH)))

(SETHASHQ HCARRAY 100 2.0)  
(SETHASHQ HCNILARRAY 100 2.0)

(RPAQQ VHASHFNS  
(HZAP HCONS HNONL HCONSP HLIST HCOPI HSUBEIS MK1 MK1\* GSTART RMARK  
BITMARK GSWEEP GSWEEPNON GSWEEPNIL GREHASH GPUNT))

(RPAQQ VHASHVARS ((HNLIST NIL)))

(RPAQQ VHASHGLOBALS (~~MARKER# BITTABLE#~~ HCARRAY HCNILARRAY HNLIST ~~HMARKLIST#~~))

(RPAQQ VHASHBLOCKS ((HSUBLIS

(HCOPI)

(RMARK RMARK BITMARK)))

H-MARKTABLE

(RPAQQ VHASHSTATS

(HCN# HCNILNEW# HC# HCNEW# HCADD# HNL# HNLNEW# HCOPI# HCOPI#  
HCOPI# HCOPI# MARK# MARKNL# MARKED# MARKNL# ~~MARK#~~))

(DEFINEQ

(HZAP

[LAMBDA NIL  
(CLRHASH HCNILARRAY)  
(CLRHASH HCARRAY)  
(SETQ HNLIST NIL)]

(\* edited (12-AUG-74 . 2025))

(HCONS

[LAMBDA (X Y)  
(COND  
[(NULL Y)  
(INC HCNIL#)  
(OR (GETHASH X HCNILARRAY)  
(PROG ((E (LIST X))  
(INC HCNILNEW#)  
(PUTHASH X E HCNILARRAY)  
(RETURN E])  
(T (INC HC#)  
(PROG ((D (GETHASH Y HCARRAY))  
E)  
(COND  
((NULL D)  
(INC HCNEW#)  
(PUTHASH Y (LIST (SETQ D (CONS X Y)))  
HCARRAY)  
(RETURN D))  
((SETQ E (FASSOC X D))  
(RETURN E))  
(T (INC HCADD#)  
(FRPLACD D (CONS (SETQ E (CONS X Y))  
(CDR D)))  
(RETURN E])

(\* edited NIL)

(HNONL

[LAMBDA (E)  
(INC HNL#)  
(CAR (COND  
((MEMBER E HNLIST))  
(T (INC HNLNEW#)  
(SETQ HNLIST (CONS E HNLIST]))

(\* edited (12-AUG-74 . 2032))

```

(HCONSP
[LAMBDA (E)
(COND
((NLISTP E)
(OR (LITATOM E)
(SMALLP E)
(FMEMB E HNLIST)))
((NULL (CDR E))
(EQ (GETHASH (CAR E)
HCNILARRAY)
E))
(T (FMEMB E (GETHASH (CDR E)
HCARRAY]))

```

(\* edited (12-AUG-74 . 2033))

```

(HLIST
[LAMBDA K
(PROG (L (N K))
LP (COND
((ZEROP N)
(RETURN L)))
(SETQ L (HCONS (ARG K N)
L))
(SUB1VAR N)
(GO LP])

```

~~(\* edited NIL)~~

```

(HCOPY
[LAMBDA (E)
(INC HCOP#)
(COND
[(NLISTP E)
(COND
((OR (LITATOM E)
(SMALLP E))
(INC HCOPNL#)
E)
(T (HNONL E)
[(NULL (CDR E))
(COND
((EQ (GETHASH (CAR E)
HCNILARRAY)
E)
(INC HCOPNIL#)
E)
(T (HCONS (HCOPY (CAR E))
NIL]
((FMEMB E (GETHASH (CDR E)
HCARRAY))
(INC HCOPN#)
E)
(T (HCONS (HCOPY (CAR E))
(HCOPY (CDR E))

```

(\* edited (12-AUG-74 . 2036))

```

(HSUBLIS
[LAMBDA (LIS X)
(COND
[(NLISTP X)
(COND
((LITATOM X)
(COND
((SETQ LIS (FASSOC X LIS))
(CDR LIS))
(T X)))
((SMALLP X)
X)
(T (HNONL X)
(T (HCONS (HSUBLIS LIS (CAR X))
(AND (SETQ X (CDR X))
(HSUBLIS LIS X))

```

~~(\* edited NIL)~~

(MK1  
[LAMBDA (OP L) (\* edited NIL)  
(HCONS OP (HCOPIY L))

(MK1\*  
[LAMBDA K (\* edited NIL)  
(PROG ((J K)  
L)  
LP [COND  
((EQ J 1)  
(RETURN (HCONS (ARG K 1)  
L])  
(SETQ L (HCONS (HCOPIY (ARG K J))  
L))  
(SUBIVAR J)  
(GO LP))

(GSTART  
[LAMBDA NIL (\* edited (3-SEP-74 . 1923))  
(~~SETQ \*MARKER\* (CONCAT (QUOTE \*MARKER\*))~~)  
(~~SETQ HMARKLIST NIL~~)  
(~~SETQ BITTABLES (ARRAY 256 0 (VAG 0))~~) (BITMARKTABLE)  
HMARKTABLE

(RMARK  
[LAMBDA (E) (\* edited (3-SEP-74 . 1924))  
(PROG NIL  
TOP [INC MARK#]  
[RETURN (COND

[(NLISTP E)  
(INC MARKNL#)  
(OR (LITATOM E)  
(SMALLP E)  
(MEMB E HMARKLIST)) (BITMARK E HMARKTABLE)  
(AND (MEMB E HMARKLIST)  
(SETQ HMARKLIST (CONS E HMARKLIST))

(=EQ (CDR E) \*MARKER\*) (BITMARK E HMARKTABLE)  
(INC MARKED#)  
[(NULL (CDR E))  
(COND

((EQ (GETHASH (CAR E)  
NIL) HCNILARRAY) (INC MARKNIL#)  
(FRPLACD E \*MARKER\*)  
(SETQ E (CAR E))  
(GO TOP))  
(T (GO NOT))

(MEMB E (GETHASH (CDR E)  
HCARRAY)) (T (RMARK (CAR E))  
[SETQ E (PROG) (CDR E)) (SETQ E (CDR E))  
(FRPLACD E \*MARKER\*) (GO TOP))  
(RMARK (CAR E))  
(GO TOP))  
(T (GO NOT))

(COND  
(BITMARK E) (\* A list cell not created by  
(RETURN (INC MARKI#) HCONS.)  
(INC MARKNIL#)  
(RMARK (CAR E))  
(SETQ E (CDR E))  
(GO TOP))

(BITMARKTABLE

[NIL

(ARRAY 256 0 (VAG 0))

(BITMARK  
[LAMBDA (A)  
(ASSEMBLE NIL

(\* edited (3-SEP-74 . 1922))

(CQ A)  
(MOVE 2 , 1)  
(ANDI 2 , 777Q)  
(IDIVI 2 , 44Q)  
(MOVEI 4 , 1)  
(ROT 4 , 0 (3))  
(PUSHN 4)  
(PUSHN 2)  
(LSHC 1 , -12Q)  
(HRLI 1 , 222202Q)  
(SKIPGE 2)  
(HRLI 1 , 2202Q)  
(ADDI 1 , 2)

✓ ~~(PUSHN)~~ ← BITA

✓ BITMARKP sim

✓ ~~(CQ2 BITTABLES)~~ ← BITA

(NREF (LDB 1 , 0))  
(JUMPN 1 , OLD)  
(CQ (ARRAY 17Q 17Q))  
(NREF (DPB 1 , 0))  
OLD (POPNN 1)  
(POPNN 2)  
(ADD 1 , 2)  
(POPNN 2)  
(TDNE 2 , 2 (1))  
(JRST ON)  
(IORM 2 , 2 (1))  
(CQ NIL)

ON])

(GSWEEP  
[LAMBDA NIL

(\* edited (3-SEP-74 . 1924))

~~(SETQ BITTABLES NIL)~~

← (FILTER HNLIST (F/X (BITMARKP X HMARKTABLE)))

(PROG [LIST (CONS (QUOTE HCARRAY)  
(GSWEEPNON HCARRAY))

(CONS (QUOTE HCNILARRAY)  
(GSWEEPNIL HCNILARRAY]) ← (SETQ HMARKTABLE NIL]

(GSWEEPNON  
[LAMBDA (A)  
(PROG [(M 0)  
(N 0)

(\* edited (2-SEP-74 . 1150))

(S (ARRAYSIZE (CAR A)  
[MAPHASH1 A (FUNCTION (LAMBDA (P)  
(PROG ((Y (CAR P))  
(X (CDR P))  
LAST)  
(SETQ M (ADD1 M))

LP [COND  
[~~(EQ (CDR Y) \*MARKER\*)~~ ← (BITMARKP (CAR Y) HMARKTABLE)  
~~(FRPLACD (CAR Y)  
X)~~

(COND  
((SETQ Y (CDR (SETQ LAST Y)))  
(GO LP])  
[LAST (COND  
[[SETQ Y (CDR (FRPLACD LAST  
(CDR Y))

(GO LP])  
((CDR Y)  
(FRPLNODE2 Y (CDR Y))  
(GO LP])  
(T (RETURN (PUTHASH X NIL A)  
(SETQ N (ADD1 N])

(GREHASH A N)  
(RETURN (LIST N M S])

```
(GSWEEPNIL
[LAMBDA (A) (* edited (2-SEP-74 . 1148))
  (PROG [(M 0)
         (N 0)
         (S (ARRAYSIZE (CAR A)
                    [MAPHASH1 A (FUNCTION (LAMBDA (P)
                                           (PROG ((Y (CAR P)))
                                                  (SETQ M (ADD1 M))
                                                  (COND
                                                    ((CDR Y)
                                                     (FRPLACD Y NIL) (BITMARKP Y HMARKTABLE)
                                                     (SETQ N (ADD1 N)))
                                                    (T (PUTHASH (CDR P)
                                                                NIL A]
                    (GREHASH A N)
  (RETURN (LIST N M S])
```

```
(GREHASH
[LAMBDA (Z N) (* edited (2-SEP-74 . 1147))
[COND
  ((ILESSP N (IQUOTIENT (ARRAYSIZE (CAR Z)
                          3))
   (PROG [(NEW (CONS (HARRAY (ITIMES N 2))
                     (CDR Z]
          (REHASH Z NEW)
          (FRPLACA Z (CAR NEW]
  Z])
```

```
(GPUNT
[LAMBDA NIL (* edited (3-SEP-74 . 1925))
```

```

(SETQ HMARKLIST NIL)
(SETQ BITTABLES NIL)
(MAPHASH1 HCNILARRAY (FUNCTION (LAMBDA (P)
  (FRPLACD (CAR P)
           NIL]
(MAPHASH1 HCARRAY (FUNCTION (LAMBDA (P)
  (PROG ((Y (CAR P))
        (X (CDR P)))
        LP (FRPLACD (CAR Y)
                    X)
  (COND
    ((SETQ Y (CDR Y))
     (GO LP])
)


```

```

)
(RPAQ HNLIST NIL)
[DECLARE: EVAL@COMPILE
  (ADDTOVAR GLOBALVARS *MARKER* BITTABLES HCARRAY HCNILARRAY HNLIST HMARKLIST)
]
[DECLARE: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY
  (BLOCK: NIL HZAP HCONS HNONL HCONSP HLIST MK1 MK1* GSTART GSWEEP GSWEEPNON
  GSWEEPNIL GREHASH GPUNT)
  (BLOCK: HSUBLIS)
  (BLOCK: HCOPY)
  (BLOCK: RMARK RMARK BITMARK)
]
(CS VHASHSTATS)
[DECLARE: DONTCOPY
  (FILEMAP (NIL (864 7623 (HZAP 876 . 1033) (HCONS 1037 . 1629) (HNONL 1633
  . 1835) (HCONSP 1839 . 2144) (HLIST 2148 . 2382) (HCOPY 2386 . 2892) (HSUBLIS
  2896 . 3231) (MK1 3235 . 3329) (MK1* 3333 . 3602) (GSTART 3606 . 3807) (RMARK
  3811 . 4823) (BITMARK 4827 . 5551) (GSWEEP 5555 . 5809) (GSWEEPNON 5813 .
  6512) (GSWEEPNIL 6516 . 6936) (GREHASH 6940 . 7215) (GPUNT 7219 . 7620))))))
STOP
```

(FILECREATED "19-MAR-75 12:18:18" <DEUTSCH>NCTX.;5 17916

changes to: NCTXSET CLRCTX CLRCTX1 GROWARRAY GENCTX NCTXBLOCKS NCTXGLOBALS  
NCTXOPENFNS DISPLAYCTX UNDOCX REDOCX SWAPVALX GCCTX PRUNEX GETX1 CTSIZEZ BPCTX  
PHX GETX SLOWGETX MAPPROP CSONS SETCSONS KILLCTX SONCTX NCTXPROPS CI BINDCTX  
NCTXCOMS PUTX SLOWPUTX UNPUTX NCTXSTATS

previous date: "19-MAR-75 10:25:46" <DEUTSCH>NCTX.;1)

Renamed VCTX

```

(LISPXPRT (QUOTE NCTXCOMS)
  T T)
(RPAQQ NCTXCOMS ((ENDUMP NCTX))
(RPAQQ NCTXSTATS
  (CDISP# CDISP#C CDISP#S CDISP#E GETX# SLOWGETX# GETXUP# GETXDOWN#
  GETXNIL# PUTX# SLOWPUTX# PUTXDOWN# PUTXNIL# UNPUTX#))
(RPAQQ NCTXOPENFNS ((CSONS T)
  (GETCSONS T)
  (PHX T)
  (DISPEND T)
  (CURCTX T)
  (SETCTX T)
  (RESETCTX T)
  (ROOTCTX T)))
[RPAQQ NCTXPROPS ((LIST (MACRO SETMACRO)
  ELTS CONTEXTS)
  (CSONL (MACRO SETMACRO)
  ELTDS CONTEXTS)
  ((CDV CDPREV CDLEV . CDNEXT)
  (MACRO SETMACRO)
  CCELL
  (CDV CDPREV CDNEXT CDLEV))
  (((CSAVES CLEVP . CIDX))
  (MACRO SETMACRO)
  ((CSPROP . CSVAL)
  (MACRO SETMACRO)
  CCELL
  (CSPROP CSVAL))
  (RPAQQ NCTXFNS
  (MAPPROP CLRPROP NCTXSET INITCTX CLRCTX CLRCTX1 EXPANDCTX GROWARRAY
  GENCTX CSONS SETCSONS DISPLAYCTX DISPEND UNDOCX REDOCX
  SWAPVALX PHX KILLCTX SONCTX GCCTX GETX.SLOWGETX GETXCTX PUTX
  SLOWPUTX /PUTX ADDX /ADDX REMX /REMX UNPUTX /UNPUTX CURCTX
  SETCTX RESETCTX ROOTCTX CXP CI CN DMAPCL CCONC KILLTREE
  KILLSONS))
(RPAQQ NCTXGLOBALS (PROPERTIES CONTEXTS NEXTCONTEXT MAXCONTEXT CURRENTCONTEXT
  ROOTCONTEXT GETXCONTEXT DISPCONTEXT DISPEND
  CDISPUSED))
(RPAQQ NCTXBLOCKS ((DISPLAYCTX))
(RPAQQ NCTX/FNS (PUTX ADDX REMX UNPUTX))
(CS NCTXSTATS)
[DECLARE: EVAL@COMPILE
  (ADDTOVAR GLOBALVARS UNPUTX# PUTXNIL# PUTXDOWN# SLOWPUTX# PUTX# GETXNIL#
  GETXDOWN# GETXUP# SLOWGETX# GETX# CDISP#E CDISP#S CDISP#C CDISP#)
]
[DECLARE: DOEVAL@COMPILE
  (DEFLIST(QUOTE(
  [CSONS ((CTX)
    (* edited (19-MAR-75 . 1116))
    (CSONL (CIDX CTX)
  (SETCSONS ((CTX L)
    (* edited (19-MAR-75 . 1116))
    (SETCSONL (CIDX CTX)
  L))))
  (PHX ((P)
    (* edited (19-MAR-75 . 1044))
    (* Change to TOPVAL later)
    (CAR P)))

```

Handwritten checkmarks and arrows pointing to the first few lines of code.

Handwritten arrow pointing to the (SETCTX T) line.

Handwritten checkmark pointing to the NCTXPROPS section.

Handwritten checkmarks and arrows pointing to the CSONL and CTXREC lines.

Handwritten arrow pointing to the NCTXGLOBALS section.

Handwritten checkmarks and arrows pointing to the NCTXBLOCKS and NCTX/FNS sections.

DEADCONTEXT  
FREECONTEXT

```
[DISPEND ((CTX)
  (* edited (18-JUN-74 . 107))
  (SETQ CDISPUSED (CDLEV (SETQ CDISPEND (CLEVP (SETQ DISPCONTEXT CTX)
(CURCTX [LAMBDA NIL CURRENTCONTEXT])
(SETCTX ((CTX)
  (SETQ CURRENTCONTEXT CTX)))
[RESETCTX ((CTX)
  (* edited (18-MAR-75 . 1950))
  (PROG1 CURRENTCONTEXT (SETCTX CTX)
(ROOTCTX [LAMBDA NIL (* edited (31-MAY-74 . 113))
  ROOTCONTEXT])
))(QUOTE MACRO))
]
(DEFINEQ
(CLIST
  [LAMBDA (X)
    (ELTS CONTEXTS X)]
(SETCLIST
  [LAMBDA (X V)
    (SETAS CONTEXTS X V)]
(CSONL
  [LAMBDA (X)
    (ELTDS CONTEXTS X)]
(SETCSONL
  [LAMBDA (X V)
    (SETDS CONTEXTS X V)]
(CDV
  [LAMBDA (X)
    (CAR X)]
(SETCDV
  [LAMBDA (X V)
    (FRPLACA X V)]
(CDPREV
  [LAMBDA (X)
    (CADR X)]
(SETCDPREV
  [LAMBDA (X V)
    (FRPLACA (CDR X)
      V)]
(CDLEV
  [LAMBDA (X)
    (CADDR X)]
(SETCDLEV
  [LAMBDA (X V)
    (FRPLACA (CDDR X)
      V)]
(CDNEXT
  [LAMBDA (X)
    (CDDDR X)]
(SETCDNEXT
  [LAMBDA (X V)
    (FRPLACD (CDDR X)
      V)]
(CDCELL
  [LAMBDA (CDV CDPREV CDNEXT CDLEV)
    (CONS CDV (CONS CDPREV (CONS CDLEV CDNEXT))])
```



```

(CSAVES
  [LAMBDA (X)
    (CAAR X)]

(SETCSAVES
  [LAMBDA (X V)
    (FRPLACA (CAR X)
      V)]

(CLEVP
  [LAMBDA (X)
    (CADAR X)]

(SETCLEVP
  [LAMBDA (X V)
    (FRPLACA (CDAR X)
      V)]

(CIDX
  [LAMBDA (X)
    (CDDAR X)]

(SETCIDX
  [LAMBDA (X V)
    (FRPLACD (CDAR X)
      V)]

(CSPROP
  [LAMBDA (X)
    (CAR X)]

(SETCSPROP
  [LAMBDA (X V)
    (FRPLACA X V)]

(CSVAL
  [LAMBDA (X)
    (CDR X)]

(SETCSVAL
  [LAMBDA (X V)
    (FRPLACD X V)]

(CSCELL
  [LAMBDA (CSPROP CSVAL)
    (CONS CSPROP CSVAL)]
)
(DEFLIST(QUOTE(
  (CLIST (CLIST SETCLIST))
  (CSOVL (CSOVL SETCSOVL))
  (CDV (CDV SETCDV))
  (CDPREV (CDPREV SETCDPREV))
  (CDLEV (CDLEV SETCDLEV))
  (CDNEXT (CDNEXT SETCDNEXT))
  (CSAVES (CSAVES SETCSAVES))
  (CLEVP (CLEVP SETCLEVP))
  (CIDX (CIDX SETCIDX))
  (CSPROP (CSPROP SETCSPROP))
  (CSVAL (CSVAL SETCSVAL))
)) (QUOTE NEWPROP))
[DECLARE: DOEVAL@COMPILE
(DEFLIST(QUOTE(
  (CLIST ((X)
    (ELTS CONTEXTS X)))
  (SETCLIST ((X V)
    (SETAS CONTEXTS X V)))
  (CSOVL ((X)
    (ELTDS CONTEXTS X)))
  (SETCSOVL ((X V)
    (SETDS CONTEXTS X V)))

```

```

(CDV ((X
      (CAR X)))
(SETCDV ((X V)
         (FRPLACA X V)))
(CDPREV ((X
          (CADR X)))
(SETCDPREV ((X V)
            (FRPLACA (CDR X)
                     V)))
(CDLEV ((X
        (CADDR X)))
(SETCDLEV ((X V)
           (FRPLACA (CDDR X)
                    V)))
(CDNEXT ((X
         (CDDDR X)))
(SETCDNEXT ((X V)
            (FRPLACD (CDDR X)
                     V)))
(CSAVES ((X
         (CAAR X)))
(SETCSAVES ((X V)
            (FRPLACA (CAR X)
                     V)))
(CLEVP ((X
        (CADAR X)))
(SETCLEVP ((X V)
           (FRPLACA (CDAR X)
                    V)))
(CIDX ((X
       (CDDAR X)))
(SETCIDX ((X V)
          (FRPLACD (CDAR X)
                   V)))
(CSPROP ((X
        (CAR X)))
(SETCSPROP ((X V)
            (FRPLACA X V)))
(CSVAL ((X
       (CDR X)))
(SETCSVAL ((X V)
           (FRPLACD X V)))
)) (QUOTE MACRO)
]
(/ADDPROP (QUOTE ELTS)
          (QUOTE NEWPROPS)
          (QUOTE (CLIST (MACRO SETMACRO)
                       CONTEXTS))
          T)
(/ADDPROP (QUOTE ELTDS)
          (QUOTE NEWPROPS)
          (QUOTE (CSONL (MACRO SETMACRO)
                       CONTEXTS))
          T)
(/ADDPROP (QUOTE CC)
          (QUOTE NEWPROPS)
          (QUOTE (CDV (MACRO SETMACRO)
                     2))
          T)
(/ADDPROP (QUOTE CC)
          (QUOTE NEWPROPS)
          (QUOTE (CDPREV (MACRO SETMACRO)
                       6))
          T)
(/ADDPROP (QUOTE CC)
          (QUOTE NEWPROPS)
          (QUOTE (CDLEV (MACRO SETMACRO)
                     14))
          T)
T)

```

```

(/ADDPROP (QUOTE CC)
  (QUOTE NEWPROPS)
  (QUOTE (CDNEXT (MACRO SETMACRO)
            15))
T)
(/ADDPROP (QUOTE CC)
  (QUOTE NEWPROPS)
  (QUOTE (CSAVES (MACRO SETMACRO)
            4))
T)
(/ADDPROP (QUOTE CC)
  (QUOTE NEWPROPS)
  (QUOTE (CLEVP (MACRO SETMACRO)
            10))
T)
(/ADDPROP (QUOTE CC)
  (QUOTE NEWPROPS)
  (QUOTE (CIDX (MACRO SETMACRO)
            11))
T)
(/ADDPROP (QUOTE CC)
  (QUOTE NEWPROPS)
  (QUOTE (CSPROP (MACRO SETMACRO)
            2))
T)
(/ADDPROP (QUOTE CC)
  (QUOTE NEWPROPS)
  (QUOTE (CSVAL (MACRO SETMACRO)
            3))
T)

```

(DEFINEQ

(MAPPROP

[LAMBDA (MPROP MCTX MFN)

(\* edited (19-MAR-75 . 1113))

```

(COND
  (EQ (OR MCTX (SETQ MCTX (CURCTX)))
    DISPCONTEXT)

```

(OR \* (DISPLAYCTX MCTX))

(MAPHASH (PHX MPROP)

(FUNCTION (LAMBDA (MP)

(MY MX)

[APPLY\* MFN (CAR MP)

(MX MY

```

(T (RESETEFORM (DISPLAYCTX MCTX))
  ((MAPPROP MPROP MCTX MFN)

```

(OR (EQ MCTX DISPCONTEXT) (DISPLAYCTX MCTX])

(INITPROP

[λ (P)

(COND

((LITATOM (PHX P))

(SETQ PROPERTIES (ADDEL P PROPERTIES))

(CLRPROP

[LAMBDA (P CTX)

(MAPPROP P CTX (FUNCTION (LAMBDA (X) (UNPUTX X P CTX]))

(NCTXSET

[LAMBDA NIL (INITCTX)]

(\* edited (19-MAR-75 . 1025))

(SETHASH P SØ 2.Ø)]

(INITCTX

[LAMBDA NIL

(\* edited (18-JUN-74 . 109))

~~(SETQ CONTEXTS (ARRAY (SETQ MAXCONTEXT 100)))~~

~~(SETQ ROOTCTX (SETQ ROOTCONTEXT (LIST (CONS 1 (CDCELL NIL NIL NIL 1))~~

(SETCDV (CLEVP ROOTCONTEXT)

ROOTCONTEXT)

(SETQ PROPERTIES NIL)

(CLRCTX)]

CTXREC

(CLRCTX

[LAMBDA NIL

(\* edited (19-MAR-75 . 1031))

[MAPC PROPERTIES (FUNCTION (LAMBDA (X)

(CLRHASH (CAR X)

PHX

(DISPEND (SETCTX (ROOTCTX)))

(SETCONS 1 NIL)

(SETQ NEXTCONTEXT NIL

(CLRCTX 2 MAXCONTEXT))

(SETCONS (ROOTCTX) NIL)

(SETCDNEXT CDISPEND NIL)

(SETQ DEADCONTEXT NIL)

(SETQ FREECONTEXT (CTXREC 2))

```

(CLRCTX1
[LAMBDA (M N) (* edited (19-MAR-75 . 1031))
(PROG ((X N))
L1 (COND
((SMLESSP X M)
(RETURN NIL)))
(SETCLIST X NIL)
(SETCSO NL X NEXTCONTEXT)
(SETQ NEXTCONTEXT X)
(SUB1VAR X)
(GO L1))

(EXPANDCTX
[LAMBDA (N)
(SETQ CONTEXTS (GROWARRAY CONTEXTS N))
(CLRCTX1 (ADD1 MAXCONTEXT)
N)
(SETQ MAXCONTEXT N)]

(GROWARRAY
[LAMBDA (A N) (* edited (19-MAR-75 . 1031))
(PROG ((M (ARRAYSIZE A))
(A1 (ARRAY N)))
(ASSEMBLE NIL
(CQ (VAG M))
(PUSHN)
(CQ A1)
(ADD1 1 , 2)
(CQ2 A)
(HRLI 1 , 2 (2))
(POPN 2)
(ADD 2 , 1)
(BLT 1 , -1 (2)))
(RETURN A1])

```

(FINDCN  
[λ (I CTX)  
(COND  
((EQP I (CIDX CTX))  
CTX)  
(T (CAR (SOME  
(CSONL CTX)  
(F/X (FINDCN I X])

```

(GENCTX
[LAMBDA (CTX) (INC GENCTX#) (FREECONTEXT) (* edited (19-MAR-75 . 1032))
(PROG ((C NEXTCONTEXT)
[ (L (CSONL (OR CTX (SETQ CTX (CURCTX))
(LEVP (CLEVP CTX))
[COND
((NULL (CDNEXT LEVP))
(SETCDNEXT LEVP (CDCELL NIL LEVP NIL (ADD1 (CDLEV LEVP)
[COND
((NULL B)
(EXPANDCTX (ITIMES 2 MAXCONTEXT))
(SETQ B NEXTCONTEXT)))
(SETQ NEXTCONTEXT (CSONL B))
(SETCSO NL B NIL)
(SETCLIST B (SETQ C (CONS (CONS B (CDNEXT LEVP))
L CTX)))
(SETCSO NL CTX (CONS C L))
(RETURN C])

```

(SETCLEVP C (CDNEXT LEVP))  
[SETQ FREECONTEXT (OR (CTAIL C)  
CTXREC (ADD1 (CIDX C])  
(SETCTAIL C CTX)

```

(CSONS
[LAMBDA (CTX) (* edited (19-MAR-75 . 1116))
(CSONL (CIDX CTX])

```

```

(SETCSONS
[LAMBDA (CTX L) (* edited (19-MAR-75 . 1116))
(SETCSO NL (CIDX CTX]
L])

```

```

(DISPLAYCTX
[LAMBDA (CTX) (* edited (19-MAR-75 . 1037))
(PROG ((LEVP (CLEVP CTX))
(C CTX)
(OLD DISPCONTEXT))

```

(\* The following error check prevents DISPLAYCTX from smashing the world if CTX is not a valid context.)

(OR (CTXP C)  
(HELP (QUOTE Bad% context:  
C))

(INC CDISP#)  
(COND  
((SMGEQ (CDLEV LEVP)  
CDISPUSED)

L1 (COND  
((NEQ CDISPEND LEVP) (SWAPVALX (CDV CDISPEND))  
(SETCDV CDISPEND NIL) (\* Undo)  
(SETQ CDISPEND (CDPREV CDISPEND))  
(INC CDISP#C)  
(GO L1)))

L3 (DISPEND C)  
(COND  
((NEQ (CDV LEVP)  
C) (SWAPVALX (CDV LEVP)) (\* Undo)  
(SETCDV LEVP C)  
(SETQ LEVP (CDPREV LEVP))  
(INC CDISP#S)  
(SETQ C (CDR C)) <UP  
(GO L3)))

(UNDOCX OLD C)  
(REDOCX CTX C) <UP  
(INC CDISP#E (SUB1 (CDLEV LEVP)))  
(RETURN OLD])  
DOWN (COND  
((NEQ C CTX)  
[SETQ C (CDV (SETQ LEVP  
(CDNEXT LEVP])  
(SWAPVALX C) (\* Redo)  
(GO DOWN)))

(DISPEND  
[LAMBDA (CTX) (\* edited (18-JUN-74 . 107))  
(SETQ CDISPUSED (CDLEV (SETQ CDISPEND (CLEVP (SETQ DISPCONTEXT CTX]))

~~(UNDOCX  
[LAMBDA (CFROM CTO) (\* edited (19-MAR-75 . 1036))  
(COND  
((NEQ CFROM CTO)  
(SWAPVALX CFROM)  
(UNDOCX (CDR CFROM)  
CTO])  
(REDOCX  
[LAMBDA (CTO CFROM) (\* edited (19-MAR-75 . 1036))  
(COND  
((NEQ CTO CFROM)  
(REDOCX (CDR CTO)  
CFROM)  
(SWAPVALX CTO])~~

(SWAPVALX  
[LAMBDA (CTX) (\* edited (19-MAR-75 . 1035))  
(MAPC (CSAVES CTX)

(FUNCTION (LAMBDA (Y)  
(PROG ((E (CAR Y)))  
(MAPC (CDR Y)  
(FUNCTION (LAMBDA (X)  
(PROG [(A (PHX (CSPROP X]) (INC SWAPX#)  
(SETCSVAL X (PROG1 (GETHASH E A)  
(PUTHASH E (CSVAL X)  
A]))

(PHX  
[LAMBDA (P) (\* edited (19-MAR-75 . 1044))  
(CAR P)) (\* Change to TOPVAL later)

```

(KILLCTX
[LAMBDA (CTX)
  (OR (AND [CTXP (OR CTX (SETQ CTX (CURCTX)
                (CDR CTX))
        (HELP))
      (PROG ((C (CDR CTX)))
            (COND
             ((NULL (CSONL CTX))
              (COND
               ((EQ CTX (CURCTX))
                (SETQ CTX C))
              (SETCSONL C (DREMV CTX (CSONL C)))
              (GETCLIST (CDBX CTX)
                       (NIL))
              (T (HELP]))
        (COND
         ((EQ CTX DISPCONTEXT)
          (DISPLAYCTX C))
         ((SETCLEVP CTX NIL)
          (SETCSAVES CTX NIL)
          (COND
           (IMMED (SETCTAIL CTX FREECONTEXT)
            (SETQ FREECONTEXT CTX))
           (T (SETCTAIL CTX DEADCONTEXT)
            (SETQ DEADCONTEXT CTX))))
        (* edited (19-MAR-75 . 1116))

```

```

(SONCTX
[LAMBDA (CTX)
  (AND (CLIST (SETQ CTX (CDBX (OR CTX (CURCTX)
                                   (CSONL CTX)))
        [CTXP (OR CTX (SETQ CTX (CURCTX]
        (* edited (19-MAR-75 . 1116))

```

```

(GCCTX
[LAMBDA NIL
  (PROG ((J MAXCONTEXT))
        (SETQ NEXTCONTEXT NIL)
        L1 (COND
            ((NULL (CLIST J))
             (SETCSONL J NEXTCONTEXT)
             (SETQ NEXTCONTEXT J))
            (SUB1VAR J)
            (COND
             ((NEQ J 1)
              (GO L1))
            (SETQ FREECONTEXT (NCONC DEADCONTEXT
                                       FREECONTEXT))
            (SETQ DEADCONTEXT NIL)
        (* edited (19-MAR-75 . 1039))

```

```

(GETX
[LAMBDA (E P CTX)
  (INC GETX#)
  (COND
   ((EQ (OR CTX (SETQ CTX (CURCTX)))
        DISPCONTEXT)
    (GETHASH E (PHX P)))
   (T (INC SLOWGETX#)
    (SLOWGETX E P CTX]))
  ((EQ CTX (CURCTX)) (INC DISPGETX#)
   (DISPLAYCTX CTX)
   (GETHASH E (PHX P)))
  (* edited (19-MAR-75 . 1046))

```

```

(SLOWGETX
[LAMBDA (E P CTX)
  (PROG ((C CTX)
        (C D)
        UP [COND
            ((NEQ (CDV (CLEVP C))
                 C)
             (INC GETXUP#)
             (COND
              ([AND (SETQ D (FASSOC E (CSAVES C)))
                   (SETQ D (FASSOC P (CDR D))
                    (RETURN (CDR D)))
              (T (SETQ C (CDR C))
               (GO UP])
             (SETQ GETXCONTEXT C)
            (SETQ C1 C)
            DOWN (INC GETXDOWN#)
            (COND
             ([AND (SETQ D (FASSOC E (CSAVES C)))
                  (SETQ D (FASSOC P (CDR D))
                   (RETURN (CDR D)))
              (SETQ V
               (L(EQ C DISPCONTEXT)
                (INC GETXNIL#)
                (RETURN (GETHASH E (PHX P))
                       (T (SETQ C (CDV (CDNEXT (CLEVP C))
                (GO DOWN]))
                (OR FLAG (RETURN V))
                (* Don't need context)
            FIND (COND
                ([AND (SETQ D (FASSOC E (CSAVES C)))
                     (FASSOC P (CDR D))
                     (SETQ GETXCONTEXT C1))
                 ((SETQ C1 (CDR C1)) (GO FIND))
                 (T (SETQ GETXCONTEXT NIL)))
                (RETURN V)

```

```

(GETXCTX
 [LAMBDA (E P CTX)
 (INC GETXCTX#) (GETX E P CTX)
 GETXCONTEXT])

```

*SLOWGETX* (handwritten)  
*(OR CTX (CURCTX)) T* (handwritten)

(\* edited (28-MAY-74 . 1658))

```

(PUTX
 [LAMBDA (E P V CTX)
 (INC PUTX#)
 (PROG ([L (CSAVES (OR CTX (SETQ CTX (CURCTX)
 (A (PHX P))
 D D1)
 (COND
 ((EQ CTX DISPCONTEXT)
 ((EQ CTX (ROOTCTX)))
 [COND
 ((NULL (SETQ D (FASSOC E L)))
 (SETCSAVES CTX (CONS (LIST E (CSCCELL P (GETHASH E A)))
 L)))
 ([NULL (SETQ D1 (FASSOC P (CDR D)
 (FRPLACD D (CONS (CSCCELL P (GETHASH E A))
 (CDR D)
 (PUTHASH E V A))
 (T (INC SLOWPUTX#)
 (SLOWPUTX E P V CTX)))
 (RETURN V])
 ((EQ CTX (CURCTX)) (INC DISAPUTX#)
 (DISPLAYCTX CTX)
 (GO TOP))

```

```

(PUTX
 [LAMBDA (E P V CTX)
 (INC PUTX#)
 (PROG ([L (CSAVES (OR CTX (SETQ CTX (CURCTX)
 (A (PHX P))
 D D1)
 (COND
 ((EQ CTX DISPCONTEXT)
 ((EQ CTX (ROOTCTX)))
 [COND
 ((NULL (SETQ D (FASSOC E L)))
 (SETCSAVES CTX (CONS (LIST E (CSCCELL P (GETHASH E A)))
 L)))
 ([NULL (SETQ D1 (FASSOC P (CDR D)
 (FRPLACD D (CONS (CSCCELL P (GETHASH E A))
 (CDR D)
 (PUTHASH E V A))
 (T (INC SLOWPUTX#)
 (SLOWPUTX E P V CTX)))
 (RETURN V])
 ((EQ CTX (CURCTX)) (INC DISAPUTX#)
 (DISPLAYCTX CTX)
 (GO TOP))

```

```

(SLOWPUTX
 [LAMBDA (E P V CTX)
 (PROG ((L (CSAVES CTX))
 (C CTX)
 D D1)
 (COND
 ((NEQ (CDV (CLEVP CTX))
 CTX)
 (* Not on current chain, just
 store value)
 (COND
 ((NULL (SETQ D (FASSOC E L)))
 (SETCSAVES CTX (CONS (LIST E (CSCCELL P V))
 L)))
 ([NULL (SETQ D1 (FASSOC P (CDR D)
 (FRPLACD D (CONS (CSCCELL P V)
 (CDR D)
 (T (FRPLACD D1 V)))
 (RETURN V)))
 DOWN(COND
 ((EQ C DISPCONTEXT)
 (INC PUTXNIL#)
 (PUTHASH E V (PHX P)))
 ([AND (SETQ D (FASSOC E L))
 (SETQ D1 (FASSOC P (CDR D)
 (FRPLACD D V))
 (T (INC PUTXDOWN#)
 [SETQ L (CSAVES (SETQ C (CDV (CDNEXT (CLEVP C)
 (GO DOWN)))
 (RETURN V])

```

```

(SLOWPUTX
 [LAMBDA (E P V CTX)
 (PROG ((L (CSAVES CTX))
 (C CTX)
 D D1)
 (COND
 ((NEQ (CDV (CLEVP CTX))
 CTX)
 (* Not on current chain, just
 store value)
 (COND
 ((NULL (SETQ D (FASSOC E L)))
 (SETCSAVES CTX (CONS (LIST E (CSCCELL P V))
 L)))
 ([NULL (SETQ D1 (FASSOC P (CDR D)
 (FRPLACD D (CONS (CSCCELL P V)
 (CDR D)
 (T (FRPLACD D1 V)))
 (RETURN V)))
 DOWN(COND
 ((EQ C DISPCONTEXT)
 (INC PUTXNIL#)
 (PUTHASH E V (PHX P)))
 ([AND (SETQ D (FASSOC E L))
 (SETQ D1 (FASSOC P (CDR D)
 (FRPLACD D V))
 (T (INC PUTXDOWN#)
 [SETQ L (CSAVES (SETQ C (CDV (CDNEXT (CLEVP C)
 (GO DOWN)))
 (RETURN V])

```

```

DOWN(COND
 ((EQ C DISPCONTEXT)
 (INC PUTXNIL#)
 (PUTHASH E V (PHX P)))
 ([AND (SETQ D (FASSOC E L))
 (SETQ D1 (FASSOC P (CDR D)
 (FRPLACD D V))
 (T (INC PUTXDOWN#)
 [SETQ L (CSAVES (SETQ C (CDV (CDNEXT (CLEVP C)
 (GO DOWN)))
 (RETURN V])

```

```

(/PUTX
 [LAMBDA (E P V CTX)
 (INC /PUTX#) (PROG ([OLD (GETX E P) (OR CTX (SETQ CTX (CURCTX) T)
 [COND
 ((EQ GETXCONTEXT CTX)
 (UNDOSAVE (LIST (QUOTE /PUTX)
 E P OLD CTX)))
 (T (UNDOSAVE (LIST (QUOTE /UNPUTX)
 E P CTX]
 (RETURN (PUTX E P V CTX])

```

*SLOWGETX* (handwritten)

(\* edited (2-JUN-74 . 1359))

```

(ADDX
 [LAMBDA (E P V CTX)
 (PUTX E P (CONS V (GETX E P CTX))
 CTX])

```

(/ADDX  
[LAMBDA (E P V CTX)  
(/PUTX E P (CONS V (GETX E P CTX))  
CTX])

(REMX  
[LAMBDA (E P CTX)  
(PUTX E P NIL CTX)])

(/REMX  
[LAMBDA (E P CTX)  
(/PUTX E P NIL CTX)])

(UNPUTX  
[LAMBDA (E P CTX)  
(INC UNPUTX#)  
(HELP)])

(\* edited (19-MAR-75 . 1217))

~~(REMX E P CTX) (\* For now.)~~

(/UNPUTX  
[LAMBDA (E P CTX)  
(PROG [(OLD (GETX E P [OR CTX (SETQ CTX (CURCTX) T)  
[COND  
((EQ GETXCONTEXT CTX)  
(UNDOSAVE (LIST (QUOTE /PUTX)  
E P OLD CTX])  
(RETURN (UNPUTX E P CTX))

(\* edited (2-JUN-74 . 1402))

← SLOWGETX

(PUTHASH E (CSVAL DI) (PHX P))  
(OR  
(CDR (DREMV DI D))  
(SETCSAVES CTX  
(DREMV D L])  
(EQ CTX (ROOTCTX))  
(PUTHASH E NIL (PHX P))

(CURCTX  
[LAMBDA NIL CURRENTCONTEXT])

(SETCTX  
[LAMBDA (CTX) (INC SETCTX#)  
(SETQ CURRENTCONTEXT CTX)])

(\* edited (18-MAR-75 . 1950))

(RESETCTX  
[LAMBDA (CTX)  
(PROG1 CURRENTCONTEXT (SETCTX CTX))

← (CURCTX)

(ROOTCTX  
[LAMBDA NIL  
ROOTCONTEXT])

(\* edited (31-MAY-74 . 113))

(CTXP  
[LAMBDA (X)  
(AND (LISTP X)  
(LISTP (CAR X))  
([LAMBDA (N)  
(AND (SMALLP N)  
(SMGREATERP N 0)  
(SMLEQ N MAXCONTEXT)  
(EQ (LIST N)  
(CAMP X))

(\* edited (7-AUG-74 . 1527))

← (CLEVP X)

(CI  
[LAMBDA (CTX)  
(CIDX (CTXP CTX)])

(\* edited (19-MAR-75 . 1117))

(CN  
[LAMBDA (N)  
(OR (AND (SMGREATERP N 0)  
(SMLEQ N MAXCONTEXT)  
(LIST N))  
(ERROR N (QUOTE "not a live context"])

← (FINDCN N ROOTCONTEXT)



```

(DMAPCL
[LAMBDA (DML DMFN)
(AND DML (PROG ((DMM (SELECTQ DML
(T (LIST (CURCTX)))
DML)))
DMY DMZ DMV)
L1 (SETCTX (CAR DMM))
[SELECTQ (SETQ DMV (APPLY* DMFN))
(NIL (SETQ DMM (CDR DMM)))
(T [COND
(DMZ (FRPLACD DMY (SETQ DMY DMM)))
(T (SETQ DMZ (SETQ DMY DMM)
(SETQ DMM (CDR DMM))
(FRPLACD DMY NIL))
(PROGN (COND
(DMZ (FRPLACD DMY DMV))
(T (SETQ DMZ DMV)))
(SETQ DMY (FLAST DMV))
(SETQ DMM (CDR DMM))
(COND
(DMM (GO L1))
(T (RETURN DMZ]))
)
)
(CCONC
[LAMBDA (X L)
(SELECTQ X
(NIL L)
(T (CONS (CURCTX)
L))
(NCONC X L])
)
(KILLTREE
[LAMBDA (CTX)
(MAPC (SONCTX CTX)
(FUNCTION KILLTREE))
(KILLCTX CTX])
)
(KILLSONS
[LAMBDA (CTX)
(MAPC (SONCTX CTX)
(FUNCTION KILLTREE))
)
]
[DECLARE: EVAL@COMPILE
(ADDOVAR GLOBALVARS PROPERTIES CONTEXTS NEXTCONTEXT MAXCONTEXT
CURRENTCONTEXT ROOTCONTEXT GETXCONTEXT DISPCONTEXT CDISPEND
CDISPUSED)
]
[DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
(BLOCK: NIL MAPPROP CLRPROP NCTXSET INITCTX CLRCTX CLRCTX1 EXPANDCTX
GROWARRAY GENCTX CSONS SETCSONS DISPEND UNDOCX REDOCX SWAPVALX PHX
KILLCTX SONCTX GCCTX GETX SLOWGETX GETXCTX PUTX SLOWPUTX /PUTX ADDX
/ADDX REMX /REMX UNPUTX /UNPUTX CURCTX SETCTX RESETCTX ROOTCTX CTXP
CI CN DMAPCL CCONC KILLTREE KILLSONS)
(BLOCK: DISPLAYCTX)
]
(MAPC NCTX/FNS (FUNCTION NEW/FN))
(LISPXPRINT (QUOTE (NCTXSET))
T)
(NCTXSET)
(DECLARE: DONTCOPY
(FILEMAP (NIL (2633 3912 (CLIST 2645 . 2690) (SETCLIST 2694 . 2747) (CSOVL
2751 . 2797) (SETCSOVL 2801 . 2854) (CDV 2858 . 2891) (SETCDV 2895 . 2939)
(CDPREV 2943 . 2980) (SETCDPREV 2984 . 3044) (CDLEV 3048 . 3085) (SETCDLEV
3089 . 3149) (CDNEXT 3153 . 3191) (SETCDNEXT 3195 . 3256) (CDCELL 3260 . 3353)
(CSAVES 3357 . 3394) (SETCSAVES 3398 . 3458) (CLEVP 3462 . 3499) (SETCLEVP
3503 . 3563) (CIDX 3567 . 3603) (SETCIDX 3607 . 3666) (CSPROP 3670 . 3706)
(SETCSPROP 3710 . 3757) (CSVAL 3761 . 3796) (SETCSVAL 3800 . 3846) (CSCCELL
3850 . 3909)) (6422 17210 (MAPPROP 6434 . 6766) (CLRPROP 6770 . 6861) (NCTXSET
6865 . 6968) (INITCTX 6972 . 7275) (CLRCTX 7279 . 7547) (CLRCTX1 7551 . 7857)

```

(EXPANDCTX 7861 . 7998) (GROWARRAY 8002 . 8360) (GENCTX 8364 . 8971) (CSONS  
8975 . 9084) (SETCSONS 9088 . 9214) (DISPLAYCTX 9218 . 10150) (DISPEND 10154  
. 10314) (UNDOCX 10318 . 10489) (REDOCX 10493 . 10662) (SWAPVALX 10666 . 11007)  
(PHX 11011 . 11142) (KILLCTX 11146 . 11531) (SONCTX 11535 . 11687) (GCCTX  
11691 . 12017) (GETX 12021 . 12264) (SLOWGETX 12268 . 12896) (GETXCTX 12900  
. 13026) (PUTX 13030 . 13583) (SLOWPUTX 13587 . 14416) (/PUTX 14420 . 14764)  
(ADDX 14768 . 14845) (/ADDX 14849 . 14929) (REMX 14933 . 14984) (/REMX 14988  
. 15041) (UNPUTX 15045 . 15163) (/UNPUTX 15167 . 15450) (CURCTX 15454 . 15493)  
(SETCTX 15497 . 15553) (RESETCTX 15557 . 15686) (ROOTCTX 15690 . 15795) (CTXP  
15799 . 16054) (CI 16058 . 16163) (CN 16167 . 16303) (DMAPCL 16307 . 16913)  
(CCONC 16917 . 17029) (KILLTREE 17033 . 17128) (KILLSONS 17132 . 17207))))  
STOP

(FILECREATED " 7-JAN-75 12:06:11" &lt;DEUTSCH&gt;VEU.;75 17159

changes to: T

previous date: "27-AUG-74 07:19:27" &lt;DEUTSCH&gt;VEU.;74)

```

(LISPXPRINT (QUOTE VEUCOMS)
  T T)
(RPAQQ VEUCOMS ((DECLARE: DOEVAL@COMPILE (ADDVARS (SPECVARS BVMODES EVALARGS))
  )
  (ADDVARS (CONSTATOMS T NIL)
    (SUPEROPS (RESTRICT . RESFIX)))
  (PROP (OPMAKE OPMODE)
    RESTRICT)
  (ENDUMP VEU)))
[DECLARE: DOEVAL@COMPILE
  (ADDTOVAR SPECVARS BVMODES EVALARGS)
]
  (ADDTOVAR CONSTATOMS T NIL)
  (ADDTOVAR SUPEROPS (RESTRICT . RESFIX))
(DEFLIST(QUOTE(
  [RESTRICT (MAKERESTRICT 2 ((RESTRICT)
    (LCOND)
    (CHOICE]
  ))(QUOTE OPMAKE))
(DEFLIST(QUOTE(
  (RESTRICT (MODERESTRICT 2))
  ))(QUOTE OPMODE))
[RPAQQ VEUOPENFNS ((MAPMKN (MAPFN]
(RPAQQ VEUBLKFNS (NLCONSTP CONSTP CONSTVAL VARP BVL MKN))
[RPAQQ VEUPROPS ((OPMAKE (MACRO NOSET))
  (OPMODE (MACRO NOSET))
  (BVLM (MACRO SETMACRO)
    HASH
    (SETHASHQ BVLA 100 1.5]
(RPAQQ VEUFNS
  (MAKECONST CONSTVAL CONSTP NLCONSTP VARP OCCURSFREE OCCFREE FREEV
    FREEV1 UCALLS UCALLS1 BVL GENFROM GENNAME PATLIS1 PATINT
    PATLIS2 PATAND PATLIS EXPORDER EXPORDL EAPPLY FSUBMAKE FSM
    FSSUB FSS RESTRICT MAKERESTRICT MODERESTRICT RESFIX RESFIX1
    MK MK* MKN MAPMKN MD))
[RPAQQ VEUVARS ((BVMODES NIL)
  (EVALARGS NIL)
  (EXPORDL1 (LIST T))
  (EXPORDL2 (LIST T))
  (GENLIST (LIST NIL (QUOTE @)
    1]
(RPAQQ VEUGLOBALS (GENLIST EXPORDL1 EXPORDL2 CONSTATOMS SUPEROPS))
[RPAQQ VEUBLOCKS ((OCCURSFREE OCCURSFREE OCCFREE (LOCALFREEVARS V))
  (FREEV FREEV FREEV1 (LOCALFREEVARS FL BL))
  (UCALLS UCALLS UCALLS1 (LOCALFREEVARS L NL))
  (PATINT PATINT PATLIS PATLIS1 PATLIS2 PATAND)
  (EXPORDER)
  (FSUBMAKE FSUBMAKE FSM (LOCALFREEVARS AL BVFLAG EA))
  (FSSUB FSSUB FSS (LOCALFREEVARS X Y)
[DECLARE: DOEVAL@COMPILE

```

```

(DEFLIST(QUOTE(
  [MAPMKN
    (OPENX (LIST [SUBPAIR (QUOTE (MAPFN))
                    (LIST (CADR OPENX))
                    (QUOTE (LAMBDA
                        (MAPFORM)
                        (* edited NIL)
                        (PROG ((MAPL (MAPCARN (CDR MAPFORM)
                                          MAPFN)))
                            (RETURN (COND
                                ((EQ (CDR MAPFORM)
                                     MAPL)
                                 MAPFORM)
                                (T (MK (CAR MAPFORM)
                                       MAPL]
                                (CAR OPENX]
                    ))(QUOTE MACRO))
  ])
[DECLARE: DOEVAL@COMPILE
  (ADDTOVAR BLKLIBRARY NLCONSTP CONSTP CONSTVAL VARP BVL MKN)
]
[DECLARE: DOEVAL@COMPILE
  (DEFLIST(QUOTE(
    [NLCONSTP (LAMBDA (X)
      (* edited NIL)
      (OR (NOT (LITATOM X))
          (FMEMB X CONSTATOMS]
    [CONSTP (LAMBDA (X)
      (* edited NIL)
      (COND ((LISTP X)
            (EQ (CAR X)
                (QUOTE QUOTE)))
            ((LITATOM X)
             (FMEMB X CONSTATOMS))
            (T T]
    [CONSTVAL (LAMBDA (X)
      (* edited NIL)
      (COND ((LISTP X)
            (CADR X))
            (T X]
    [VARP (LAMBDA (X)
      (* edited (9-JUN-74 . 2246))
      (AND (LITATOM X)
           (NOT (FMEMB X CONSTATOMS]
    [BVL (LAMBDA (BV)
      (* edited NIL)
      (OR (BVL M BV)
          (PROG [(B (MAPCAR (CDR BV)
                          (FUNCTION CADR]
                    (SETBVLM BV B)
                    (RETURN B]
    [MKN (LAMBDA (E L)
      (* edited NIL)
      (COND ((EQ L (CDR E))
            E)
            (T (MK (CAR E)
                   L]
    ))(QUOTE BLKLIBRARYDEF))
  ])
(DEFINEQ
  (OPMAKE
   [LAMBDA (X)
     (FGETP X (QUOTE OPMAKE])
  (OPMODE
   [LAMBDA (X)
     (FGETP X (QUOTE OPMODE])

```

```

(BVLM
  [LAMBDA (X)
    (GETHASH X BVLA)]

(SETBVLM
  [LAMBDA (X V)
    (PUTHASH X V BVLA)]
)
(DEFLIST(QUOTE(
  (OPMAKE (OPMAKE))
  (OPMODE (OPMODE))
  (BVLM (BVLM SETBVLM))
)))(QUOTE NEWPROP)
[DECLARE: DOEVAL@COMPILE
(DEFLIST(QUOTE(
  [OPMAKE ((X)
    (FGETP X (QUOTE OPMAKE)]
  [OPMODE ((X)
    (FGETP X (QUOTE OPMODE)]
  (BVLM ((X)
    (GETHASH X BVLA)))
  (SETBVLM ((X V)
    (PUTHASH X V BVLA)))
)))(QUOTE MACRO))
]
(/ADDPROP (QUOTE PROP)
  (QUOTE NEWPROPS)
  (QUOTE (OPMAKE (MACRO NOSET))))
  T)
(/ADDPROP (QUOTE PROP)
  (QUOTE NEWPROPS)
  (QUOTE (OPMODE (MACRO NOSET))))
  T)
(SETHASHQ BVLA 100 1.5)
(/ADDPROP (QUOTE HASH)
  (QUOTE NEWPROPS)
  (QUOTE (BVLM (MACRO SETMACRO)
    BVLA))
  T)
[DECLARE: EVAL@COMPILE
  (ADDTOVAR GLOBALVARS BVLA)
]
(DEFINEQ

(MAKECONST
  [LAMBDA (X)
    (COND
      ((OR (AND (NLISTP X)
        (NOT (LITATOM X)))
        (FMEMB X CONSTATOMS))
        (HCOPY X))
      (T (HCONS (QUOTE QUOTE)
        (HCONS (HCOPY X)
          NIL]))
    )
  )
  (* edited NIL)

(CONSTVAL
  [LAMBDA (X)
    (COND
      ((LISTP X)
        (CADR X))
      (T X))
  )
  (* edited NIL)

(CONSTP
  [LAMBDA (X)
    (COND
      ((LISTP X)
        (EQ (CAR X)
          (QUOTE QUOTE)))
      (:(LITATOM X)
        (FMEMB X CONSTATOMS))
      (T T))
  )
  (* edited NIL)

```

```

(NLCONSTP
[LAMBDA (X)                                (* edited NIL)
 (OR (NOT (LITATOM X))
      (FMEMB X CONSTATOMS]))

(VARP
[LAMBDA (X)                                (* edited (9-JUN-74 . 2246))
 (AND (LITATOM X)
       (NOT (FMEMB X CONSTATOMS]))

(OCCURSFREE
[LAMBDA (V E)                              (* edited (19-JUN-74 . 128))
 (COND
  ((LISTP V)
   (OCCFREE E))
  (V (SETQ V (FRPLACA (QUOTE (K))
                     V))
      (OCCFREE E]))

(OCCFREE
[LAMBDA (E)                                (* edited (8-AUG-74 . 1021))
 (PROG NIL
  L1 [RETURN (COND
            ((NLISTP E)
             (FMEMB E V))
            (T (SELECTQ (CAR E)
                        ((FA EX FU CHOICE FUNCTION)
                         ([LAMBDA (V)
                           (COND
                            (V (OCCFREE (CDR E])
                             (REML (BVL (CADR E))
                                    V))))
                          (QUOTE NIL)
                          (COND
                           ((SETQ E (CDR E))
                            (GO L2])

            L2 (COND
                ((NULL (CDR E))
                 (SETQ E (CAR E))
                 (GO L1))
                ((OCCFREE (CAR E))
                 (RETURN T))
                (T (SETQ E (CDR E))
                   (GO L2]))

(FREEV
[LAMBDA (E FL BL)                          (* edited (19-JUN-74 . 131))
 (FREEV1 E)]

```

```

(FREEV1
[LAMBDA (E)
  (PROG NIL
    L1 [RETURN (COND
      [(NLISTP E)
        (COND
          ((NOT (LITATOM E))
            FL)
          ((FMEMB E BL)
            FL)
          ((FMEMB E FL)
            FL)
          ((FMEMB E CONSTATOMS)
            FL)
          (T (CONS E FL)
            (T (SELECTQ (CAR E)
              ((FA EX FU CHOICE FUNCTION)
                ([LAMBDA (BL)
                  [MAPC (CDADR E)
                    (FUNCTION (LAMBDA (X)
                      (OR (FMEMB (SETQ X
                        (CADR X))
                          BL)
                        (SETQ BL (CONS X BL)
                          (FREEV1 (CDR E]
                            BL))
                    (QUOTE FL)
                  (COND
                    ((SETQ E (CDR E))
                      (GO L2))
                    (T FL]
                L2 (COND
                  ((NULL (CDR E))
                    (SETQ E (CAR E))
                    (GO L1)))
                  (SETQ FL (FREEV1 (CAR E)))
                  (SETQ E (CDR E))
                  (GO L2])

```

```

(UCALLS
[LAMBDA (E L NL)
  (UCALLS1 E)]
(* edited (19-JUN-74 . 132))

```

```

(UCALLS1
[LAMBDA (E)
  (PROG NIL
    L1 (COND
      ((NLISTP E)
        (RETURN L))
      (T (SELECTQ (CAR E)
        (QUOTE (RETURN L))
        [CALL (COND
          ((FMEMB (CAR (SETQ E (CDR E)))
            NL))
          ((NOT (FMEMB (CAR E)
            L))
            (SETQ L (CONS (CAR E)
              L]
            NIL)))
        (COND
          ((NLISTP (SETQ E (CDR E)))
            (RETURN L)))
        L2 (COND
          ((LISTP (CDR E))
            (SETQ L (UCALLS1 (CAR E)))
            (SETQ E (CDR E))
            (GO L2)))
          (SETQ E (CAR E))
          (GO L1])

```





```

[? (COND
    ((APPLY* (CADR P)
              E)
      (LIST L])
[! (MAPCONC (CDR P)
            (FUNCTION (LAMBDA (X)
                      (PATINT X E L])
      (/ (PATINT (CADDR P)
              (GETX E (CADR P))
          L))
      (& (PATAND (CDR P)
                  E L))
[+?
  (COND
    ((SETQ E (APPLY* (CADDR P)
                     E))
      (LIST (CONS (CONS (CADR P)
                        E)
                  .L])
            (PATLIS P E L]))

```

```

(PATLIS2
 [LAMBDA (PL AL L) (* edited NIL)
  (COND
    ((NULL PL)
     (LIST L))
    (T (MAPCONC AL (FUNCTION (LAMBDA (Y)
                            (MAPCONC (PATINT (CAR PL)
                                        Y L)
                                      (FUNCTION (LAMBDA (Z)
                                                (PATLIS1 (CDR PL)
                                                            AL Z]))

```

```

(PATAND
 [LAMBDA (C E L) (* edited NIL)
  (COND
    ((NULL C)
     (LIST L))
    (T (MAPCONC (PATINT (CAR C)
                        E L)
                (FUNCTION (LAMBDA (X)
                          (PATAND (CDR C)
                                  E X]))

```

```

(PATLIS
 [LAMBDA (P E L) (* edited NIL)
  (COND
    ((OR (FMEMB (QUOTE --)
                (CDR P))
         (EQ (FLENGTH P)
             (FLENGTH E))))
    (PATLIS1 P E L])

```

```

(EXPORDER
 [LAMBDA (E1 E2) (* edited NIL)

```

```

(* Returns T iff E1 is strictly less than E2 in expression
order. Order is: atomic constants
(numbers, CONSTATOMS), then QUOTEd constants, then
variables, then lists; order within each class is SLESS
(ALPHORDER).)

```

```

(PROG NIL
 C [RETURN (COND
      [(LISTP E1)
       (COND
        [(EQ (CAR E1)
             (QUOTE QUOTE))
         (OR (NOT (CONSTP E2))
              (AND (LISTP E2)
                    (SLESS (CADR E1)
                           (CADR E2))
                ((LISTP E2)
                 (COND
                  ((NEQ (CAR E2)
                       (QUOTE QUOTE))
                   (GO L])
                ((LISTP E2)
                 (OR (NEQ (CAR E2)
                          (QUOTE QUOTE))
                     (NLCONSTP E1)))
                ((NLCONSTP E2)
                 (AND (NLCONSTP E1)
                      (ALPHORDER E1 E2)))
                ((NLCONSTP E1)
                 (T (ALPHORDER E1 E2))
                (RETURN (COND
      [(EQ (CAR E1)
           (CAR E2))
       (COND
        ((SETQ E1 (CDR E1))
         (AND (SETQ E2 (CDR E2))
              (GO L)))
        (T (CDR E2])
        (T (SETQ E1 (CAR E1))
           (SETQ E2 (CAR E2))
           (GO C])

```

```

(EXPORDL
 [LAMBDA (L1 L2) (* edited NIL)
 (EXPORDER (FRPLACD EXPORDL1 L1)
            (FRPLACD EXPORDL2 L2])

```

```

(EAPPLY
 [LAMBDA (OP L) (* edited NIL)
 (MAKECONST (COND
  ((NULL L)
   (APPLY* OP))
 [(NULL (CDR L))
  (APPLY* OP (CONSTVAL (CAR L])
 [(NULL (CDDR L))
  (APPLY* OP (CONSTVAL (CAR L])
   (CONSTVAL (CADR L])
 (T (APPLY OP (MAPCARN L (FUNCTION CONSTVAL])

```

```

(FSUBMAKE
 [LAMBDA (AL E BVFLAG) (* edited (25-AUG-74 . 831))
 (COND
  ((NULL AL)
   E)
 (T ([LAMBDA (EVALARGS EA)
      (FSM E)
      EVALARGS EVALARGS])

```

```

(FSM
[LAMBDA (E)
(COND
  ((NLISTP E)
   ([LAMBDA (D)
    (COND
      [D (COND
          ((EQ (CDR D)
              (QUOTE *BV*))
           (SETQ EVALARGS NIL)
           E)
          (T (CDR D]
           ((FASSOC E BVMODES)
            (SETQ EVALARGS NIL)
            E)
          (T E]
           (FASSOC E AL)))
    (T
     (OR
      (PROG ((EVALARGS EA))
        [SETQ E
          (SELECTQ
           (CAR E)
           [(FA EX FU CHOICE FUNCTION)
            (MKN
             E
             (PROG ((AL AL)
                  (BVMODES BVMODES))
                (RETURN
                 (COND
                  [BVFLAG
                   (PROG ((L1 (CADR E))
                        L2)
                    LP (COND
                        ((SETQ L1 (CDR L1))
                         [COND
                          ((NOT (FASSOC (CADAR L1)
                                          AL))
                           (SETQ AL
                                (CONS (CONS (CADAR L1)
                                             (QUOTE *BV*))
                                       AL))
                           (SETQ BVMODES
                                (CONS (CDAR L1)
                                       BVMODES))
                           (SETQ L2 (CONS (CAR L1)
                                           L2]
                          (GO LP)))
                        (RETURN
                         (CONSN
                          (CDR E)
                          (MKBV L2)
                          (MAPCARN (CDDR E)
                                   (FUNCTION FSM]
                          (T [MAPC (CDADR E)
                             (FUNCTION (LAMBDA (X)
                               (SETQ AL
                                (CONS (CONS (CADR X)
                                             (QUOTE *BV*))
                                       AL))
                               (SETQ BVMODES
                                    (CONS (CDR X)
                                           BVMODES]
                               (MAPCARN (CDR E)
                                        (FUNCTION FSM]
                               (QUOTE E)
                               (MAPMKN E (FUNCTION FSM]
                               (RETURN EVALARGS))
                               (SETQ EVALARGS NIL))
                             E])

```

(\* edited (27-AUG-74 . 654))

```
(FSSUB
[LAMBDA (X Y E)
(FSS E)] (* edited NIL)

(FSS
[LAMBDA (E)
(COND
((EQ E Y)
X)
((CONSTP E)
E)
((NLISTP E)
E)
(T (MAPMKN E (FUNCTION FSS))
(* edited NIL)

(RESTRIC
[LAMBDA (E P)
(MK (QUOTE RESTRIC)
(LIST E P))] (* edited (25-AUG-74 . 834))

(MAKERESTRIC
[LAMBDA (EXP PRED)
(COND
((EQ PRED T)
EXP)
[(EQCAR EXP (QUOTE RESTRIC))
(MAKERESTRIC (CADR EXP)
(MKAND (CONS PRED (CDDR EXP)
(T (MK1* (QUOTE RESTRIC)
EXP PRED]))
(* edited (25-AUG-74 . 835))

(MODERESTRIC
[LAMBDA (EXP PRED)
(MD EXP)] (* edited (25-AUG-74 . 837))

(RESFIX
[LAMBDA (OP L Z)
(RESFIX1 OP L Z (CADDAR Z))] (* edited (25-AUG-74 . 837))

(RESFIX1
[LAMBDA (OP L Z C)
(* edited (25-AUG-74 . 837))
(* Called by numerous RES
functions.)
(RESTRIC [MK OP (NCONC (LDIFF L Z)
(CONS (CADAR Z)
(CDR Z)
C))]
```

```
(MK
[LAMBDA (OP L)                                     (* edited (25-AUG-74 . 904))
  (PROG ((Z L)
    (OPM (FGETP OP (QUOTE OPMAKE)))
    D X)
  L1 [COND
    [(NULL Z)
      [COND
        (EVALARGS (SETQ L (MAPCARN L EVALARGS)
          (RETURN (SELECTQ (CADR OPM)
            (ARB (APPLY* (CAR OPM)
              L))
            (NIL (HELP))
            (FORM (APPLY* (CAR OPM)
              OP L))
            (APPLY (CAR OPM)
              L])
          ((LISTP (SETQ X (CAR Z)))
            (COND
              ((SETQ D (FASSOC (CAR X)
                SUPEROPS))
                (COND
                  ([NULL (SETQ X (FASSOC (CAR X)
                    (CADDR OPM)
                    (RETURN (APPLY* (CDR D)
                      OP L Z)))
                  ((CDR X)
                    (RETURN (APPLY* (CDR X)
                      OP L Z])
                (SETQ Z (CDR Z))
                (GO L1])
```

```
(MK*
[LAMBDA K                                           (* edited (25-AUG-74 . 839))
  (MK (ARG K 1)
    (PROG ((J K)
      L)
    LP (COND
      ((EQ J 1)
        (RETURN L)))
      (SETQ L (CONS (ARG K J)
        L))
      (SUBIVAR J)
      (GO LP])
```

```
(MKN
[LAMBDA (E L)                                       (* edited NIL)
  (COND
    ((EQ L (CDR E))
      E)
    (T (MK (CAR E)
      L])
```

```
(MAPMKN
[LAMBDA (MAPFORM MAPFN)                             (* edited NIL)
  (PROG ((MAPL (MAPCARN (CDR MAPFORM)
    MAPFN)))
    (RETURN (COND
      ((EQ (CDR MAPFORM)
        MAPL)
        MAPFORM)
      (T (MK (CAR MAPFORM)
        MAPL])
```



(FILECREATED "25-FEB-75 13:35:33" <DEUTSCH>VIN.;4 7632

changes to: PERROR PUNCGET NNODE NCUT ARBMARK ARBNOE ARBCUT SINT  
PARSEGOAL INSET READLN FGCC ENDCONC1 LEXLN TRIM

previous date: "25-FEB-75 13:01:48" <DEUTSCH>VIN.;1)

✓ VINGLOBALS = (LINERDTBL)

(LISPXPRINT (QUOTE VINCOMS)

T T)

(RPAQQ VINCOMS ((ENDUMP VIN)))

(RPAQQ VINFNS

(PERROR PUNCGET NNODE NCUT ARBMARK ARBNOE ARBCUT SINT PARSEGOAL INSET  
READLN TRIM LEXLN FGCC ENDCONC1))

[RPAQQ VINVARS ((XEOLCODE 31)

(XEOL (CHARACTER XEOLCODE))

✓ -> (~~LINEBRKLIST (LIST XEOL)~~)

[RPAQQ VINBLOCKS ((SINT SINT NNODE NCUT ARBMARK ARBNOE ARBCUT PUNCGET)  
(LEXLN LEXLN ENDCONC1 FGCC (GLOBALVARS XEOLCODE)

(DEFINEQ

(PERROR

[LAMBDA (IN)

(RETFROM (STKPOS (QUOTE PARSEGOAL)

1)

(CONS (QUOTE ?) IN])

(PUNCGET

[LAMBDA (IN)

(SELECTQ (CAR IN)

(((\$ID \$NUM \$QUOTE \$EOL)

NIL)

(PROGN (SETQ STACK (CONS (CAR IN)  
STACK))

(CDR IN])

✓ (VINSET  
[ ] NIL

(SETQ LINERDTBL

(COPYREADTABLE 'ORIG))

(SETBRK (LIST XEOL)

NIL LINERDTBL)

(SETSEPR NIL NIL  
LINERDTBL])

(NNODE

[LAMBDA (A N)

(PROG ((L (SMNTH STACK N))

R)

(SETQ R (CDR L))

(FRPLACD L NIL)

(SETQ STACK (CONS (CONS A (DREV STACK))  
R])

(NCUT

[LAMBDA (N)

(PROG ((L (SMNTH STACK N))

R)

(SETQ R (CDR L))

(FRPLACD L NIL)

(SETQ STACK (CONS (DREV STACK)  
R])

(ARBMARK

[LAMBDA (N)

(COND

((ZEROP N)

(SETQ STACK (CONS (QUOTE \*MARK\*)  
STACK)))

(T (PROG ((L (FNTH STACK N))

(FRPLACD L (CONS (QUOTE \*MARK\*)  
(CDR L]))

(ARBNOE

[LAMBDA (A)

(PROG ((L (FMEMB (QUOTE \*MARK\*)  
STACK))

R)

(SETQ R (CDR L))

(FRPLACD L NIL)

(SETQ STACK (CONS (FRPLACA (DREV STACK)  
A)

R])

(ARBCUT

[LAMBDA NIL

(PROG ((L (FMEMB (QUOTE \*MARK\*)  
STACK))

R)

(SETQ R (CDR L))

(FRPLACD L NIL)

(FRPLACA (SETQ STACK (DREV STACK))

(CDR STACK))

(FRPLACD STACK R])

(SINT

[LAMBDA (IN2 L)

(PROG (X R NOBACK)

LP (COND

((NULL L)

(RETURN IN2))

[(NLISTP (SETQ X (CAR L)))

(COND

([SETQ IN2 (SELECTQ (CAR IN2)

[(\$ID \$NUM \$QUOTE)

(COND

((EQ (CAR IN2)

X)

(SETQ STACK (CONS (CADR IN2)  
STACK))

(CDDR IN2))

((EQ (CADR IN2)

X)

(CDDR IN2]

(COND

((EQ (CAR IN2)

X)

(CDR IN2))

((EQ X (QUOTE \$PUNC))

(PUNCGET IN2]

(SETQ L (CDR L))

(GO LP]

(T (SETQ R (CDR X))

(SETQ L (CDR L))

[SETQ NOBACK (SELECTQ

(CAR X)

[: (COND

((NULL (CDR R))

(SETQ STACK (CONS (CAR R)  
STACK)))

((EQ (CADR R)

(QUOTE \$))

(ARBNOSE (CAR R)))

(T (NNOSE (CAR R)

(CADR R]

[; (COND

((EQ (CAR R)

(QUOTE \$))

(ARBCUT))

(T (NCUT (CAR R]

(\$ (ARBMARK (CAR R)))

[# (COND

[(EQ (SETQ X (CADR R))

1)

(FRPLACA STACK (APPLY\* (CAR R)

(CAR STACK]

(T (COND

((EQ X (QUOTE \$))

(ARBCUT))

(T (NCUT X)))

(FRPLACA STACK (APPLY (CAR R)

(CAR STACK]

(EITHER (GO V))



```

(REPEAT (GO RP))
(LIST (GO LST))
(OPTION (SETQ IN2 (OR (SINT IN2 R)
                      IN2)))

(* (GO LP))
[TEST (COND
      (([LAMBDA (STACK)
          (SINT IN2 R]
          STACK)
          (GO LP))
      (T (GO NO]
(COND
  ([SETQ X (COND
    [(SETQ R (PARSEDEF (CAR X)))
    (COND
      (L (SINT IN2 R))
      (T (SETQ L R)
          (GO LP]
      (T (APPLY* (CAR X)
                 IN2
                 (CDR X]
        (SETQ IN2 X))
      (T (GO NO]
      (GO LP)))
NO (COND
  (NOBACK (PERROR IN2))
  (T (RETURN NIL)))
LST (COND
  ((SETQ X (SINT IN2 (CDR R)))
   (SETQ NOBACK (SETQ IN2 X)))
  (T (GO NO)))
RP (COND
  ((SETQ X (SINT IN2 R))
   (SETQ NOBACK (SETQ IN2 X))
   (GO RP)))
(GO LP)
V (COND
  ((NULL R)
   (GO NO))
  ((SETQ X (SINT IN2 (CAR R)))
   (SETQ NOBACK (SETQ IN2 X))
   (GO LP))
  (T (SETQ R (CDR R))
   (GO V]))

(PARSEGOAL
 [LAMBDA (LN GOAL)
 (SINT LN (COND
  ((LITATOM GOAL)
   (PARSEDEF GOAL))
  (T GOAL]))

(INSET
 [LAMBDA (BL CL A)
 [COND
  [(LITATOM (CAR A))
   (RPLACA A (SETQ A (ARRAY 128]
   (T (SETQ A (CAR A))
      (MAPINT 1 128 (FUNCTION (LAMBDA (X)
                             (SETA A X NIL]
 [MAPC BL (FUNCTION (LAMBDA (X)
                    (SETA A (ADD1 (OR (NUMBERP X)
                                       (CHCON1 X)))
                    (QUOTE BRK]
 [MAPC CL (FUNCTION (LAMBDA (X)
                    (SETA A [ADD1 (OR (NUMBERP (CAR X))
                                       (CHCON1 (CAR X]
                    (CDR X]
 (SETA A (ADD1 XEOLCODE)
 (QUOTE EOL])

```

→ VINSET

(READLN

[LAMBDA (FILE)

~~(SETBRK LINEBRKLIST)~~~~(SETSEPR NIL)~~

(PROG1 (RSTRING FILE)

(READC FILE])

← LINEARDTBL

(TRIM

[LAMBDA (S)

(PROG ((M 1)

(N -1))

LQ (SELECTQ (NTHCHAR S M)

((% % )

(ADDIVAR M)

(GO LQ))

(NIL (RETURN (QUOTE ""))))

NIL)

LP (RETURN (COND

((EQ (NTHCHAR S N)

(QUOTE % )

(SUBIVAR N)

(GO LP))

((AND (EQ M 1)

(EQ N -1))

S)

(T (SUBSTRING S M N]))

(LEXLN

[LAMBDA (S CCA L)

(\* edited (25-FEB-75 . 1329))

(PROG ((U (OR (SUBSTRING S 1 -1)

(QUOTE ""))))

(W (QUOTE ""))

(R (OR (LISTP L)

(CONS)))

(K 0)

K1 CC1 END A C CC)

(SETQ END R)

L1 (ADDIVAR K)

(SETQ C (ELTS CCA (SMPLUS (SETQ CC (OR (FGNCC U)

XEOLCODE))

1)))

L3 (SELECTQ C

(BRK (SETQ END (ENDCONC1 END (FCHARACTER CC)))

(GO L1))

(NIL)

(SEPR (GO L1))

[EOL (SETQ END (ENDCONC1 END (QUOTE \$EOL)))

(RETURN (PROG1 (CDR R)

(COND

((CDR END)

(FRPLACD (FLAST END)

(CDR L))

(FRPLACD L (CDR END))

(FRPLACD END NIL]

(QUOTE [PROG ((K2 K))

SLP (ADDIVAR K)

(COND

((AND (SETQ CC1 (FGNCC U))

(NEQ CC1 CC))

(GO SLP)))

(SETQ END (ENDCONC1 (ENDCONC1 END (QUOTE \$QUOTE)

)

(SUBSTRING S (ADD1 K2)

(SUB1 K]

(GO L1))

(HELP))

(SETQ K1 K)

L2 (SETQ C (ELTS CCA (SMPLUS (SETQ CC1 (OR (FGNCC U)  
 XEOLCODE)))

1)))

[SETQ A (COND  
 ((NULL C)  
 (ADD1VAR K)  
 (GO L2))  
 ((EQ K1 K)  
 (FCHARACTER CC))  
 (T (MKATOM (SUBSTRING S K1 K W)  
 (SETQ END (ENDCONC1 [ENDCONC1 END (COND  
 ((NUMBERP A)  
 (QUOTE \$NUM))  
 (T (QUOTE \$ID)

A))

(SETQ CC CC1)  
 (ADD1VAR K)  
 (GO L3])

(FGNCC

[LAMBDA (S)

(ASSEMBLE NIL

(CQ S)

(LDB 2 , = 251701000000Q)

(JUMPE 2 , NUL)

(LDB 2 , = 2501000000Q)

(MOVE 3 , = -7777777Q)

(ADDM 3 , 0 (1))

(IDIVI 2 , 5)

(IMUL 3 , = -70000Q)

(HRLI 2 , 350700Q (3))

(LDB 1 , 2)

(ADDI 1 , (LOC 0))

(SKIPA)

NUL (CQ NIL])

(ENDCONC1

[LAMBDA (Z X)

(COND

((CDR Z)

(FRPLACA (CDR Z)

X))

(T (CDR (FRPLACD Z (FRPLACD (CONS X Z)

NIL])

)

(RPAQ XEOLCODE 31)

(RPAQ XEOL (CHARACTER XEOLCODE))

(RPAQ LINEBRKLIST (LIST XEOL))

[DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY

(BLOCK: NIL PERROR PARSEGOAL INSET READLN TRIM)

(BLOCK: SINT SINT NNODE NCUT ARBMARK ARBNODE ARBCUT PUNCGET)

(BLOCK: LEXLN LEXLN ENDCONC1 FGNCC (GLOBALVARS XEOLCODE))

]

(DECLARE: DONTCOPY

(FILEMAP (NIL (682 7288 (PERROR 694 . 796) (PUNCGET 800 . 969) (NNODE 973

. 1154) (NCUT 1158 . 1327) (ARBMARK 1331 . 1525) (ARBNOE 1529 . 1744) (ARBCUT

1748 . 1981) (SINT 1985 . 4209) (PARSEGOAL 4213 . 4327) (INSET 4331 . 4789)

(READLN 4793 . 4909) (TRIM 4913 . 5280) (LEXLN 5284 . 6774) (FGNCC 6778 .

7142) (ENDCONC1 7146 . 7285))))))

STOP

(FILECREATED "25-FEB-75 13:28:30" <DEUTSCH>VOUT.;3 8313

changes to: OUTQ OSPA OFMT UNOPR PRINTGRAPH PGRAPH1 ARROWTAB LOCMAP  
PRINLOC1 LOCMAP2 FORMPRIN1 FORMPRIN\* VOUTVARS UNPARSE OUT VOUTCCMS

previous date: "25-FEB-75 13:01:52" <DEUTSCH>VOUT.;1)

(LISXPXPRINT (QUOTE VOUTCOMS)

T T)  
[RPAQQ VOUTCOMS

((DECLARE: DOEVAL@COMPILE (PROP MACRO OUTQ)  
(ADDVARS (FUNNYATOMLST \*L1\* \*PAREN\* \*PAREN\*FMT \*OTHER\*FMT  
\*FILE\* \*SP\* \*POS\* \*LEVEL\*)))

~~((DECLARE: DOEVAL@COMPILE COMPILEVAR  
(ADDVARS (GLOBALVARS \*L1\* \*MINUS\* \*PAREN\* \*PAREN\*FMT  
\*OTHER\*FMT  
(LOCALFREEVARS \*FILE\* \*SP\* \*POS\* \*LEVEL\*)))~~

→ VOUT GLOBALS

(ENDUMP VOUT)  
(DECLARE: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILEVAR  
(ADDVARS (NLAMA)  
(NLAML OUTQ)

[DECLARE: DOEVAL@COMPILE  
(DEFLIST(QUOTE(  
[OUTQ (X (LIST (QUOTE OUT)  
(KWOTE (CAR X)  
)))(QUOTE MACRO))

(ADDTOVAR FUNNYATOMLST \*L1\* \*PAREN\* \*PAREN\*FMT \*OTHER\*FMT \*FILE\* \*SP\* \*POS\*  
\*LEVEL\*)

] [DECLARE: DOEVAL@COMPILE COMPILEVAR  
(ADDTOVAR GLOBALVARS \*L1\* \*MINUS\* \*PAREN\* \*PAREN\*FMT \*OTHER\*FMT)  
(ADDTOVAR LOCALFREEVARS \*FILE\* \*SP\* \*POS\* \*LEVEL\*)

] (RPAQQ VOUTBLKFNS (OSPA OFMT))  
[RPAQQ VOUTPROPS ((PARSEDEF (MACRO NOSET))  
(PGVISIT (MACRO SETMACRO)  
HASH  
(SETHASHQ PGARRAY 100 1.5]

(RPAQQ VOUTFNS  
(UNPARSE OUT OUTQ OSPA OFMT UNOPR PRINTGRAPH PGRAPH1 ARROWTAB LOCMAP  
PRINLOC1 LOCMAP2 FORMPRIN1 FORMPRIN\*))

[RPAQQ VOUTVARS (\*OTHER\*FMT \*PAREN\*FMT (\*L1\* (LIST T))  
(\*PAREN\* (CONS

→ (RPAQQ VOUTBLOCKS ((PRINTGRAPH PRINTGRAPH PGRAPH1 ARROWTAB) (UNPARSE UNPARSE UNOPR OUT)  
(LOCMAP2 LOCMAP2 PRINLOC1)))

[DECLARE: DOEVAL@COMPILE  
(ADDTOVAR BLKLIBRARY OSPA OFMT)

] [DECLARE: DOEVAL@COMPILE  
(DEFLIST(QUOTE(  
[OSPA (LAMBDA (X)  
(AND (LITATOM X)  
(FGETP X SPA]  
[OFMT (LAMBDA (X)  
(AND (LITATOM X)  
(FGETP X FMTA]

))(QUOTE BLKLIBRARYDEF))

] (DEFINEQ

(PARSEDEF  
[LAMBDA (X)  
(FGETP X (QUOTE PARSEDEF])

(PGVISIT  
[LAMBDA (X)  
(GETHASH X PGARRAY])

```

(SETPGVISIT
[LAMBDA (X V)
(PUTHASH X V PGARRAY)]
)
(DEFLIST(QUOTE(
(PARSEDEF (PARSEDEF))
(PGVISIT (PGVISIT SETPGVISIT))
))(QUOTE NEWPROP))
[DECLARE: DOEVAL@COMPILE
(DEFLIST(QUOTE(
[PARSEDEF ((X)
(FGETP X (QUOTE PARSEDEF))
(PGVISIT ((X)
(GETHASH X PGARRAY)))
(SETPGVISIT ((X V)
(PUTHASH X V PGARRAY)))
]))(QUOTE MACRO))
]
(/ADDPROP (QUOTE PROP)
(QUOTE NEWPROPS)
(QUOTE (PARSEDEF (MACRO NOSET)))
T)
(SETHASHQ PGARRAY 100 1.5)
(/ADDPROP (QUOTE HASH)
(QUOTE NEWPROPS)
(QUOTE (PGVISIT (MACRO SETMACRO)
PGARRAY))
T)
[DECLARE: EVAL@COMPILE
(ADDTOVAR GLOBALVARS PGARRAY)
]
(DEFINEQ
(UNPARSE
[LAMBDA (E FMTA SPA *FILE*) (* edited (25-FEB-75 . 1319))
(PROG ((*SP* 0)
(*POS*(POSITION *FILE*))
(*LEVEL* 0))
(UNOPR E 0])
)
(OUT
[LAMBDA (X) (* edited (25-FEB-75 . 1319))
(COND
((IGREATERP (IPLUS (LOGAND *SP* 7)
(RSH (SETQ *SP*(OR (OSPA X)
26))
3))
4)
(PRIN1 (QUOTE %)*FILE*)))
(PRIN1 X *FILE*])
)
(OUTQ
[NLAMBDA (X)
(OUT X)]
)
(OSPA
[LAMBDA (X)
(AND (LITATOM X)
(FGETP X SPA))
]
)
(OFMT
[LAMBDA (X)
(AND (LITATOM X)
(FGETP X FMTA))
]
)

```

```

(UNOPR
[LAMBDA (E L)
  (PROG (P X)
    L1 (COND
      [(NUMBERP L)
        (COND
          [(NLISTP E)
            (COND
              ((LITATOM E)
                (OUT E))
              [(NUMBERP E)
                (COND
                  ((MINUSP E)
                    (FRPLACA (CDR *MINUS*)
                      (MINUS E))
                    (SETQ E *MINUS*)
                    (GO L1))
                  (T (OUT E))]
              ((STRINGP E)
                (OUTQ %)
                (PRIN1 E)
                (OUTQ %)))
              (T (OUT E))]
          ([NOT (NUMBERP (SETQ X (CAR (SETQ P (OR (OFMT (CAR E))
            (OFMT (QUOTE *OTHER*))
            *OTHER*FMT]

              (SETQ L P)
              (GO L3))
            ((IGREATERP L X)
              (SETQ L (CDR (OR (OFMT (QUOTE *PAREN*)
                *PAREN*FMT)))
              (SETQ E (FRPLACA *PAREN* E)
              (GO L1))
              (T (SETQ L (CDR P))
              (GO L1)
              (L (GO L3)))
            (RETURN (SUB1VAR *LEVEL*)))
    L3 [COND
      ((SMLEQ (PRINTLEVEL)*LEVEL*)
        (RETURN (OUTQ ?)
        (ADD1VAR *LEVEL*))
    L2 [COND
      ((NLISTP (SETQ X (CAR L)))
        (OUT X))
      [(LITATOM (SETQ P (CAR X)))
        (SETQ E (BLKAPPLY P (FRPLACD (FRPLACA *L1* E)
          (CDR X)

          ((SMINUSP P)
            (SUB1VAR *LEVEL*)
            [MAP (SMNTH (CDR E)
              (SMINUS P))
              (FUNCTION (LAMBDA (Y)
                (UNOPR Y (CDR X)
                (ADD1VAR *LEVEL*))
              (T (UNOPR (CAR (SMNTH E (SMADD1 P)))
                (CDR X]

          (COND
            ((SETQ L (CDR L))
            (GO L2)))
          (SUB1VAR *LEVEL*])

```

(PRINTGRAPH

```

[LAMBDA (ROOT SUCCFN P1FN P2FN KEEP)
 (PROG (COL (ROW (OR (NUMBERP KEEP)
 1))
 (POS (POSITION))
 (LST (LIST ROOT))
 N)
 (COND
 ((NOT KEEP)
 (CLRHASH PGARRAY)))
 LP (COND
 ((NULL LST)
 (RETURN ROW)))
 (SETQ N (CAR LST))
 (SETQ LST (CDR LST))
 (COND
 ((EQ (PGVISIT N)
 T)
 (SETPGVISIT N NIL)
 (TAB POS)))
 (SETQ COL 1)
 (PGRAPH1 N NIL)
 (SETQ ROW (ADD1 ROW))
 (GO LP])

```

(PGRAPH1

```

[LAMBDA (NODE L)
 (PROG (FLAG Z)
 (COND
 [(SETQ FLAG (PGVISIT NODE))
 (PRINQ %[)
 (APPLY* P1FN NODE T)
 (PRINQ %])
 (COND
 ((EQ FLAG T)
 (PRINTQ **))
 (T (PRIN2 (CAR FLAG))
 (PRINQ ,)
 (PRINT (CDR FLAG))
 ((APPLY* P1FN NODE NIL)
 (SETPGVISIT NODE (CONS ROW COL))
 [MAP (SETQ Z (APPLY* SUCCFN NODE))
 (FUNCTION (LAMBDA (X)
 [COND
 [FLAG (ARROWTAB L T)
 (SETQ ROW (ADD1 ROW))
 (SETQ COL 1)
 (COND
 ((NULL (CDR X))
 (SETQ L (CDR L))
 (T [COND
 ((CDR Z)
 (SETQ FLAG (SETQ L (CONS (SUB1 (POSITION))
 L]
 (SETQ COL (ADD1 COL))
 (SETQ X (CAR X))
 (APPLY* P2FN NODE X)
 (PGRAPH1 X L]
 (OR Z (TERPRI)))
 (T (SETPGVISIT NODE T)
 (SETQ LST (NCONC1 LST NODE))
 (PGRAPH1 NODE L])

```

(ARROWTAB

```

[LAMBDA (L LAST)
 (COND
 ((NULL L)
 NIL)
 (T (ARROWTAB (CDR L))
 (TAB (CAR L))
 (PRIN1 (COND
 (LAST (QUOTE +))
 (T (QUOTE !]))

```

```

(LOCMAP
  [LAMBDA (X)
    (RESETFORM (RADIX 8)
      (LOCMAP2 X))
    (TERPRI])

(PRINLOC1
  [LAMBDA (X)
    (PRIN1 (LOC X))
    (PRINQ ": ")])

(LOCMAP2
  [LAMBDA (E)
    (COND
      ((NLISTP E)
        (COND
          ((NOT (OR (ARRAYP E)
                    (SMALLP E)
                    (LITATOM E)))
            (PRINLOC1 E)))
          (PRIN2 E))
      (T (PROG NIL
          (PRINQ %())
          L1 (PRINLOC1 E)
            (LOCMAP2 (CAR E))
            (COND
              ((LISTP (SETQ E (CDR E)))
                (PRINQ % )
                (GO L1))
              (E (PRINQ " . ")
                (LOCMAP2 E)))
            (PRINQ %)
            (RETURN T]))

(FORMPRIN1
  [LAMBDA (FORM L)
    (APPLY (QUOTE FORMPRIN*)
      (CONS FORM L])

(FORMPRIN*
  [LAMBDA K
    (PROG ((J 2)
      (MAPC (ARG K 1)
        (FUNCTION (LAMBDA (Z)
          (COND
            ((STRINGP Z)
              (PRIN1 Z))
            ((LITATOM Z)
              (APPLY* Z (ARG K J))
              (ADD1VAR J))
            ([[LAMBDA (X)
              (EVAL Z]
              (ARG K J))
              (ADD1VAR J])
          )
      (RPAQQ *OTHER*FMT (1500 % (0 . 10000)
        (-1 % (0 . 10000))
        %)))
      (RPAQQ *PAREN*FMT (1500 % (0 . 0)
        %)))
      (RPAQ *L1* (LIST T))
      (RPAQ *PAREN* (CONS))
    [DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
      (BLOCK: NIL UNPARSE OUT OUTQ OSPA OFMT UNOPR LOCMAP FORMPRIN1 FORMPRIN*)
      (BLOCK: PRINTGRAPH PRINTGRAPH PGRAPH1 ARROWTAB)
      (BLOCK: LOCMAP2 LOCMAP2 PRINLOC1)
    ]
    [DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVERS
      (ADDTOVAR NLAMA)

```



```

(ADDOVAR NLAML OUTQ)
]
(LISXPXPRINT (QUOTE VOUTCOMS)
  T T)
[RPAQQ.VOUTCOMS
  ((DECLARE: DOEVAL@COMPILE (PROP MACRO OUTQ)
    (ADDVARS (FUNNYATOMLST *L1* *PAREN* *PAREN*FMT *OTHER*FMT
      *FILE* *SP* *POS* *LEVEL*)))
  (DECLARE: DOEVAL@COMPILE COMPILERVARS (ADDVARS (NLAMA)
    (NLAML OUTQ)))
  (ENDUMP VOUT)
  (DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS
    (ADDVARS (NLAMA)
      (NLAML OUTQ)
    )
  )
[DECLARE: DOEVAL@COMPILE COMPILERVARS
  (ADDOVAR NLAMA)
  (ADDOVAR NLAML OUTQ)
]
(DECLARE: DONTCOPY
  (FILEMAP (NIL (1901 2085 (PARSEDEF 1913 . 1969) (PGVISIT 1973 . 2022) (
  SETPGVISIT 2026 . 2082)) (2702 7270 (UNPARSE 2714 . 2894) (OUT 2898 . 3156)
  (OUTQ 3160 . 3195) (OSPA 3199 . 3259) (OFMT 3263 . 3324) (UNOPR 3328 . 4698)
  (PRINTGRAPH 4702 . 5247) (PGRAPH1 5251 . 6157) (ARROWTAB 6161 . 6328) (LOCMAP
  6332 . 6417) (PRINLOC1 6421 . 6485) (LOCMAP2 6489 . 6897) (FORMPRIN1 6901
  . 6981) (FORMPRIN* 6985 . 7267))))))
STOP

```

} Classified?

(FILECREATED "25-FEB-75 13:35:12" <DEUTSCH>VEP.;33 8564

changes to: TRIM UNPLUS UNPOSP UNPOSL UNCFAC UNREL UNCOND UNQT UNPLUS UNPOSP UNPOSL UNCFAC UNREL UNCOND UNQT VEPCC:IS

previous date: " 2-FEB-75 17:55:23" <DEUTSCH>VEP.;32)

(LISXPRT (QUOTE VEPCOMS)

T T)

(RPAQQ VEPCOMS

~~((DECLARE: DOEVAL@COMPILE (PROP FILEGROUP VEP)))~~

~~(PROP SEFMT EITHER REPEAT OPTION LIST SEQ TEST \*PAREN\*)~~

~~(PROP SENAME \$PUNC \$ID \$NUM \$QUOTE \$ { } % [ % ] % |)~~

~~(ENDUMP VEP))~~

~~((DECLARE: DOEVAL@COMPILE~~

~~(DEFLIST(QUOTE(~~

~~(VEP (VEP VOUT)~~

~~))(QUOTE FILEGROUP))~~

(DEFLIST(QUOTE(

[EITHER (40 (1 . 40)

(-2 % % (0 . 40]

(REPEAT (60 \$ (1 . 60)))

(OPTION (1000 %[ (1 . 0)

%]))

(LIST (60 \$ < (1 . 0)

>

(2 . 60)))

[SEQ (50 (1 . 50)

(-2 % (0 . 50]

(TEST (1000 < (1 . 0)

(-2 % (0 . 0))

>))

(\*PAREN\* (1100 { (0 . 0)

}))

))(QUOTE SEFMT))

(DEFLIST(QUOTE(

(\$PUNC <punc>

(\$ID <id>

(\$NUM <num>

(\$QUOTE <str>

(\$ "\$")

({ "{")

() "}")

(% [ "[")

(% ] "]"")

(% | "|")

))(QUOTE SENAME))

(RPAQQ VEPFNS

(PRINTEST PCTX PCTXX SPCTX EXPRIN SENAME VEPSET SEPRINT SEP1 SEP2

LEXSTAT PARSE EDITLINE READSTAT UNPLUS UNPOSP UNPOSL UNCFAC

UNREL UNCOND UNQT))

[RPAQQ VEPVARS (LEXBRKLIST LEXCLIST SYSPRINTLIST (\*MINUS\*

(LIST (QUOTE MINUS)

0))

(\*TIMES\* (LIST (QUOTE TIMES)

0]

(RPAQQ VEPGLOBALS (STLEXA STLEXL \*MINUS\* \*TIMES\*))

~~(RPAQQ VEPBLOCKS (PSBLOCK UNPARSE UNOPR UNPLUS UNPOSP UNPOSL UNCFAC UNREL~~

~~UNCOND UNQT OUT (ENTRIES UNPARSE)~~

~~(BLKAPPLYENS UNPLUS UNCFAC UNREL UNCOND UNQT)~~

(DEFINEQ

(PRINTEST

[LAMBDA (X FILE)

(COND

((HCONSP X)

(EXPRIN X FILE)

T)

((CTXP X)

(PCTX X FILE)

T])

(SEP1 SEP1 SEP2)

(\* Called from NPRIN)

✓ ✗ to VSEP

✗ get rid of OUTA

OUTPARSE

OUTPARSE

✗

```
(PCTX
[LAMBDA (CTX)
(PRINQ <)
[PRINZ (CI (OR CTX (SETQ CTX (CURCTX)
[COND
((CDR CTX)
(PRINQ /)
(PRINZ (CI (CDR CTX)
(PRINQ >])
```

```
(PCTXX
[LAMBDA (CTX)
(PRINQ <)
(PRINZ (CI CTX))
[MAP (CDR CTX)
(FUNCTION (LAMBDA (X)
(PRINQ /)
(PRINZ (CI X)
(PRINQ >])
```

```
(SPCTX
[LAMBDA (CTX)
(OR CTX (SETQ CTX (CURCTX)))
(SPACES (COND
[(CDR CTX)
(IPLUS 3 (NCHARS (CI CTX))
(NCHARS (CI (CDR CTX)
(T 3])
```

```
(EXPRIN
[LAMBDA (E FILE)
(UNPARSE E (QUOTE STFMT)
(QUOTE STSPA)
FILE])
```

*OUTPARSE*

```
(SENAME
[LAMBDA (X)
(COND
((GETP X (QUOTE SENAME)))
(T (SETQ NL (ADDELT X NL))
(PROG [(P (QUOTE (< T >))
(FRPLACA (CDR P)
(L-CASE X))
(RETURN (PACK P])
```

```
(VEPSET
[LAMBDA NIL
(SETQ STLEXL (CONS))
(INSET LEXBRKLIST LEXCLIST (QUOTE STLEXA))
(PRINTYPE 8 (FUNCTION PRINTEST))
(MAPC SYSPRINTLIST (FUNCTION (LAMBDA (X)
[MAPC (CDR X)
(FUNCTION (LAMBDA (Y)
(ADVISE (LIST (CAR Y)
(QUOTE IN)
(CAR X))
(QUOTE BEFORE)
NIL
(LIST (QUOTE RETURN)
(CONS (CDR Y)
(SMARTARGLIST (CAR Y]))
```

(\* edited (2-FEB-75 . 1752))

*RELINK (CAR X]*

```

(SEPRINT
[LAMBDA (D)
(COND
((NEQ (GETP D (QUOTE SENAME))
(QUOTE NONE))
(PROG (NL P E)
(TERPRI)
(TAB 3)
(PRIN2 (SENAME D))
(PRINQ " := ")
(SETQ P (POSITION))
(SETQ E (SEP1 (PARSEDEF D)))
[COND
[(EQCAR E (QUOTE EITHER))
(MAP (CDR E)
(FUNCTION (LAMBDA (X)
(UNPARSE (CAR X) <OUTPARSE
(QUOTE SEFMT)
(QUOTE STSPA))
(COND
((CDR X)
(PRINTQ % |)
(TAB P]
(T (UNPARSE E (QUOTE SEFMT)
(QUOTE STSPA]
(TERPRI)
(RETURN NL])

```

```

(SEP1
[LAMBDA (L)
(SETQ L (MAPCONC L (FUNCTION SEP2)))
(COND
((CDR L)
(CONS (QUOTE SEQ)
L))
(T (CAR L])

```

```

(SEP2
[LAMBDA (E)
(COND
[(NLISTP E)
(LIST (COND
((NOT (LITATOM E))
E)
((GETP E (QUOTE SENAME)))
(T E]
(T (SELECTQ
(CAR E)
((; : $ # * TEST)
NIL)
[EITHER (SETQ E (MAPCAR (CDR E)
(FUNCTION SEP1)))
(LIST (COND
[(EQUAL (LAST E)
(QUOTE (NIL)))
[SETQ E (DREVERSE (CDR (REVERSE E)
(LIST (QUOTE OPTION)
(COND
((CDR E)
(CONS (QUOTE EITHER)
E))
(T (CAR E]
(T (CONS (QUOTE EITHER)
E]
[(REPEAT OPTION)
(LIST (LIST (CAR E)
(SEP1 (CDR E]
[LIST (LIST (LIST (CAR E)
(SEP1 (LIST (CADR E)))
(SEP1 (CDDR E]

```



```

(COND
  [(NULL (CDR E))
   (PROG [(N (GETP (CAR E))
              (QUOTE SENAME))
          (RETURN (SELECTQ N
                    (NIL (LIST (SENAME E)))
                    (NONE (MAPCONC (PARSEDEF (CAR E))
                                   (FUNCTION SEP2)))
                    (LIST N))
          (T (LIST (CONS (QUOTE TEST)
                        E)))]

```



```

(LEXSTAT
  [LAMBDA (S FLAG)
    (LEXLN S STLEXA (OR (LISTP FLAG)
                        (AND FLAG STLEXL)])

```

```

(PARSE
  [LAMBDA (LN GOAL)
    (PROG (STACK)
      [COND
        ((STRINGP LN)
         (SETQ LN (LEXSTAT LN T))
         (RETURN (COND
                  ((EQUAL (QUOTE ($EOL))
                          (SETQ LN (PARSEGOAL LN GOAL)))
                   (CAR STACK)]

```

```

(EDITLINE
  [LAMBDA (L FILE)
    (READSTAT FILE)]

```

```

(READSTAT
  [LAMBDA (FILE)
    (PROG [(L1 (TRIM (READLN FILE))
              (RETURN (COND
                      ((STRPOS (QUOTE "..")
                               L1 -2)
                       (CONCAT (SUBSTRING L1 1 -3)
                               (READSTAT FILE)))
                      (T L1)]

```

```

(UNPLUS
  [LAMBDA (E P)
    (PROG ((C (CADR E)))
      (UNPOSL C (CDDR E)
              P)
      [MAPC (CDDR E)
            (FUNCTION (LAMBDA (X)
                      (COND
                        ((NOT (UNPOSP X))
                         (UNOPR X P))
                        (COND
                          ((MINUSP C)
                           (UNOPR C P))

```

(OUTPARSE  
 [λ (E FMTA SPA FILE)  
 (UNPARSE E FMTA SPA FILE])

```

(UNPOSP
  [LAMBDA (X)
    (NOT (AND (LISTP X)
              (EQ (CAR X)
                  (QUOTE TIMES))
              (MINUSP (CADR X))

```

```
(UNPOSL
[LAMBDA (C L P)
  (PROG (FLAG)
    [MAPC L (FUNCTION (LAMBDA (X)
      (COND
        ((UNPOSP X)
          (COND
            (FLAG (OUTQ +)))
            (UNOPR (SETQ FLAG X)
              P]
        ]
      [COND
        ((IGREATERP C 0)
          (AND FLAG (OUTQ +))
          (OUT (SETQ FLAG C)
            (RETURN FLAG])
        ]
      ]
    ]
  )
)
```

```
(UNCFAC
[LAMBDA (E P)
  (PROG ((C (CADR E)))
    (SELECTQ C
      (1)
      (-1 (OUTQ -))
      (PROGN (UNOPR C (ADD1 P))
        (OUTQ *)))
    (RETURN E])
)
```

```
(UNREL
[LAMBDA (E P R C1 R1)
  (PROG ((C (CADDR E))
    (E1 (CADR E))
    F FLAG)
    (COND
      ((EQ C C1)
        (SETQ C 0)
        (SETQ R R1)))
    (COND
      ([SETQ F (AND (LISTP E1)
        (EQ (CAR E1)
          (QUOTE PLUS]
        (UNPOSL 0 (CDDR E1)
          P)
        )
        (T (UNOPR E1 P)))
      (OUT R)
      [COND
        (F (MAPC (CDDR E1)
          (FUNCTION (LAMBDA (X)
            (COND
              ((NOT (UNPOSP X))
                (COND
                  (FLAG (OUTQ +)))
                  (UNOPR (SETQ FLAG
                    (COND
                      ((AND (EQ (CADR X)
                        -1)
                        (NULL (CDDR X)))
                    (CADDR X))
                    (T.(FRPLACA (CDR *TIMES*)
                      (IMINUS (CADR X)))
                    (FRPLACD (CDR *TIMES*)
                      (CDDR X))
                    *TIMES*)))
                ]
              ]
            ]
          ]
        (COND
          ((OR (NULL FLAG)
            (MINUSP C))
            (UNOPR C P))
          ((NEQ C 0)
            (OUTQ +)
            (OUT C])
          ]
        ]
      ]
    ]
  )
)
```

(\* Always prints something)

```
(UNCO.ID
[LAMBDA (E P)
(MAP (CDR E)
[FUNCTION (LAMBDA (X)
(QUITQ CASE)
(UNOPR (CAR X)
P)
(QUITQ :)
(OUTQ %)
(UNOPR (CADR X)
P)
(COND
((CDDR X)
(OUTQ ;]
(FUNCTION CDDR])
```

```
(UNQT
[LAMBDA (X)
(QUITQ ')
(SETQ *SP*(COND
((LISTP (PRINT (CADR X)*FILE*))
17)
(T 26])
)
(RPAQ LEXBRKLIST
(! # $ % & ' % ( %) * + , - % . / : ; < = > ? @ % [ \ % ] + + { % | } ~))
(RPAQ LEXCLIST ((% . SEPR)
(%" . QUOTE)
(% . SEPR)))
[RPAQ SYSPRINTLIST ((LISPXBLOCK (PRINT . NPRINT))
(BREAK1BLOCK (PRINT . NPRINT)
(RPAQ *MINUS* (LIST (QUOTE MINUS)
0))
(RPAQ *TIMES* (LIST (QUOTE TIMES)
0))
[DECLARE: EVAL@COMPILE
(ADDOVAR GLOBALVARS STLEXA STLEXL *MINUS* *TIMES*)
]
[DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
(BLOCK: NIL PRINTEST PCTX PCTXS SPECTX EXPRIN SENAME VEPSET SEPRINT SEP1 SEP2
LEXSTAT PARSE EDITLINE READSTAT)
(BLOCK: PSBLOCK UNPARSE UNOPR UNPLUS UNPOSP UNPOSL UNCFAC UNREL UNCOND UNQT
OUT (ENTRIES UNPARSE)
(BLKAPPLYFNS UNPLUS UNCFAC UNREL UNCOND UNQT))
]
(LISPXPRINT (QUOTE (VEPSET))
T)
(VEPSET)
[DECLARE: DONTCOPY
(FILEMAP (NIL (1669 7724 (PRINTEST 1681 . 1853) (PCTX 1857 . 2017) (PCTX
2021 . 2180) (SPCTX 2184 . 2356) (EXPRIN 2360 . 2452) (SENAME 2456 . 2653)
(VEPSET 2657 . 3132) (SEPRINT 3136 . 3722) (SEP1 3726 . 3866) (SEP2 3870 .
4868) (LEXSTAT 4872 . 4961) (PARSE 4965 . 5196) (EDITLINE 5200 . 5251) (READSTAT
5255 . 5468) (UNPLUS 5472 . 5723) (UNPOSP 5727 . 5840) (UNPOSL 5844 . 6156)
(UNCFAC 6160 . 6334) (UNREL 6338 . 7293) (UNCOND 7297 . 7546) (UNQT 7550 .
7721))))))
STOP
```

(\* edited (25-FEB-75 . 1319))

(FILECREATED " 8-AUG-74 10:35:32" VPP.;52 16527

changes to: PCTX PCTXX SPCTX PQSUB PXVAR PSUBL PGOAL PXF CPRINT PNCLAUSE  
PCLAUSE1 PPUT1 PFNVP PFNPP

previous date: "26-JUL-74 00:56:59" VPP.;51)

```

(LISXPRT (QUOTE VPPCOMS)
  T)
(RPAQQ VPPCOMS
  ((PROP TRACEGROUP T XX X E GOAL GC PUT FINPUT G C)
  (PROP PPFN THCLAUSEP)
  (ENDUMP VPP)))
(DEFLIST(QUOTE(
  [T ((PPROOF AFTER NIL (COND ((EQ CTX TOPCTX)
    (LISTCLAUSES CTX)
  [XX (X (XLVAR AFTER NIL (COND ((NEQ !VALUE XLE)
    (PXVAR XLE !VALUE]
  [X ([XLVAL AFTER NIL (COND ((NEQ !VALUE XLE)
    (PXLV XLE !VALUE]
    (FNVRET BEFORE NIL (PFNV E VAL H CTX]
  [E ((CEVRR BEFORE NIL (PCEVRR E R V CTX))
    [CEVTV AFTER NIL (COND ((NEQ !VALUE E)
      (PCEVTV E !VALUE CTX]
    (TESTRR BEFORE NIL (PTESTRR R D L V CTX]
  [GOAL ((CGOAL BEFORE NIL (PGOAL E (CURCTX))
    (CGOAL AFTER NIL (COND ((NULL !VALUE)
      (PRINTQ "No proof needed, Eval(goal) = T"]
  [GC ((XLPLY2 BEFORE NIL (PCCTX (CURCTX]
  [PUT ((PUTX BEFORE NIL (PPUT E P V CTX]
  [FINPUT ((ADDFNVALUE BEFORE NIL (PFNVP E VAL))
    (ADDFNPATTERN BEFORE NIL (PFNPP E VARS COND VAL]
  [G ((CGEXP AFTER NIL (PCGE A !VALUE]
  [C ((NEWCTX AFTER NIL (PGCTX !VALUE))
    (PROOFEND BEFORE NIL (PPRET VAL CTX]
  ))(QUOTE TRACEGROUP))
  (DEFLIST(QUOTE(
    (THCLAUSEP PTHCP) (FNVAL NILL)
  ))(QUOTE PPFN))
  [RPAQQ VPPPROPS ((PPFN (NOSET))
    (JFORM (NOSET))
    (HFORM (NOSET))
  (RPAQQ VPPFNS
    (PQSUB PXVAR PCGE PLCN PEQS PVSUB PFAM PAL PSUBL PGOAL PCCTX PXLV PXF
    ETERPRI PXFORM PGCTX PPROOF PPRET PJUST JDUMP JPRINT PCEFF
    CDUMP CPRINT PRINTHM LISTCLAUSES LISTCLL GNCLAUSES PNCLAUSE
    PNUMCL PCLAUSE1 PCN CLNO PCLNO LISTCSL LISTPH PCEV1 PCEVTV
    PCEVRR PTESTRR PFNV PPUT PPUT1 PTHCP PFNVP PFNPP PLIS PTRY PRPC
    PNTN PLOTCE GETCE))
  (RPAQQ VPPVARS (TRACEPROPS))
  (APPLYMAPC VPPPROPS (FUNCTION NEWPROP))
(DEFINEQ

```

*✓ Editfix, recompile*

*✓*



(PQSUB  
 [LAMBDA (V X E Y)  
 (PRINQ %[ )  
 (NPRIN2 V)  
 (PRINQ =>)  
 (NPRIN2 X)  
 (PRINQ :% )  
 (PXF E Y)  
 (PRINTQ %]])

(OPTRACE  
 [X (OP ARGS VALUE)  
 (PRIN1 OP)  
 (MAPRINT ARGS NIL ?( ?) , (F' NPRIN2))  
 (PRINQ ?%=>?%)  
 (ETERPRI VALUE 1)  
 (NPRINT VALUE)]

(PXVAR  
 [LAMBDA (V E)  
 (FORMPRIN\* (QUOTE ("CVAL("NPRIN2 ") = " NPRINT))  
 V E])

(PCGE  
 [LAMBDA (A E)  
 (COND  
 (E (SELECTQ (CAR E)  
 (GOAL (TERPRI))  
 NIL)  
 (PPST A])

(PLCN  
 [LAMBDA (A)  
 (PRIN1 (TLNO A])

(PEQS  
 [LAMBDA (S L)  
 (PRINTQ Equalities:)  
 (MAP2C (REVERSE S)  
 (REVERSE L)  
 (FUNCTION (LAMBDA (X Y)  
 (TAB 2)  
 (PRINQ %(from% )  
 (PCN Y)  
 (PRINQ %))  
 [SPACES (MAX 1 (IDIFFERENCE 15 (POSITION)  
 (PVSUB X])

(PVSUB  
 [LAMBDA (Z)  
 (PXF (CAR Z)  
 (CDR Z])

(PFAM  
 [LAMBDA (Q E AL SCORE)  
 (FORMPRIN\*(QUOTE ("Matching " PCN " against " PCN " (score=" PRIN2 "):" PAL)  
 )  
 E Q SCORE AL])

(PAL  
 [LAMBDA (AL)  
 (MAPC AL (FUNCTION (LAMBDA (Z)  
 (TAB 2)  
 (PVSUB Z])

```
(PSUBL
 [LAMBDA (AL)
  (MAP AL (FUNCTION (LAMBDA (Z)
    (COND
      ((NEQ Z AL)
       (PRINQ ,% )
       (ETERPRI (CDAR Z)
                3)))
      (NPRIN2 (CAAR Z))
      (PRINQ =)
      (NPRIN2 (CDAR Z]))
```

```
(PGOAL
 [LAMBDA (E CTX)
  (PRINQ Goal:% )
  (NPRINT E)]
```

```
(PCCTX
 [LAMBDA (CTX)
  (PRINQ In% )
  (PCTX CTX)
  (PRINTQ :)]
```

```
(PXLV
 [LAMBDA (E E1)
  (PRINQ Eval:% )
  (PXF E E1)]
```

```
(PXF
 [LAMBDA (E E1)
  (NPRIN2 E)
  (ETERPRI E1 4)
  (PRINQ % =>% )
  (NPRIN2 E1)]
```

```
(ETERPRI
 [LAMBDA (E N)
  (COND
    ((IGREATERP (IPLUS (POSITION)
                       (ITIMES 2 (COUNT E))
                       N)
                 (LINELENGTH))
     (TAB 3)
     T])
```

```
(PXF E E1)
 (PROG1 (PXF E E1)
 (ETERPRI])
```

```
(PGCTX
 [LAMBDA (CTX)
  (PRINQ New% )
  (PCTXX CTX)
  (JPRINT CTX)]
```

(PPROOF

```
[LAMBDA (CTX)
  (PRINQ Proving% )
  (PCTX CTX)
  (PRINTQ ...)]
```

(PPRET

```
[LAMBDA (VAL CTX)
  (SELECTQ (CAR VAL)
    (T (PRINQ Success% in% ))
    (NIL (PRINQ Failure% in% ))
    (CX (PRINQ Counterexample% in% ))
    (HELP))
  (PCTX CTX)
  (TERPRI)]
```

(PJUST

```
[LAMBDA (CTX)
  (PROG ((JA (CJUST CTX))
    FORM)
    (BINDCTX CTX (COND
      ((SETQ FORM (JFORM (CAR JA)))
        (FORMPRIN1 FORM (CDR JA)))
      (JA (APPLY (CAR JA)
        (CDR JA))
```

(JDUMP

```
[LAMBDA (CTX UP DOWN)
  (COND
    ((NULL CTX)
      (SETQ CTX (ROOTCTX))
      (SETQ DOWN T))
    ((NUMBERP CTX)
      (SETQ CTX (CN CTX))
      (SETQ DOWN T)))
  (PROG ([D (OR (NUMBERP DOWN)
    (COND
      (UP (ITIMES 2 (FLENGTH CTX)))
      (T 2]
    POS)
  (COND
    ((AND UP (CDR CTX))
      (JDUMP (CDR CTX)
        T NIL)))
  (TAB D)
  (PCTX CTX)
  (SETQ POS (POSITION))
  (JPRINT CTX)
  (COND
    ((OR (PROOFDONE CTX)
      (CEFFORT1 CTX))
      (TAB (ADD1 POS))
      (PCEFF CTX)))
  (COND
    (DOWN (MAPC (SONCTX CTX)
      (FUNCTION (LAMBDA (X)
        (JDUMP X NIL (IPLUS D 2]))
```

```
(JPRINT
 [LAMBDA (CTX)
 (SPACES 1)
 (PJUST CTX)
 (TERPRI])
```

```
(PCEFF
 [LAMBDA (CTX)
 (PROG ((V (PROOFDONE CTX))
 (EFF (CEFFORT CTX)))
 (COND
 (V (SELECTQ (CAR V)
 (T (PRINQ Proved))
 (NIL (PRINQ No% proof))
 (CX (PRINQ Counterexample))
 (HELP)))
 ((NULL EFF)
 (PRINTQ Not% attempted)
 (RETURN))
 (T (PRINQ Being% proved))))
 (COND
 (EFF (PRINQ :% )
 (PEFF EFF)))
 (TERPRI])
```

```
(CDUMP
 [LAMBDA (FLAG PROPS)
 (COND
 ((NUMBERP FLAG)
 (CPRINT FLAG PROPS))
 (T (PROG (CL)
 (SPACES 2)
 [PRINTGRAPH (ELT CONTEXTS 1)] (ROOTCTX)
 (FUNCTION SONCTX)
 [FUNCTION (LAMBDA (X)
 (PCTX X)
 (SETQ CL (NCONC1 CL X]
 (FUNCTION (LAMBDA (X)
 (PRINQ --]
 (AND FLAG (MAPC CL (FUNCTION (LAMBDA (CTX)
 (COND
 ([OR (EQ FLAG (QUOTE ALL))
 (NOTANY CL (FUNCTION (LAMBDA (X)
 (EQ CTX (CDR X]
 (PRINT CTX)
 (CPRINT CTX PROPS]))
```

```

(CPRINT
 [LAMBDA (CTX PROPS)
 [COND
 ((NUMBERP CTX)
 (SETQ CTX (CN CTX)
 (MAPC (SELECTQ PROPS
 (T TRACEPROPS)
 (NIL PROPERTIES)
 (OR (LISTP PROPS)
 (LIST PROPS)))
 (FUNCTION (LAMBDA (PROP)
 (PROG (L (N (FLENGTH CTX)))
 [MAPPROP PROP CTX
 (FUNCTION (LAMBDA (X Y)
 (SETQ L
 (CONS (CONS X (FLENGTH (GETXCX X PROP CTX)))
 L]
 (COND
 (L (SPACES 2)
 (PRINT PROP)
 (MAPC [SORT L (FUNCTION (LAMBDA (X Y)
 (ILESSP (CDR X)
 (CDR Y)
 (FUNCTION (LAMBDA (Z)
 (PROG [(M (DIFFERENCE N (CDR Z)
 (FRPTQ M (PRINQ *))
 (OR (GEQ M 4)
 (TAB 4))
 (NPRIN2 (CAR Z))
 (PRINQ ".....")
 (NPRINT (GETX (CAR Z)
 PROP CTX))
 (PRINTM
 [LAMBDA (L PROPS)
 (NMAPCAR L (FUNCTION (LAMBDA (X)
 (PROG ((Y (CN X)))
 (PRINT Y)
 (RETURN (COND
 ((PROOFDONE Y)
 (PRINT (PROOFDONE Y))
 NIL)
 (T (LISTCLAUSES Y)
 (CPRINT X PROPS)
 X])
 (LISTCLAUSES
 [LAMBDA (CTX XL)
 (LISTCLL (GNCLAUSES CTX XL)
 CTX])

```

```

(COND
 (L (SPACES 2)
 (PRINT PROP)
 (MAPC [SORT L (FUNCTION (LAMBDA (X Y)
 (ILESSP (CDR X)
 (CDR Y)
 (FUNCTION (LAMBDA (Z)
 (PROG [(M (DIFFERENCE N (CDR Z)
 (FRPTQ M (PRINQ *))
 (OR (GEQ M 4)
 (TAB 4))
 (NPRIN2 (CAR Z))
 (PRINQ ".....")
 (NPRINT (GETX (CAR Z)
 PROP CTX))
 (PRINTM
 [LAMBDA (L PROPS)
 (NMAPCAR L (FUNCTION (LAMBDA (X)
 (PROG ((Y (CN X)))
 (PRINT Y)
 (RETURN (COND
 ((PROOFDONE Y)
 (PRINT (PROOFDONE Y))
 NIL)
 (T (LISTCLAUSES Y)
 (CPRINT X PROPS)
 X])
 (LISTCLAUSES
 [LAMBDA (CTX XL)
 (LISTCLL (GNCLAUSES CTX XL)
 CTX])

```

```

(CPRINT1
 [λ (X PROP CTX M)

```

```

(CPRINT1 (CAR Z)
 PROP CTX
 (IDIF ' N (CDR Z)))

```

```

(FUNCTION (LAMBDA (Z)
 (PROG [(M (DIFFERENCE N (CDR Z)
 (FRPTQ M (PRINQ *))
 (OR (GEQ M 4)
 (TAB 4))
 (NPRIN2 (CAR Z))
 (PRINQ ".....")
 (NPRINT (GETX (CAR Z)
 PROP CTX))
 (PRINTM
 [LAMBDA (L PROPS)
 (NMAPCAR L (FUNCTION (LAMBDA (X)
 (PROG ((Y (CN X)))
 (PRINT Y)
 (RETURN (COND
 ((PROOFDONE Y)
 (PRINT (PROOFDONE Y))
 NIL)
 (T (LISTCLAUSES Y)
 (CPRINT X PROPS)
 X])
 (LISTCLAUSES
 [LAMBDA (CTX XL)
 (LISTCLL (GNCLAUSES CTX XL)
 CTX])

```

```

(PRINTM
 [LAMBDA (L PROPS)
 (NMAPCAR L (FUNCTION (LAMBDA (X)
 (PROG ((Y (CN X)))
 (PRINT Y)
 (RETURN (COND
 ((PROOFDONE Y)
 (PRINT (PROOFDONE Y))
 NIL)
 (T (LISTCLAUSES Y)
 (CPRINT X PROPS)
 X])
 (LISTCLAUSES
 [LAMBDA (CTX XL)
 (LISTCLL (GNCLAUSES CTX XL)
 CTX])

```

```

(LISTCLAUSES
 [LAMBDA (CTX XL)
 (LISTCLL (GNCLAUSES CTX XL)
 CTX])

```



```
(CLNO
 [LAMBDA (X CTX)
  (OR CTX (SETQ CTX (CURCTX)))
  (OR (CLAUSENO X CTX)
      (AND (CDR CTX)
           (CLNO X (CDR CTX]))
```

```
(PCLNO
 [LAMBDA (M W)
  (COND
   (M (PRINW (CI (CAR M))
            W)
      (PRINQ %.)
      (PRIN2 (CDR M)))
   (T (PRINW (QUOTE -)
            W)
      (PRINQ .-]))
```

```
(LISTCSL
 [LAMBDA (CTX)
  [PSUBL (SUBFIX (MAPCON CTX (FUNCTION (LAMBDA (C)
                                         (APPEND (CSL C)
                                                (TERPRI]))
```

```
(LISTPH
 [LAMBDA (CTX)
  (PROG ((I (CI CTX))
        (L (CPH CTX))
        (N 1))
    (MAPREDC L (FUNCTION (LAMBDA (X Y)
                          (PRINW I 4)
                          (PRINQ :)
                          (PRIN2 N)
                          (TABMIN 8 1)
                          (FORMPRIN1 (HFORM (CAR X))
                                       (CDR X))
                          (TERPRI)
                          (ADDIVAR N]))
```

```
(PCEV1
 [LAMBDA (V X)
  (PRINQ From% )
  (PCN X CTX)
  (PRINQ :% )
  (PXFOM E V]) (* USES E, CTX FREE)
```

```
(PCEVTV
 [LAMBDA (E V CTX)
  (PCEV1 V (COND
   (V E)
   (T (NOTOF E]))
```

```
(PCEVRR
 [LAMBDA (E X V CTX)
  (PCEV1 V X])
```

```

(PTESTRR
[LAMBDA (E D L V CTX)
[PROG ((M (QUOTE From% )))
(MAP (NCONS (NLIST D)
L)
(FUNCTION (LAMBDA (Z)
(MAP (CDDAR (CAR Z))
(FUNCTION (LAMBDA (Y)
(COND
([NOT (SOME (CDR Z)
(FUNCTION (LAMBDA (X)
(FMEMB (CAR Y)
(CDDAR X]
(PRIN1 M)
(SETQ M (QUOTE " & "))
(PCN (CAR Y)
CTX]
(PRINQ :% )
(PXFORM E V])

```

```

(PFNV
[LAMBDA (E V H CTX)
[PROG ((SEP (QUOTE From% )))
[MAPC H (FUNCTION (LAMBDA (X)
(PRIN1 SEP)
(PCN X CTX)
(SETQQ SEP " & "]
(PRINQ :% )
(PXFORM E V])

```

```

(PPUT
[LAMBDA (E P V CTX)
(COND
((FMEMB P TRACEPROPS)
(APPLY* (OR (PPFN P)
(FUNCTION PPUT1))
E P V CTX])

```

```

(PPUT1
[LAMBDA (E P V CTX)
(COND
((AND (LISTP V)
(EQ (CDR V)
(GETX E P CTX)))
(PRINQ Add% )
(SETQ V (CAR V)))
(T (PRINQ Set% )))
(FORMPRIN* (QUOTE (PRIN2 "(" NPRINT2 ")";" NPRINT))
P E V])

```

```

(PTHCP
[LAMBDA (E P V CTX)
(COND
((EQ E T))
(V (PNCLAUSE E CTX))
(T (PRINQ Cancel% )
(PCLAUSE1 E CTX])

```



(PFNVP

[LAMBDA (E VAL)

(FORMPRIN\* (QUOTE ("Record " NPRIN2 " = " NPRINT))  
E VAL])

(PFNPP

[LAMBDA (E VARS COND VAL)

(FORMPRIN\* (QUOTE ("Record " NPRIN2 " (for " PLIS "|" NPRIN2 ") = " NPRINT))  
E VARS COND VAL])

(PLIS

[LAMBDA (L FN)

(MAPRINT L NIL NIL NIL (QUOTE ,)  
FN])

(PTRY

[LAMBDA (M E)

(FORMPRIN\*(QUOTE ("Try " PRIN1 " on " PCLAUSE1))  
M E])

(PRPC

[LAMBDA (X Y J)

(SELECTQ J

(SFSUBST (PRINTQ Expanding% special% functions:))  
(EVBOOL (PRINTQ Re-evaluating% disjunction:))  
(PROGN (PRINTQ Using% )  
(PRIN2 J)  
(PRINTQ :])

(PNTH

[LAMBDA (N)

(PRIN1 N)

(PRIN1 (COND

((NEQ (IQUOTIENT (IREMAINDER N 100)  
10)

1)

(SELECTQ (IREMAINDER N 10)

(1 (QUOTE st))

(2 (QUOTE nd))

(3 (QUOTE rd))

(QUOTE th)))

(T (QUOTE th])



(PCGE (8-AUG-74 . 1032))  
(PLCN (8-AUG-74 . 1032))  
(PEQS (8-AUG-74 . 1032))  
(PVSUB (8-AUG-74 . 1032))  
(PFAM (8-AUG-74 . 1032))  
(PAL (8-AUG-74 . 1032))  
(PSUBL (8-AUG-74 . 1032))  
(PGOAL (8-AUG-74 . 1032))  
(PCCTX (8-AUG-74 . 1032))  
(PXLV (8-AUG-74 . 1032))  
(PXF (8-AUG-74 . 1033))  
(ETERPRI (8-AUG-74 . 1033))  
(PXFOM (8-AUG-74 . 1033))  
(PGCTX (8-AUG-74 . 1033))  
(PPROOF (8-AUG-74 . 1033))  
(PPRET (8-AUG-74 . 1033))  
(PJUST (8-AUG-74 . 1033))  
(JDUMP (8-AUG-74 . 1033))  
(JPRINT (8-AUG-74 . 1033))  
(PCEFF (8-AUG-74 . 1033))  
(CDUMP (8-AUG-74 . 1033))  
(CPRINT (8-AUG-74 . 1034))  
(PRINTHM (8-AUG-74 . 1034))  
(LISTCLAUSES (8-AUG-74 . 1034))  
(LISTCLL (8-AUG-74 . 1034))  
(GNCLAUSES (8-AUG-74 . 1034))  
(PNCLAUSE (8-AUG-74 . 1034))  
(PNUMCL (8-AUG-74 . 1034))  
(PCLAUSE1 (8-AUG-74 . 1034))  
(PCN (8-AUG-74 . 1034))  
(CLNO (8-AUG-74 . 1034))  
(PCLNO (8-AUG-74 . 1034))  
(LISTCSL (8-AUG-74 . 1034))  
(LISTPH (8-AUG-74 . 1034))  
(PCEV1 (8-AUG-74 . 1034))  
(PCEVTV (8-AUG-74 . 1034))  
(PCEVRR (8-AUG-74 . 1034))  
(PTESTRR (8-AUG-74 . 1034))  
(PFNV (8-AUG-74 . 1034))  
(PPUT (8-AUG-74 . 1034))  
(PPUT1 (8-AUG-74 . 1035))  
(PTHCP (8-AUG-74 . 1035))  
(PFNVP (8-AUG-74 . 1035))  
(PFNPP (8-AUG-74 . 1035))  
(PLIS (8-AUG-74 . 1035))  
(PTRY (8-AUG-74 . 1035))  
(PRPC (8-AUG-74 . 1035))  
(PNTH (8-AUG-74 . 1035))  
(PLOTCE (8-AUG-74 . 1035))  
(GETCE (8-AUG-74 . 1035))  
) (QUOTE EDITDATE))  
(RPAQQ TRACEPROPS (THCLAUSEP CVAL FNDEF))  
(DECLARE: DONTCOPY  
(FILEMAP (NIL (1889 14895 (PQSUB 1901 . 2042) (PXVAR 2046 . 2138) (PCGE 2142

. 2281) (PLCN 2285 . 2327) (PEQS 2331 . 2656) (PVSUB 2660 . 2724) (PFAM 2728  
. 2873) (PAL 2877 . 2982) (PSUBL 2986 . 3193) (PGOAL 3197 . 3260) (PCCTX 3264  
. 3338) (PXLV 3342 . 3406) (PXF 3410 . 3506) (ETERPRI 3510 . 3722) (PXF  
3726 . 3796) (PGCTX 3800 . 3878) (PPROOF 3882 . 3964) (PPRET 3968 . 4202)  
(PJUST 4206 . 4494) (JDUMP 4498 . 5356) (JPRINT 5360 . 5432) (PCEFF 5436 .  
5982) (CDUMP 5986 . 6883) (CPRINT 6887 . 7706) (PRINTHM 7710 . 8182) (  
LISTCLAUSES 8186 . 8271) (LISTCLL 8275 . 8734) (GNCLAUSES 8738 . 9067) (PNCLAUSE  
9071 . 9299) (PNUMCL 9303 . 9520) (PCLAUSE1 9524 . 9603) (PCN 9607 . 9671)  
(CLNO 9675 . 9816) (PCLNO 9820 . 10021) (LISTCSL 10025 . 10146) (LISTPH 10150  
. 10456) (PCEV1 10460 . 10609) (PCEVTV 10613 . 10709) (PCEVRR 10713 . 10761)  
(PTESTRR 10765 . 11518) (PFNV 11522 . 11736) (PPUT 11740 . 11906) (PPUT1 11910  
. 12145) (PTHCP 12149 . 12290) (PFNVP 12294 . 12392) (PFNPP 12396 . 12540)  
(PLIS 12544 . 12617) (PTRY 12621 . 12710) (PRPC 12714 . 12930) (PNTH 12934  
. 13325) (PLOTCE 13329 . 14627) (GETCE 14631 . 14892))))))  
STOP

(FILECREATED " 5-AUG-74 13:14:40" VSS.;46 16201

changes to: VSSCOMS VMODEVAR VMODEFN DATASTAT

previous date: " 3-AUG-74 18:18:31" VSS.;44)

(LISPXPRT (QUOTE VSSCOMS)  
T)

*✓ Editfix, 1 fix*

(RPAQQ VSSCOMS

((PROP REFOP UAELT URELT)

(PROP (PARSEDEF SENAME)

PROCDEC DECDEC USELIST DECSTAT MODEC DATASTAT ~~DECSTAT~~ DECITEM

CDECITEM FNDEC MODEEXPR STRUCLIST APARS OPTPARS XSTAT STEPTAIL

SIMSTL GOSTAT SETSTAT + SIMTAIL LABEL IDLIST PARLIST PARSPEC

VALSPEC LHS PRIM CPRIM MID RELATION RELCL NOREL RELDISJ RELCONJ

GEXPR QRLIST FATAIL EXTAIL QUTAIL QUTEST QUSEP TERM FACTOR

PRIM1 EXPR RELEXPR QUREL AEXPR)

~~(PROP (OPTIMIZE T) (MACRO) (SFDEF~~

~~UIMP UGREATERP ULESSP)~~

(PROP STTL ASRT DECL NONE DECVAR DECMODE ASSERT DECLARE CANCEL)

(PROP STLEVADJ WHILE END BLKIF BLKELSE LOOP BLKELSEIF BEGIN UNTIL  
STEP FOR)

(PROP STFMT \*OTHER\* PLUS TIMES MINUS QUOTIENT REMAINDER SETQ EXPT  
IEXPT EL ELT LENGTH ABS SUBSEQ QUOTE LCOND EQUAL NEQUAL GEQ LEQ  
EQUALP NEQUALP FA FU EX CHOICE BV EQV NEQV OR AND NOT VAR  
SEQMODE STRUCMODE PTRMODE ATMODE CONSTSEQ CONSTSEQ\* CONSTRUCT  
UNDEFINED)

(VARS \* (MAPCAR STSPNAMES (FUNCTION CAR)))  
(ENDUMP VSS)))

(DEFLIST(QUOTE(  
(UAELT T)  
(URELT T)  
)))(QUOTE REFOP))

(DEFLIST(QUOTE(  
→ (PROCDEC (PROCEDURE \$ID ((PARSPEC)

(VALSPEC)

~~(# PARCONS 3)~~

(USELIST)))

*(: QUOTE 1)*

→ (DECDEC (DECLARATIONS \$ID ((PARSPEC)

(: NIL)

~~(# PARCONS 3)~~

(USELIST)))

*(# CONSWAP 2)  
(# CONS 2)*

→ [USELIST ((EITHER (USING (IDLIST))

((: NIL]

→ [DECSTAT ((EITHER ((MODEC)

(DECL (\$ 0)

(LIST , (DECITEM))

(: DECVAR \$]

```

(MODEC (MODE $ID (EITHER (= (MODEXPR)
                          (: CVAR 2))
                          [%($ 0)
                          (LIST , $ID (: (MODEVAR MODE))
                          (: VAR 2))
                          (: BV $)
                          (EITHER (= (MODEXPR)
                                      (: FUNCTION 2)
                                      (: FNVAR 2))
                                  ((# VMODEFN 2]
                                  ((# VMODEVAR 1)))
                          (: DECVAR 1)))

```

```

[DATASTAT ((EITHER ((MODEC))
                   (DECL ($ 0)
                   (LIST , (CDECITEM))
                   (: DECVAR $)
                   (ASSERT (QRLIST)
                   (: ASSERT 1]

```

```

[DECLIST ((DECITEM)
          (REPEAT , (DECITEM))
[DECITEM ($ID (EITHER (: (MODEXPR)
                       (: VAR 2))
                       (= (GEXPR)
                       (: CVAR 2))
                       ((*)
                       (GEXPR)
                       (: IVAR 2))
                       ((FNDEC)

```

```

[CDECITEM ($ID (EITHER (= (GEXPR)
                          (: CVAR 2))
                          ((FNDEC)
                          (: (MODEXPR) (: VAR 2))

```

```

(FNDEC (%($ 0)
        (PARLIST)
        %)
        (: BV $)
        =
        (GEXPR)
        (: FUNCTION 2)
        (: FNVAR 2)))
[MODEXPR ((EITHER (SEQ %($ 0)
                  %)
              (: SEQMODE 1))
          (PTR %($ 0)
          %)
          (: PTRMODE 1))
          (STRUCT %($ 0)
          (STRUCLIST)
          %)
          (: STRUCMODE $))
          ($ID (EITHER (%($ 1)
                      (LIST , (MODEXPR))
                      %)
                  (: $)
                  (: MODECALL 1))
          ((: MODEVAR 1]

```

```

[STRUCLIST ((LIST , (EITHER ($ (: $))
                          ($ID : (MODEXPR)
                          (: VAR 2]

(APARS (($ 1)
        (OPTION (EXPR)
                (REPEAT , (EXPR)))
        (: $)))
[OPTPARS ((EITHER (%( (APARS)
                    %))
          ((; 1]
[XSTAT ((EITHER (ASSERT (ORLIST)
                  (: ASSERT 1))
             (LEMMA (ORLIST)
                    (: LEMMA 1))
             (ASSUME (ORLIST)
                    (: ASSUME 1))
             [IF (RELEXPR)
               THEN
                 (EITHER (BEGIN (: BLKIF 1))
                        (($ 1)
                        (SIMSTL)
                        (: IF $]
             (BEGIN (: (BEGIN)))
             (LOOP (: (LOOP)))
             [WHILE (RELEXPR)
               :
                 (EITHER (BEGIN (: BLKWHILE 1))
                        (($ 1)
                        (SIMSTL)
                        (: WHILE $]
             [UNTIL (RELEXPR)
               :
                 (EITHER (BEGIN (: BLKUNTIL 1))
                        (($ 1)
                        (SIMSTL)
                        (: UNTIL $]
             (REPEAT : BEGIN (: (BLKREPEAT)))
             (FOR $ID (←)
                 (EXPR)
                 (: FOR 2))
             (BY (EXPR)
                TO
                (EXPR)
                (STEPTAIL))
             (TO (EXPR)
                (EITHER (BY (EXPR))
                        ((: 1)))
                (STEPTAIL)
                (# RSTEP 1))

```

```

[ELSE (EITHER (BEGIN (: (BLKELSE)))
              [IF (RELEXPR)
                THEN
                (EITHER (BEGIN (: BLKELSEIF 1))
                        (($ 1)
                          (SIMSTL)
                          (: ELSEIF $])
                        (($ 0)
                          (SIMSTL)
                          (: ELSE $])
                (END (: (END)))
                (DECLARE (ORLIST)
                        (: DECLARE 1))
                (CANCEL (LABEL)
                        (: CANCEL 1))
                (%. %. %. (: (EMPTY)))
                ((GOSTAT)
                 (# CDR 1))
                ((SETSTAT)
                 (EITHER (; ($ 1)
                        (SIMSTL)
                        (: PROGN $])
                        ((# CDR 1]
                [STEPTAIL (: (EITHER (BEGIN (: BLKSTEP 2))
                                     (($ 2)
                                       (SIMSTL)
                                       (: STEP $])
                [SIMSTL ((EITHER ((GOSTAT))
                                ((SETSTAT)
                                (SIMTAIL]
                [GOSTAT ((EITHER [RETURN (: RETURN)
                                (EITHER ((GEXPR)
                                         (: STMT: 2))
                                         ((: STMT: 1]
                                ((EITHER (EXIT (: EXIT))
                                         (NEXT (: NEXT)))
                                (LABEL)
                                (: STMT: 2]
                (SETSTAT ((: SETQ)
                        (LHS)
                        (+)
                        (EXPR)
                        (: STMT: 3)))
                [+ ((EITHER (+)
                        (: =]
                [SIMTAIL ((OPTION ; (EITHER ((GOSTAT))
                                             ((SETSTAT)
                                             (SIMTAIL]
                (LABEL ($ID (# LITQ 1)))
                (IDLIST ($ID ($ 1)
                        (REPEAT , $ID)
                        (: $)))
                [PARLIST ((LIST , $ID : (MODEXPR)
                        (: VAR 2]

```



[PARSPEC ((EITHER (%( \$ 0)  
(PARLIST)  
%)  
(; \$))

(: NIL]  
[VALSPEC ((EITHER (: (MODEXPR))  
(: NIL]

(LHS ((GEXPR)  
(# CKLH 1)))

[PRIM ((EITHER [CHOICE %( (MID)  
(: BV 1)  
(QUSEP)  
(QUREL)  
(EITHER (: : (EXPR)  
%)  
(: CHOICE 3))  
(%) (# CHOICE2FORM 2]

[CONST %( (MODEXPR)  
(EITHER (SIZE (EXPR)  
(EITHER (INITIAL (GEXPR)  
(: CONSTSEQ 3))  
(: UNDEFSEQ 2)))  
%))

(INITIAL (\$ 1)  
(LIST , (GEXPR))  
%)  
(: CONSTSEQ\* \$))

(OF (\$ 1)  
(LIST , (GEXPR))  
%)  
(: CONSTRUCT \$]

(\$ID (EITHER (%( (APARS)  
%)  
(# CALLFORM 1))  
NIL))

(\$ (MID)  
(# QVSAVE 1))  
(% (GEXPR)  
%))  
(| (GEXPR)  
|  
(: UABS 1))  
(CPRIM))

(REPEAT (EITHER (%. (QID)  
(: URELT 2))  
%  
(EXPR)  
(EITHER (TO (EXPR)  
(: SUBSEQ 3))  
(: UAELT 2)))  
%]]

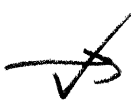
[CPRIM ((EITHER (\$NUM)  
( (EITHER (\$ID)  
(SPUNC))  
(: QUOTE 1]

(SEQ %( (MODEXPR) %)  
(: SEQMODE 1))  
(PTR %( (MODEXPR) %)  
(: PTRMODE 1))  
(STRUCT %( (\$ φ)  
(STRUCLIST) %)  
(: STRUCMODE \$))  
(PROC %( (: FMODE) (EITHER  
((MODEXPR) (\$ 1)  
(REPEAT , (MODEXPR))  
(; \$))  
(: NIL))) % : (MODEXPR)  
(# CONSWAP 2) (# CONS 2))  
(EXPR %( (\$ φ) (PARLIST)  
% ( : STRUCMODE \$)  
(GEXPR) (: FUNCTION 2))

(MID (SID : (MODEXPR) (: VAR 2)))

[RELATION ((EITHER (NOT (RELATION) (: NOT 1)) (%. FA %( (FATAIL)) (%. EX %( (EXTAIL)) ((AEXPR) (OPTION (RELCL) (\$ 3) (REPEAT (RELCL)) (; \$) (# RELCLGRP 1)

(RELCL ([EITHER (= (OPTION =) (EQUALP . \*) (: EQUALP)) (# (OPTION #) (: NEQUALP))



[< (EITHER (= (: LEQ) (: LESS) (NOT (GEQ . \*)))

(> (EITHER (= (: GEQ) (: GREATER) (NOT (LEQ . \*)))

(AEXPR)))

(NQREL ((RELDISJ)

(EITHER (EQV (RELDISJ) (: EQUALP 2))

(IMP (RELDISJ) (: NOT 1))



(NIL)))

*OR*

[RELDISJ ((RELCONJ)

(REPEAT (EITHER (OR) (!))

(RELCONJ)

(: OR 2]

[RELCONJ ((RELATION)

(REPEAT (EITHER (AND) (&))

(RELATION)

(: AND 2]

[GEXPR ((EITHER (IF (RELEXPR)

THEN

(GEXPR)

ELSE

(: T)

(GEXPR)

(: DCOND 4))

(TEST (GEXPR)

CASE

(\$ 0)

(REPEAT (CPRIM)

(\$ 1)

(REPEAT , (CPRIM))

(; \$)

:

(GEXPR)

;

(# CONS 2))

ENDCASE

(; \$)

(GEXPR)

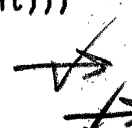

(# CASEXP 3))

((NQREL]

```

[QRLIST ((QUREL)
        (REPEAT , (QUREL)
          (: AND 2]
(FATAIL ((: FA)
        (QUTAIL)
        (# QUFIX 5)))
(EXTAIL ((: EX)
        (QUTAIL)
        (# QUFIX 5)))
(QUTAIL ((: *QV*)
        (# LIST 1)
        (QUTEST)
        ($ 1)
        (REPEAT , (QUTEST))
        (; $)
        (EITHER ((QUSEP)
                (QRLIST))
                ((: T)))
        %)
        (RELATION)))
[QUTEST ((EITHER ((TEST $ID :)
                (MID))
                ((GEXPR)
                (QUSEP (1))
[TERM ((FACTOR)
        (REPEAT (EITHER (* (FACTOR)
                          (: TIMES 2))
                        (/ (FACTOR)
                          (: QUOTIENT 2))
                        (MOD (FACTOR)
                          (: REMAINDER 2]
[FACTOR ((PRIM1)
        (REPEAT (EITHER (+)
                      (* *))
                (PRIM1)
                (: IEXPT 2]
[PRIM1 ((EITHER (- (PRIM1)
                  (: MINUS 1))
                ((PRIM]
(EXPR ((GEXPR)
      (* Mode checks will assure an INT result)))
(RELEXPR ((GEXPR)
      (* No longer worth checking for absence of quantifiers)))
(QUREL ((GEXPR)
      (* Mode checks will assure a BOOL result)))
[AEXPR ((TERM)
        (REPEAT (EITHER (+ (TERM)
                          (: PLUS 2))
                        (- (TERM)
                          (: DIFFERENCE 2]
)))(QUOTE PARSEDEF))
(DEFLIST(QUOTE(
  (PROCDEC <procedure-head>

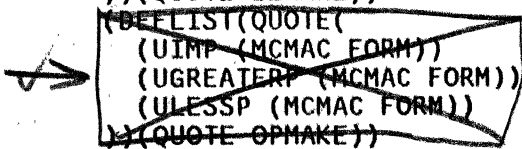
```

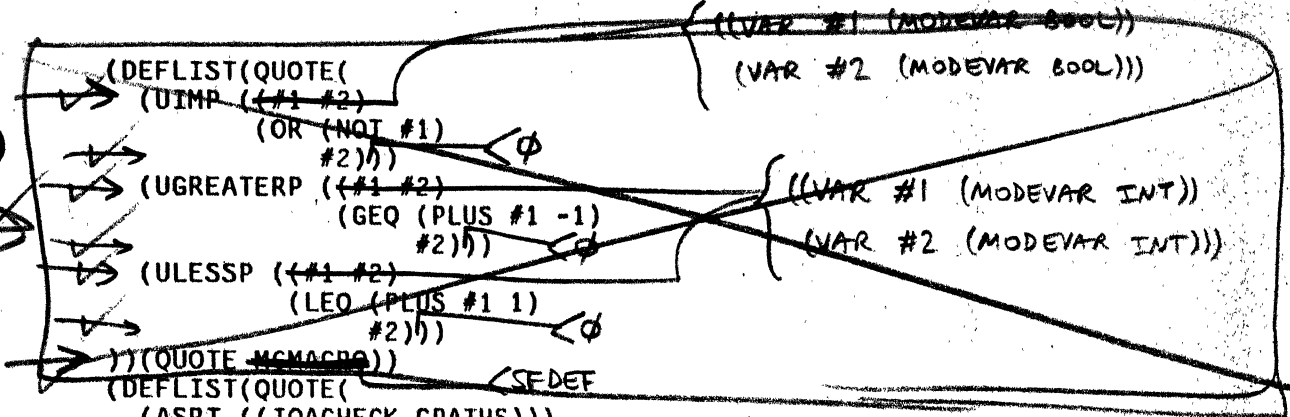
 (MODEXP  
 ((PRIM) (: MODEX '1)  
 (\* Mode checks will assure a MODE result)))

```

(DECDEC <declarations-head>)
(USELIST <environment-list>)
(DECSTAT <declaration>)
(MODEC <mode-declaration>)
(DATASAT <declarations-statement>)
(DECLIST NONE)
(DECITEM <declaration-item>)
(CDECITEM <constant-declaration-item>)
(FNDEC <function-declaration>)
(MODEXPR <mode-expression>)
(STRUCLIST <structure-list>)
(APARS <actual-parameters>)
(OPTPARS NONE)
(XSTAT <statement>)
(STEPTAIL NONE)
(SIMSTL <simple-statement-list>)
(GOSTAT <exit-statement>)
(SETSTAT <assignment-statement>)
(← NONE)
(SIMTAIL <simple-statement-tail>)
(LABEL <label>)
(IDLIST <id-list>)
(PARLIST <formal-parameter-list>)
(PARSPEC <parameter-spec>)
(VALSPEC <value-spec>)
(LHS <left-hand-side>)
(PRIM <primary>)
(CPRIM <constant-primary>)
(MID NONE)
(RELATION <relation-primary>)
(RELCL <relation-clause>)
(NQREL <implication>)
(RELDISJ <disjunction>)
(RELCONJ <conjunction>)
(GEXPR <general-expression>)
(QRLIST <relational-list>)
(FATAIL NONE)
(EXTAIL NONE)
(QUTAIL <quantifier-tail>)
(QUTEST <q-test>)
(QUSEP NONE)
(TERM <term>)
(FACTOR <factor>)
(PRIM1 <negative>)
(EXPR <arithmetic-expression>)
(RELEXPR <non-q-relational>)
(QUREL <relational-expression>)
(AEXPR <sum>)
))(QUOTE SENAME))
(DECLIST(QUOTE(
  (UIMP (MCMAC FORM))
  (UGREATERN (MCMAC FORM))
  (ULESSP (MCMAC FORM))
  ))(QUOTE OPMAKE))

```





```

(DEFUN (QUOTE #MACRO))
(DEFUN (QUOTE #MACRO))
(ASRT ((IOACHECK GPATHS)))
(DECL ((DECSCAN IOACHECK GPATHS)))
(NONE ((GRAFTREE GPATHS)))
(DECLVAR DECL)
(DECLMODE DECL)
(ASSERT ASRT)
(DECLARE ASRT)
(CANCEL ASRT)
))(QUOTE STTL)
(DEFUN (QUOTE #MACRO))
(WHILE -1)
(END -1)
(BLKIF 1)
(BLKELSE 1)
(LOOP 1)
(BLKELSEIF 1)
(BEGIN 1)
(UNTIL -1)
(STEP -1)
(FOR 1)
))(QUOTE STLEVADJ)
(DEFUN (QUOTE #MACRO))
(*OTHER* (1200 (0 . 0)
            % (
              (1 . 0)
              (-2 . (0 . 0))
              %)))
(PLUS (100 (UNPLUS 100)))
(TIMES (120 (UNCFAC 120)
           (2 . 121)
           (-3 * (0 . 121)
                /
                (2 . 121))))
(REMAINDER (120 (1 . 120)
                MOD
                (2 . 121)))
(SETQ (5 (1 . 1000)
         :=
         (2 . 6)))

```

```

(EXPT (140 (1 . 141)
      ↑
      (2 . 141)))
(IEXPT (140 (1 . 141)
           ↑
           (2 . 141)))
(EL (1100 (1 . 1100)
      %[
      (2 . 0)
      %]))
(ELT (1100 (1 . 1100)
      %[
      (2 . 0)
      %]))
(LENGTH (1200 | (1 . 0)
         |))
(ABS (1200 | (1 . 0)
      |))
(SUBSEQ (1100 (1 . 1100)
          %[
          (2 . 0)
          TO
          (3 . 0)
          %]))
(QUOTE ((UNQT)))
(LCOND (3 (UNCOND 4)))
(EQUAL (40 (UNREL 41 =)))
(NEQUAL (40 (UNREL 41 #)))
(GEQ (40 (UNREL 41 >= 1 >)))
(LEQ (40 (UNREL 41 <= -1 <)))
(EQUALP (40 (1 . 41)
          = = (2 . 41)))
(NEQUALP (40 (1 . 41)
            # # (2 . 41)))
(FA (35 % . FA %( (1 . 0)
                  % |%
                  (2 . 0)
                  %)
    (3 . 35)))
(FU (35 % . FU %( (1 . 0)
                  % |%
                  (2 . 0)
                  %)
    (3 . 35)))
(EX (35 % . EX %( (1 . 0)
                  % |%
                  (2 . 0)
                  %)
    (3 . 35)))
(CHOICE (35 CHOICE %( (1 . 0)
                     % |%
                     (2 . 0)
                     %)
    (3 . 35)))

```

```

[BV (1 (1 . 0)
      (-2 , (0 . 0)
(EQV (15 (1 . 20)
      EQV
      (2 . 20)))
(NEQV (15 (1 . 20)
      NEQV
      (2 . 20)))
[OR (20 (1 . 20)
      (-2 ! (0 . 20)
[AND (25 (1 . 25)
      (-2 & (0 . 25)
(NOT (30 NOT (1 . 30)))
(VAR (1050 (1 . 1000)
      :
      (2 . 1000)))
(SEQMODE (1020 SEQ %( (1 . 0)
      %)))
(STRUCTMODE (1020 STRUCT %( (1 . 0)
      (-2 , (0 . 0)
      %)))
(PTRMODE (1020 PTR %( (1 . 0)
      %)))
[ATMODE (1020 (1 (1 . 0)
(CONSTSEQ (1020 CONST %( (1 . 0)
      SIZE
      (2 . 0)
      INITIAL
      (3 . 0)
      %)))
(CONSTSEQ* (1020 CONST %( (1 . 0)
      INITIAL
      (2 . 0)
      (-3 , (0 . 0)
      %)))
(CONSTRUCT (1020 CONST %( (1 . 0)
      OF
      (2 . 0)
      (-3 , (0 . 0)
      %)))
(UNDEFINED (1200 UNDEFINED %( (1 (1 . 0)
      %)))

```

((15φφ %( (1 (1 . φ) %))

(2 . φ) (-3 , (φ . φ))

(FNMODE (1020 PROC %( (1 . φ) %)) : (1 . 1φφφ))

(FUNCTION (1 ~~PROC~~ <sup>EXPR</sup> %( (1 . φ) %)) ≡ (2 . 1))

(CALL (1020 (1 . 1020) %( (2 . φ) (-3 , (φ . φ)) %)))

(RESTRICT (1 (1 . 1φφφ) REQUIRING (2 . 1φ))

(1 (1 . φ) (-2 , (φ . φ)))

```

)))(QUOTE STFMT))
(RPAQQ STPUNCL
  (% " # $ % ' % ( % ) * + , - % . / : < = > ? @ % [ \ % ] ! ~ { | } ~ > =
  < = : = ))
(RPAQQ STINITL (; IF CASE NOT))
(RPAQQ STWDL
  (! & THEN ELSE AND OR MOD SIZE INITIAL OF TO EQV NEQV))
(RPAQQ STNULLL (% % |% ))
[RPAQQ VSSPROPS ((REFOP (NOSET)
(RPAQQ VSSFNS
  (VSSSET PARCONS RELCLGRP LITQ CKLH CHOICE2FORM CALLFORM QVSAVE QUFIX
  QFVARS RSTEP CASEXP VMODEVAR VMODEFN))
(RPAQQ VSSVARS (STSPNAMES))
(APPLYMAPC VSSPROPS (FUNCTION NEWPROP))
(DEFINEQ

```

REQUIRING

```
(VSSSET
 [LAMBDA NIL
  (MAPC STSPNAMES (FUNCTION (LAMBDA (Z)
    (MAPC (CAAR Z)
      (FUNCTION (LAMBDA (X)
        (PUT X (QUOTE STSPA)
          (CDR Z]))
```

~~(PARCONS~~  
~~[LAMBDA (NAME PARS VALMODE)~~  
~~(CONS (COND~~  
~~(VALMODE (LIST (QUOTE VAR)~~  
~~NAME VALMODE))~~  
~~(T (KWOTE NAME)))~~  
~~PARS])~~

```
(RELCLGRP
 [LAMBDA (L)
  (PROG ((E (CAR L)))
    (SETQ L (MAPLIST (CDR L)
      [FUNCTION (LAMBDA (X)
        (LIST (CAR X)
          E
          (SETQ E (CADR X))
          (FUNCTION CDDR))])
    (RETURN (COND
      ((CDR L)
        (CONS (QUOTE AND)
          L))
      (T (CAR L]))
```

```
(SUBST
 (LIST E
  (SETQ E (CADR X)))
 (QUOTE *)
 (CAR X))
```

```
(LITQ
 [LAMBDA (X)
  (CONS (QUOTE LIT:)
    X)]
```

```
(CKLH
 [LAMBDA (E)
  (COND
    ([OR (VARP E)
      (AND (LISTP E)
        (REFOP (CAR E)
          E)
      (T (INPUTERROR1 (QUOTE Improper% left-hand-side]))
```

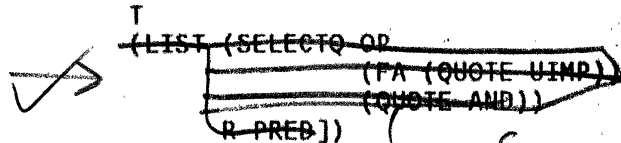
```
(CHOICE2FORM
 [LAMBDA (VAR PRED)
  (LIST (QUOTE CHOICE)
    VAR PRED (CADADR VAR])
```



```
(CALLFORM
 [LAMBDA (E)
  (PROG [(Z (OPMAKE (CAR E)
    (RETURN (COND
      ((CADDR Z)
       E)
      (Z (INPUTERROR (CAR E)
        (QUOTE
          is% not% allowed% as% a% function% name)))
      (T (LIST CONS (QUOTE CALL
        E)))]
```

```
(QVSAVE
 [LAMBDA (X)
  (PROG ((L (ASSOC (QUOTE *QV*
    STACK)))
  [COND
    ((NULL L)
     (INPUTERROR1 (QUOTE $% is% only% legal% in% quantifiers]
    (NCONC1 L X)
    (RETURN (CADR X]))
```

```
(QUFIX
 [LAMBDA (OP QV VARS TEST PRED)
  (PROG [(R (CONS (QUOTE AND)
    (CONS TEST (QFVARS VARS QV)
  [COND
    ((NULL (CDR QV))
     (INPUTERROR1 (QUOTE No% variables% for% quantifier]
    (RETURN (LIST OP (CONS (QUOTE BV)
      (CDR QV)))]
```



(SELECT OF  
 (FA <'OR <'NOT R> PRED>)  
 (<'AND R PRED>))

```
(QFVARS
 [LAMBDA (L VL)
  (MAPCONC L (FUNCTION (LAMBDA (X)
    (COND
      ((EQCAR X (QUOTE VAR))
       (NCONC1 VL X)
       NIL)
      ((EQCAR X (QUOTE AND))
       (QFVARS (CDR X)
        VL))
      (T (LIST X))
```

```
(RSTEP
 [LAMBDA (S)
  [FRPLACA (CDDR S)
  (PROG1 (CADR S)
    (FRPLACA (CDR S)
      (CADDR S]
  S])
```



(DECLARE: DONTCOPY

(FILEMAP (NIL (13006 15871 (VSSSET 13018 . 13178) (PARCONS 13182 . 13328)  
(RELCLGRP 13332 . 13634) (LITQ 13638 . 13690) (CKLH 13694 . 13846) (CHOICE2FORM  
13850 . 13937) (CALLFORM 13941 . 14193) (QVSAVE 14197 . 14422) (QUFIX 14426  
. 14797) (QFVAR 14801 . 15024) (RSTEP 15028 . 15147) (CASEXP 15151 . 15577)  
(VMODEVAR 15581 . 15668) (VMODEFN 15672 . 15868))))))

STOP

(FILECREATED " 5-MAY-74 22:05:29" VTG 5170

changes to: STEDL

previous date: " 3-MAY-74 18:54:20")

~~7~~ COMS

(DEFINEQ

ADD

(NEWBLK

[LAMBDA (P TBS)

(SETTSUCC P NIL)

(SETTLAST P P)

(SETTLNO P 0)

(SETTPGN P P)

~~(SETTBSYN P TBS)~~

(SETQ BLKNAMES (CONS P BLKNAMES))

P))

(PUTIN2

[LAMBDA (LN N P A LB EXP)

(PROG ((R (PREVST N P)))

(COND

((EQ N (TLNO R))

(SETQ A R))

(T [COND

(A (MARKEDIT A (QUOTE DELETE)))

(T (SETQ A (GENSYM

(TJOIN A (TSUCC R))

(TJOIN R A)

(SETTLNO A N)

(SETTPGN A P)))

(SETTEXT A LN)

(SETTLABEL A LB)

(SETTEXP A EXP)

(MARKEDIT A (QUOTE INSERT))

(LASTFIX A)

(RETURN A)]

(MARKEDIT

[LAMBDA (A TYP)

(PROG ((P (TPGN A))

(Z (STEDL A)))

(RETRANSL A)

(SETTRANSL P (NREML (CAR Z)

(TRANSL P)))

(COND

((SETQ Z (SELECTQ TYP

(INSERT (CADR Z))

(DELETE (CADDR Z))

(HELP)))

(APPLY\* Z A)]

(STEDL

[LAMBDA (A)

(PROG [(Z (STTL (CAR (TEXP A)

LP (COND

((NULL Z)

(SETQQ Z NONE)

(GO LP))

((LITATOM Z)

(SETQ Z (STTL Z))

(GO LP)))

(RETURN Z)]

(DELETE BLK  
[A (A) (MAPSTL (TSUCC P) NIL  
(SETQ BLKNAMES (F' ERASE))  
(REMOVE P BLKNAMES)])



```

(PRINTEXT
[LAMBDA (A FILE)
  (PROG ((L (TEXT A))
        (P (POSITION FILE))
        K)
    LP (SETQ K (FINDBREAK L (DIFFERENCE TLENGTH P)))
      (COND
        ((NTHCHAR L (ADD1 K))
         (PRIN1 (SUBSTRING L 1 K)
                FILE)
         (SETQ L (SUBSTRING L (ADD1 K)
                             -1))
         (PRINTQ ".." FILE)
         (TAB (ADD1 P)
              0 FILE)
         (GO LP))
        (T (PRIN1 L FILE]))

```

```

(FINDBREAK
[LAMBDA (L K)
  (COND
    ((LEQ (NCHARS L)
         K)
     (NCHARS L))
    (T (PROG (S (J K))
            (SETQ S (COND
                    ((STRPOS (QUOTE " " )
                             L
                             (QUOTIENT (TIMES K 2)
                                         3))
                    (QUOTE (% )))
          (T LEXBRKLIST)))
      LP (COND
        ((ZEROP J)
         (RETURN K))
        ((FMEMB (NTHCHAR L J)
                S)
         (RETURN J)))
        (SETQ J (SUB1 J))
        (GO LP))

```

```

(GETLNO
[LAMBDA (A)
  (OR (TLNO A)
      (HELP))

```

```

(LISTLCN
[LAMBDA (A FILE)
  (PROG ((N (TLNO A)))
        (AND (ZEROP (POSITION FILE))
              (SPACES (MAX 0 (DIFFERENCE 4 (NCHARS N)))
                      FILE))
        (PRIN1 N FILE))

```

```

(TABLEV
[LAMBDA (FILE)
  (TAB (IPLUS 4 (ITIMES 3 LISTLEVEL))
       2 FILE))

```

(TJOIN

```
[LAMBDA (A B)
  (SETSUCC A B)
  (COND
    (B (SETTPRED B A]))
```

(TDELT

```
[LAMBDA (A B)
  (PROG ((C A))
    LP (MARKEDIT C (QUOTE DELETE))
    ↗ (COND ((NEQ C B)
      (SETQ C (TSUCC C))
      (GO LP)))
    (TJOIN (TPRED A)
      (TSUCC B))
    (LASTFIX (TPRED A]))
```

(ORDCHK

```
[LAMBDA (A B)
  (AND (EQ (TPGN A)
    (TPGN B))
    (LEQ (TLNO A)
    (TLNO B]))
```

(PREVST

```
[LAMBDA (N P)
  (PROG ((S (TLAST P)))
    L1 (COND
      ((IGREATERP (TLNO S)
        N)
      (SETQ S (TPRED S))
      (GO L1))
      (T (RETURN S]))
```

(MAPSTL

```
[LAMBDA (MAPX MAPY MAPFN)
  (PROG (MAPZ)
    L1 (SETQ MAPZ (TSUCC MAPX))
      (APPLY* MAPFN MAPX)
      (COND
        ((AND (NEQ MAPX MAPY)
          (SETQ MAPX MAPZ))
        (GO L1))
        (T (RETURN MAPX]))
```

(TLEVADJ

```
[LAMBDA (A)
  (OR (STLEVADJ (CAR (TEXP A)))
    0))
```

)

(LISPXPRINT (QUOTE VTGFNS)

T)

(RPAQQ VTGFNS

```
(NEWBLK PUTIN2 MARKEDIT STEDL DELST1 LASTST LASTFIX LISTBLK LISTBN
LISTBH LISTSTL LISTST PRINTTEXT FINDBREAK GETLNO LISTLCN TABLEV
TJOIN TDELT ORDCHK PREVST MAPSTL TLEVADJ))
```

(LISPXPRINT (QUOTE VTGVARS)

T)

(RPAQQ VTGVARS (TLENGTH (ENDUMP VTG)))

(RPAQQ TLLENGTH 70)  
(RPAQQ VTGPROPS ((TLAST (UNDOABLE))  
 (STTL (NOSET))  
 (STLEVADJ (NOSET))  
 (TLABEL (UNDOABLE))  
 (TSUCC (UNDOABLE))  
 (TPRED (UNDOABLE))  
 (TEXT (UNDOABLE))  
 (TEXP (UNDOABLE))  
 (TPGN (UNDOABLE))  
 (TLNO (UNDOABLE))

~~(TBTYP)~~  
~~(TBTYP))~~

(APPLYMAPC VTGPROPS (FUNCTION NEWPROP))  
(PROGN (QUOTE JUSTEVALUATE)  
(FILEMAP (NIL (122 4478 (NEWBLK 134 . 310) (PUTIN2 314 . 779) (MARKEDIT 783  
 . 1086) (STEDL 1090 . 1305) (DELST1 1309 . 1364) (LASTST 1368 . 1406) (LASTFIX  
 1410 . 1499) (LISTBLK 1503 . 1728) (LISTBN 1732 . 1783) (LISTBH 1787 . 1855)  
 (LISTSTL 1859 . 2179) (LISTST 2183 . 2461) (PRINTTEXT 2465 . 2856) (FINDBREAK  
 2860 . 3267) (GETLNO 3271 . 3321) (LISTLCN 3325 . 3514) (TABLEV 3518 . 3596)  
 (TJOIN 3600 . 3680) (TDEL 3684 . 3899) (ORDCHK 3903 . 4002) (PREVST 4006  
 . 4173) (MAPSTL 4177 . 4407) (TLEVADJ 4411 . 4475))))))  
STOP



(FILECREATED " 5-AUG-74 13:18:54" VTS.;32 15901

changes to: MCMODECALL DECMODE1 MDEVAL MCMODEVAR COPYDECS DECSCAN DECST  
DECVAR

previous date: "26-JUL-74 01:32:08" VTS.;30)

(LISPXPRINT (QUOTE VTSCOMS)  
T)

(RPAQQ VTSCOMS ((PROP AEFN/FA FU EX CHOICE FUNCTION)

(PROP (OPMAKE OPMAKE)  
STMT: LIT: QUOTE VAR IVAR CVAR FNVAR)

(PROP (OPMAKE OPMAKE)  
MODEVAR MODECALL)

~~(PROP (OPMODE MCFN)  
CHOICE QUOTE NOT FAVOR FUNCTION LET DEF EQUAL NEQUAL AND OR  
LCOND DCOND TIMES QUOTIENT REMAINDER TEXT PLUS DIFFERENCE  
MINUS IVAR CVAR FNVAR)~~

(PROP (OPMODE MCFN)

LEMMA ASSERT ASSUME BLKIF BLKWHILE BLKUNTIL BLKELSEIF DECLARE  
UNTIL IF WHILE ELSEIF BEGIN LOOP BLKELSE END ELSE PROGN CANCEL  
EXIT NEXT SETQ EMPTY DECVAR DECMODE RETURN FOR STEP BLKSTEP)

(PROP (OPMODE MCFN OPMAKE)

URELT UAELT UABS)

(ENDUMP VTS)))

(DEFLIST(QUOTE(

(FA AEBIND)

(FU AEBIND)

(EX AEBIND)

(CHOICE AEBIND)

(FUNCTION AEBIND)

))(QUOTE AEFN))

(DEFLIST(QUOTE(

(STMT: AESTMT)

(LIT: I)

(QUOTE I)

(VAR AEVAR)

(IVAR AEVAR)

(CVAR AEVAR)

(FNVAR AEVAR)

))(QUOTE AEFN))

(DEFLIST(QUOTE(

(STMT: (I ARB ~~CHOICE~~))

(LIT: (I ARB))

(QUOTE (MAKECONST 1))

(VAR (MK1 FORM))

(IVAR ~~NI~~)

(CVAR ~~NI~~)

(FNVAR ~~NI~~)

))(QUOTE OPMAKE))

(DEFLIST(QUOTE(

(MODEVAR (MCMODEVAR 1))

(MODECALL (MCMODECALL 1))

))(QUOTE OPMAKE))

← OPMODE MCFN

→ (GLOBALVARS \*  
(MAPCAR BASEMODES  
(F' CADR)))

← MODEX

(LOCALFREEVARS BVMODES ROOT)

→ → VTSBLOCKS = ((CANON))

← ((LCOND) (CHOICE) (RESTRICT))

(MK1 FORM)

→ (DEFLIST(QUOTE( (MODEVAR (MCMODEVAR 1)) (MODECALL (MCMODECALL 1)) ))(QUOTE OPMAKE))

~~(DEFLIST(QUOTE(  
(MODEVAR MODE)  
(MODECALL MODE)  
) (QUOTE OPMODE))~~

(DEFLIST(QUOTE(  
~~(CHOICE (MODECHOICE 4))~~  
(QUOTE (MODEQUOTE 1))

~~(NOT BOOL)  
(FA BOOL) → (FU BOOLE)  
(EX BOOL)~~

~~(FUNCTION (MODEFUNCTION 2))~~ → to VPROC

~~(LEQ BOOL)  
(GEO BOOL)  
\* (EQUAL BOOL)  
(NEQUAL BOOL)~~

~~(AND BOOL)  
\* (OR BOOL)~~ → \*'ed to VBOOL/VQC

~~(LCOND (MODECOND ARB))  
(DCOND (MODECOND ARB))~~

→ \*'ed to VIA/VREL

~~(TIMES INT)  
(QUOTIENT INT)  
(REMAINDER INT)  
(TEXT INT)~~

\* ~~(PLUS INT)  
(DIFFERENCE INT)  
(MINUS INT)~~

(VAR MODE)  
(IVAR (MODEXVAR 2))  
(CVAR (MODEXVAR 2))  
(FNVAR MODE)

))(QUOTE OPMODE))

(DEFLIST(QUOTE(  
~~(CHOICE OMCHOICE)~~

~~(QUOTE NIL)~~ → to VPROC

~~(NOT (BOOL))  
(FA OMQUANT)  
(EX OMQUANT)  
(FUNCTION NIL) → (FU OMQUANT)~~

\* ~~(LEQ (INT INT))  
(GEO (INT INT))  
(EQUAL (INT INT))  
(NEQUAL (INT INT))~~

~~(AND (BOOL BOOL))  
\* (OR (BOOL BOOL))  
(LCOND OMCOND)  
(DCOND OMCOND)~~

~~(TIMES (INT INT))  
(QUOTIENT (INT INT))  
(REMAINDER (INT INT))~~

\* ~~(TEXT (INT INT))  
(PLUS (INT INT))  
(DIFFERENCE (INT INT))  
(MINUS (INT))~~

```
(VAR OMVAR)
(IVAR NIL)
(CVAR NIL)
(FNVAR NIL)
))(QUOTE MCFN))
(DEFLIST(QUOTE(
  (LEMMA STMT:)
  (ASSERT STMT:)
  (ASSUME STMT:)
  (BLKIF STMT:)
  (BLKWHILE STMT:)
  (BLKUNTIL STMT:)
  (BLKELSEIF STMT:)
  (DECLARE STMT:)
  (UNTIL STMT:)
  (IF STMT:)
  (WHILE STMT:)
  (ELSEIF STMT:)
  (BEGIN STMT:)
  (LOOP STMT:)
  (BLKELSE STMT:)
  (END STMT:)
  (ELSE STMT:)
  (PROGN STMT:)
  (CANCEL STMT:)
  (EXIT STMT:)
  (NEXT STMT:)
  (SETQ STMT:)
  (EMPTY STMT:)
  (DECVAR STMT:)
  (DECMODE STMT:)
  (RETURN STMT:)
  (FOR STMT:)
  (STEP STMT:)
  (BLKSTEP STMT:)
))(QUOTE OPMODE))
(DEFLIST(QUOTE(
  (LEMMA (BOOL))
  (ASSERT (BOOL))
  (ASSUME (BOOL))
  (BLKIF (BOOL))
  (BLKWHILE (BOOL))
  (BLKUNTIL (BOOL))
  (BLKELSEIF (BOOL))
  (DECLARE (BOOL))
  (UNTIL (BOOL . STMT:))
  (IF (BOOL . STMT:))
  (WHILE (BOOL . STMT:))
  (ELSEIF (BOOL . STMT:))
  (BEGIN NIL)
  (LOOP NIL)
  (BLKELSE NIL)
  (END NIL)
```

```

(ELSE (STMT: . STMT:))
(PROGN (STMT: . STMT:))
(CANCEL NIL)
(EXIT NIL)
(NEXT NIL)
(SETQ OMSETQ)
(EMPTY NIL)
(DECVAR NIL)
(DECMODE NIL)
(RETURN OMRETURN)
(FOR (INT INT))
(STEP (INT INT . STMT:))
(BLKSTEP (INT INT))
))(QUOTE MCFN))
(DEFLIST(QUOTE(
  (URELT NIL)
  (UAELT NIL)
  (UABS NIL)
  ))(QUOTE OPMODE))
(DEFLIST(QUOTE(
  (URELT OMRELT)
  (UAELT OMAELT)
  (UABS OMABS)
  ))(QUOTE MCFN))
(DEFLIST(QUOTE(
  (URELT (MCRELT 2))
  (UAELT (MCAELT 2))
  (UABS (MCABS 1))
  ))(QUOTE OPMAKE))
(RPAQQ VTSPPRS ((SFDEF (NOSET))
  (MCFN (NOSET MACRO))
  (AEFN (NOSET MACRO))
  (VMODE (CLR MAP)
    CTX)
  (VRO (CLR)
    CTX)
  ((FNDEF (CLR MAP)
    CTX))
  (MCMACRO (NOSET))
  (TRANSL (UNDOABLE ADD MEMB DEL REM))
  (TBTYP)
  (PENV)
  (TLS (REM))
  (TLP (REM))
  (TUP)
  (SEXP)))
(RPAQQ VTSFNS
(DOTRANS SCANERROR SCANERROR1 GRAFTREE GRAFT1 COPYDECS DECSCAN DECST
DECVAR DECFN SFSET SFDEFS ATMODE ECONSTP TCANON CANON CANON1
AESTMT AEVAR AEBIND MDEVAL MCMODEVAR MCMODECALL OMAELT MCAELT
OMRELT MCRELT DEREV OMABS MCABS OMQUANT OMCHOICE MODECHOICE
MODEQUOTE MODEFUNCTION OMCOND MODECOND OMVAR MODEXVAR OMSETQ
ELVAR LOOPSET OMRETURN CURPROC MCMAC SOMEINDEX))
(RPAQQ VTSVARS (BASEMODES))
(APPLYMAPC VTSPPRS (FUNCTION NEWPROP))
(DEFINEQ

```

→ (MODEX MODE)

→ (MODEX (MODE))

→ (MODEX (MCMODEX 1))

← REM

← REM

→ ~~(FNDEF (CLR MAP) CTX)~~

(DOTRANS  
[LAMBDA (A FN ALL)

(\* A must be a SPECVAR  
(used free))

(COND  
((NOT (MEMBTRANSL FN A))  
(APPLY\* FN A)  
(ADDTRANSL A FN))  
(ALL (APPLY\* FN A]))

(SCANERROR  
[LAMBDA (MESS1 MESS2 A)

(\* Does a stack search for  
DOTRANS)

(COND  
([OR A (AND (SETQ A (STKPOS (QUOTE DOTRANS)))  
(SETQ A (EVALV (QUOTE A)  
A])  
(LISPXPRI1 Line% T)  
(LISPXPRI2 (OR (TLNO A)  
A)  
T)  
(LISPXPRI3 :% T)))  
(ERRORMESS (LIST 17 (CONS MESS1 MESS2)))  
(ERROR!]))

(SCANERROR1  
[LAMBDA (MESS1 A)  
(SCANERROR MESS1 (QUOTE "")  
A])

(GRAFTREE  
[LAMBDA (P)  
(COND  
((SETQ P (GRAFT1 P NIL))  
(SCANERROR1 (QUOTE Unpaired% END)  
P]))

(GRAFT1  
[LAMBDA (A FROM)  
(PROG (B C)  
L1 (REMTLP A)  
L2 (SETTUP A FROM)  
(REMTLS A)  
(SELECTQ (TLEVADJ A)  
(-1 (RETURN A))  
[1 (SETQ B (GRAFT1 (TSUCC A)  
A))  
(COND  
((SETQ C (TSUCC B))  
(SETTLS A C)  
(SETTLP C A)  
(SETQ A C)  
(GO L2))  
(T (SETQ A B)  
NIL)  
(COND  
((SETQ A (TSUCC A))  
(GO L1))  
(NULL FROM)  
(RETURN))  
(T (SCANERROR1 (QUOTE No% matching% END) FROM]))

(COPYDECS

[LAMBDA (C1 C2)

(MAPVMODE [FUNCTION (LAMBDA (X Y)

(COND

((NOT (VMODE X C2))

(SETVMODE X Y C2)

(SETVRO X T)

(SETCVAL X (CVAL X C1)  
C2))

((NEQ (VMODE X C2)  
Y)

(SCANERROR (QUOTE Attempt% to% redeclare)  
X])

C1)

~~(MAPFNDEF [FUNCTION (LAMBDA (X Y)~~

~~(COND~~

~~((NOT (FNDEF X C2))~~

~~(SETFNDEF X Y C2))~~

~~((NEQ (FNDEF X C2)  
Y)~~

~~(SCANERROR (QUOTE Attempt% to% redefine)  
Y])~~

C1)

(MAPTHCLAUSEP [FUNCTION [LAMBDA (X)  
(PUTCLAUSE X C2))

C1])

(DECSCAN

[LAMBDA (P)

(SETCTX (ROOTC P))

(SETPATHL P NIL)

(CLRVMODE)

~~(CLRFNDEF)~~ ← (CLRCVAL)

(CLRvro)

(CLRPROP (QUOTE THCLAUSEP))

[MAPC (PENV P)

(FUNCTION (LAMBDA (X)

(COND

~~((NEQ (TBTYPE X)~~

~~(QUOTE DATA))~~

~~(SCANERROR X (QUOTE "is not a DECLARATIONS module")  
P))~~

~~(T (COPYDECS (ROOTC X)~~

~~NIL])~~

← DECLARATIONS

← (CURCTX)

(MAPSTL P NIL (FUNCTION (LAMBDA (X)

(DOTRANS X (QUOTE DECST)  
T])

```

(DECST
 [LAMBDA (A)
  (PROG ((E (TEXP A))
        N D)
    (SELECTQ (CAR E)
      [ASSERT (COND
        [(EQ (TBTYP (TPGN A))
          (QUOTE DECLARATIONS))
        ]
        [COND
          [ECONSTP (SETQ E (CANON (CADR E))
            (PUTCLAUSE E))
          ]
          (T (HELP))
        ]
        (T (DELTRANSL A (QUOTE TCANON))
          (DECVAR (MAPC (CDR E)
            (FUNCTION DECVAR)))
          [START [COND
            [(CADDR (CADR E))
            ]
            ((CADDR (CADR E)) (* Procedure returns a value)
            (DECVAR (CADR E))
            (MAPC (CDDR E)
              (FUNCTION (LAMBDA (X)
                (DECVAR X)
                (SETVRO (CADR X)
                  T)
              )
            )
            (DELTRANSL A (QUOTE TCANON))
          ]
        ]
      ]
    )
  )

```

```

(DECVAR (VAR)
 [LAMBDA (VAR)
  (PROG ([N (CADR (SETQ VAR (CANON VAR)
    (E (CADDR VAR)))
    (COND
      [(EQ (CAR VAR)
        (QUOTE FNVAR))
      ]
      (DECFN (CONS (MK1* (QUOTE VAR)
        (CADR VAR)
        (MD E))
        (CDADR E))
        (CADDR E))
      ]
      ((VMODE N)
        (SCANERROR N (QUOTE "is already declared")))
      (T (SELECTQ (CAR VAR)
        (VAR (SETVMODE N E))
        (IVAR (SETVMODE N (MD E)))
        [EVAR (COND
          [(ECONSTP E)
          ]
          (SETVMODE N (MD E))
          (SETCVAR N E)
          (SETVRO N T))
          (T (SCANERROR N (QUOTE
            "- value is not constant")
          )
        ]
      ]
    )
  )

```

(HELP)]

```

(DEC FN
 [LAMBDA (VARS DEF)
 (PROG ((N (CADR VARS)))
 (COND
 ((OR (FNDEF N)
 (OPMAKE N))
 (SCANERROR (QUOTE Attempt% to% redefine)
 N))
 (T (SETFNDEF N (LIST VARS DEF T))

```

```

(SFSET
 [LAMBDA NIL
 (SETCTX (ROOTCTX))
 (CLRVMODE)
 (CLRCVAL)
 [MAPC BASEMODES (FUNCTION (LAMBDA (X)
 (PROG [(M (SET (CADR X)
 (ATMODE (CAR X)
 (COND
 ((CDDR X)
 (SETCVAL (CAR X)
 M))
 (SETVMODE (CAR X)
 MODEMODE]

```

(SETVRO (CAR X) T)

(DECVAR [LIST 'FNVAR FN  
(CANON (CONS 'FUNCTION  
(CONS (CONS 'STRCMODE (CAR X))  
(CDR X) T]

```

(SETVMODE T BOOLMODE)
(SETVMODE NIL BOOLMODE)
(CLRFNDEF)
(MAPCR (SFDEFS)
 (FUNCTION (LAMBDA (FN)
 (PROG ((X (SFDEF FN))

```

```

(SETQ ARGS (MAPCAR (CAR X)
 (FUNCTION MDEVAL)))
[SETQ E (CANON (CADR X)
 (SETQ B (MAPCAR ARGS
 (FUNCTION CDR]
 (SETFNDEF FN (NLIST (CONS (MK1* (QUOTE VAR)
 FN
 ([LAMBDA (BVMODES)
 (MD E)
 B))
 ARGS)
 E
 (CADDR X)]

```

```

(SFDEFS
 [LAMBDA NIL
 (SETQ SFDEFS (MAPCONC FILELST (FUNCTION (LAMBDA (X)
 (FPROPSAVES X (QUOTE SFDEF))

```

```

(ATMODE
 [LAMBDA (FN ARGS)
 (MK1* (QUOTE ATMODE)
 (CONS (MAKECONST FN)
 (MAKECONST X)
 ARGS])

```





```

(T
  (MAPC
    L
    (FUNCTION (LAMBDA (X)
      (COND
        ([AND Z (NEQ (MD X)
          (CVAL (COND
            [(LISTP Z)
              (PROG1 (CAR Z)
                (SETQ Z (CDR Z))
              (T Z)
            (SCANERROR (CAR E)
              (QUOTE "- mode error" ]
          (RETURN L ]))

```

```

(AESTMT
  [LAMBDA (L FN)
    (CONS (CAR L)
      (MAPCAR (CDR L)
        FN ]))

```

```

(AEVAR
  [LAMBDA (L FN)
    (LIST (CAR L)
      (APPLY* FN (CADR L))

```

```

(AEBIND
  [LAMBDA (L FN)
    (PROG [(BVMODES BVMODES)
      (B (APPLY* FN (CAR L)
        (SETQ BVMODES (ADDBV B BVMODES))
        (RETURN (CONS B (MAPCARN (CDR L)
          FN ]))

```

→

```

(MDEVAL
  [LAMBDA (E)
    (PROG ((E1 (CANON E)))
      (RETURN (COND
        ((EQ (MD E1)
          MODEMODE)
          E1)
        (T (SCANERROR1 (QUOTE "Improper mode" ]))

```

→

```

(MCMODEVAR
  [LAMBDA (X)
    ([LAMBDA (Y)
      (OR (COND
        (Y (AND (EQ (CADR Y)
          MODEMODE)
          X))
        ((EQ (VMODE X)
          MODEMODE)
          (CVAL X)))
        (SCANERROR (QUOTE Undefined% mode)
          X]
      (FASSOC X BVMODES ]))

```

```

(MCMODECALL
[LAMBDA (E)
  (PROG [(DEF (FNDEF (CAR E)
    (RETURN (COND
      ((NULL DEF)
        (SCANERROR (QUOTE Undefined% function)
          (CAR E)))
      ((NEQ (CADDR (CAR DEF))
        MODEMODE)
        (SCANERROR (CAR E)
          (QUOTE "is not a mode function"))))
    (T (FSUBMAKE (PARAMATCH (CAR E)
      (CDAR DEF)
      (CDR E))
      (CADR DEF])

```

```

(OMAELT
[LAMBDA (L)
  (AND (EQ (MD (CADR L))
    INTMODE)
    (EQCAR (MD (CAR L))
      (QUOTE SEQMODE)])

```

```

(MCAELT
[LAMBDA (X J)
  (MK* (QUOTE EL)
    (DEREF X)
    J)]

```

```

(OMRELT
[LAMBDA (L)
  (PROG [(M (MD (CAR L)))
    (S (CADR (CADR L))
      (RETURN (AND (EQ (CAR M)
        (QUOTE STRUCMODE))
        (SOME (CDR M)
          (FUNCTION (LAMBDA (X)
            (EQ (CADR X)
              S]))

```

```

(MCRELT
[LAMBDA (X S)
  (PROG [(M (MD (SETQ X (DEREF X))
    (S (CADR S)))
    (MK* (QUOTE EL)
      X
      (SOMEINDEX (CDR M)
        (FUNCTION (LAMBDA (Y)
          (EQ (CADR Y)
            S]))

```

```
(DEREF
 [LAMBDA (E)
  (COND
   ((EQCAR (MD E)
    (QUOTE PTRMODE))
    (MK* (QUOTE VAL)
     E))
   (T E]))
```

```
(OMABS
 [LAMBDA (L)
  ([LAMBDA (M)
   (SELECTQ (CAR M)
    ((SEQMODE STRUCMODE)
     T)
    (ATMODE (EQ M INTMODE))
    NIL])
  (MD (CAR L]))
```

```
(MCABS
 [LAMBDA (X)
  (MK* (COND
   ((EQ (MD X)
    INTMODE)
    (QUOTE ABS))
   (T (QUOTE LENGTH)))
  X])
```

```
(OMQUANT
 [LAMBDA (L)
  (PROG ((BVMODES (ADDBV (CAR L)
   BVMODES)))
   (RETURN (AND (EQ (MD (CADR L))
    BOOLMODE)
    (EQ (MD (CADDR L))
    BOOLMODE))
```

```
(OMCHOICE
 [LAMBDA (L)
  (PROG ((BVMODES (ADDBV (CAR L)
   BVMODES)))
   (RETURN (EQ (MD (CADR L))
    BOOLMODE]))
```

```
(MODECHOICE
 [LAMBDA (VL B E)
  (PROG ((BVMODES (ADDBV VL BVMODES)))
   (RETURN (MD E]))
```

```
(MODEQUOTE
 [LAMBDA (X)
  (COND
   ((LITATOM X)
    CHARMODE)
   (T (MD X]))
```

→ to vec

```

(MODEFUNCTION
 [LAMBDA (V E)
  ([LAMBDA (BVMODES) (MKI 'IFNMODE
   (CONS * (MAPCAR (CADR V)
    (MD E) (ADDBV V BVMODES)))
   (F' CADR))))])

```

to VPROC

```

(OMCOND
 [LAMBDA (L)
  (PROG [(M (MD (CADR L)
   (RETURN (EVERY L [FUNCTION (LAMBDA (X Z)
    (AND (EQ (MD X)
     BOOLMODE)
     (EQ (MD (CADR Z))
      M])
    (FUNCTION CDDR))])

```

to VOC

```

(MODECOND
 [LAMBDA (L)
  (MD (CADR L])

```

```

(OMVAR
 [LAMBDA (L)
  (EQ (MD (CADR L))
   MODEMODE])

```

```

(MODEXVAR
 [LAMBDA (V E)
  (MD E])

```

```

(OMSETQ
 [LAMBDA (L)
  (PROG [(V (ELVAR (CAR L)
   [COND
    ((OR (VRO V)
     (LOOPSET V A))
     (SCANERROR V (QUOTE "is read-only")
      (RETURN (EQ (MD (CAR L))
       (MD (CADR L))

```

(\* Uses A from DOTRANS)

```

(ELVAR
 [LAMBDA (E)
  (COND
   ((VARP E)
    E)
   ((EQCAR E (QUOTE EL))
    (ELVAR (CADR E])

```

```

(LOOPSET
 [LAMBDA (V A)
  (PROG ((B A)
   E)
   LP (RETURN (COND
    ((SETQ B (TUP B))
     (OR (AND (EQ (CAR (SETQ E (TEXP B)))
      (QUOTE FOR))
      (EQ (CADR E)
       V))
     (GO LP])

```

```

(OMRETURN
 [LAMBDA (L)
  (OR (NULL L)
   (PROG [(M (CADDR (CADR (TEXP (CURPROC)
    (RETURN (COND
      ((NULL M)
       (SCANERROR1 (QUOTE
         "Procedure does not return a value"))
        (T (EQ (MD (CAR L)
          M))

```

```

(CURPROC
 [LAMBDA NIL
  (OR [CAR (SOME BLKNAME (FUNCTION (LAMBDA (X)
    (FTAILP (ROOTC X)
    (CURCTX)
    (HELP])

```

```

(MCMAC
 [LAMBDA (OP L)
  (PROG ((D (MCMACRO OP)))
   (RETURN (FSUBMAKE (MAP2CAR (CAR D)
    L
    (FUNCTION CONS))
    (CADR D])

```

```

(SOMEINDEX
 [LAMBDA (SOMEXX SOMEXFN)
  (PROG ((SOMEXN 0))
   (RETURN (AND [SOME SOMEXX (FUNCTION (LAMBDA (SOMEXY)
    (SETQ SOMEXN (ADD1 SOMEXN))
    (APPLY* SOMEXFN SOMEXY)
    SOMEXN])

```

```

)
(DEFLIST(QUOTE(
 (COPYDECS (5-AUG-74 . 1229))
 (DECSCAN (5-AUG-74 . 1230))
 (DECST (5-AUG-74 . 1234))
 (DECVAR (5-AUG-74 . 1233))
 (SFSET (18-JUN-74 . 1443))
 (SFDEFS (18-JUN-74 . 1440))
 (ECONSTP (4-JUN-74 . 43))
 (MDEVAL (5-AUG-74 . 119))
 (MCMODEVAR (5-AUG-74 . 120))
 (MCMODECALL (5-AUG-74 . 25))
 (OMABS (8-JUN-74 . 1844))
 (MCABS (8-JUN-74 . 1844))
 ))(QUOTE EDITDATE))
 (RPAQQ BASEMODES ((MODE MODEMODE . T)
  (INT INTMODE . T)
  (BOOL BOOLMODE . T)
  (REAL REALMODE . T)
  (CHAR CHARMODE . T)
  (STMT: STMTMODE)
  (NULL: NULLMODE)
  (ANY: ANYMODE)))

```

```

(MCMODEX
 [X (X)
 (COND
 ((AND (LETATOM X)
  (VRO X)
  (CVAL X)))
 (T X])

```

(DECLARE: DONTCOPY

(FILEMAP (NIL (4694 15278 (DOTRANS 4706 . 4955) (SCANERROR 4959 . 5359) (SCANERROR1 5363 . 5441) (GRAFTREE 5445 . 5559) (GRAFT1 5563 . 6071) (COPYDECS 6075 . 6673) (DECSCAN 6677 . 7116) (DECST 7120 . 7721) (DECVAR 7725 . 8313) (DECFN 8317 . 8527) (SFSET 8531 . 9314) (SFDEFS 9318 . 9436) (ATMODE 9440 . 9533) (ECONSTP 9537 . 9798) (TCANON 9802 . 9906) (CANON 9910 . 9956) (CANON1 9960 . 10784) (AESTMT 10788 . 10862) (AEVAR 10866 . 10934) (AEBIND 10938 . 11121) (MDEVAL 11125 . 11295) (MCMODEVAR 11299 . 11546) (MCMODECALL 11550 . 11930) (OMAELT 11934 . 12043) (MCAELT 12047 . 12112) (OMRELT 12116 . 12340) (MCRELT 12344 . 12552) (DEREF 12556 . 12679) (OMABS 12683 . 12832) (MCABS 12836 . 12954) (OMQUANT 12958 . 13141) (OMCHOICE 13145 . 13280) (MODECHOICE 13284 . 13384) (MODEQUOTE 13388 . 13474) (MODEFUNCTION 13478 . 13567) (OMCOND 13571 . 13780) (MODECOND 13784 . 13827) (OMVAR 13831 . 13888) (MODEXVAR 13892 . 13931) (OMSETQ 13935 . 14203) (ELVAR 14207 . 14309) (LOOPSET 14313 . 14532) (OMRETURN 14536 . 14765) (CURPROC 14769 . 14901) (MCMAC 14905 . 15062) (SOMEINDEX 15066 . 15275))))

STOP

(FILECREATED " 5-AUG-74 13:05:37" VX.;17 14135

changes to: VXSET AUXDESC EXECLP GETPROMPT TRIM GET2A GETBLKA INBLK  
TEXTADDR KWSET VXVARS

previous date: " 6-JUN-74 00:22:17" VX.;16)

*→ Editfix*

(LISPXPRINT (QUOTE VXCOMS)

T)

(RPAQQ VXCOMS ((PROP (PARSEDEF SENAME)  
XCOMQ TADDR SADDR LADDR T1ADDR T2ADDR BADDR OBADDR QID  
FILENAME TENAME)

~~(ADDVARS (TRACEGROUPS)  
(GLOBALVARS TRACELIST TRACEGROUPS))~~  
(ENDUMP VX)))

*VXGLOBALS =*

(DEFLIST(QUOTE(  
[XCOMQ ((EITHER (OK (: (RTFRM EXECLP T)))  
(STOP (: (RTFRM EXECLP NIL)))  
(? (: (PRINT KWLIST)

*→ (TRACELIST  
TRACEGROUPS)*

[TADDR ((EITHER (: (BADDR)  
:  
(LADDR)  
(# RPLBA 2))  
((LADDR]  
(SADDR ((LADDR)))  
[LADDR ((EITHER (\$NUM (: (GETBLKA LASTBLK)  
(: TEXTADDR 2))  
(\$QUOTE (EITHER (\$NUM  
((: NIL)))  
(: (GETBLKA LASTBLK))  
(: TXAFIND 3))  
(\$ (: (GETBLKA LASTBLK))  
(: TXALAST 1]

(T1ADDR ((TADDR)  
(: GET1A 1)))  
[T2ADDR ((TADDR)  
(EITHER (- (TADDR)  
(: GET2A 2))  
((: GET2A1 1]

(BADDR ((QID)  
(: GETBLKA 1)))  
[OBADDR ((EITHER ((BADDR))  
((: (GETBLKA LASTBLK]  
(QID (\$ID (: QUOTE 1)))  
[FILENAME ((EITHER (\$QUOTE (# MKATOM 1))  
((\$ 0)  
(OPTION < (: <)  
(TENAME)  
>  
(: >))  
(TENAME)  
(OPTION % (: %.)  
(TENAME))  
(REPEAT ; (: ;)  
(TENAME))  
(; \$)  
(# PACK 1]



```

[TENAME ((EITHER ($ID)
              ($NUM))
  (REPEAT - (: -)
            (EITHER ($ID)
                    ($NUM)]
)))(QUOTE PARSEDEF))
(DEFLIST(QUOTE(
  (XCOMQ <basic-exec-command>)
  (TADDR <text-address>)
  (SADDR NONE)
  (LADDR <line-address>)
  (T1ADDR NONE)
  (T2ADDR <text-range>)
  (BADDR <module-name>)
  (OBADDR NONE)
  (QID NONE)
  (FILENAME <filename>)
  (TENAME <TENEX-name-field>)
)))(QUOTE SENAME))
  (ADDTOVAR TRACEGROUPS)
  (ADDTOVAR GLOBALVARS TRACELIST TRACEGROUPS)
  (RPAQQ VXFNS
    (VXSET EXECLP GETPROMPT AUXDESC EXEC1 TRIM READFIRST FORLISPX? VX
      PRINTVXHIST1 VXUNDOPRINT FIXWORD INPUTERROR INPUTERROR1 GET2A
      GET1A GET2A1 GETB2A GETBLKA INBLK GETPATH GETNCLAUSE RPLBA
      TEXTADDR TXAFIND TXALAST KWSET PKWDS PKWDS1 PTRACE EXPANDGROUPS
      PTRACE1 PTRACE2))
  (RPAQQ VXVARS ((TRACELIST NIL)
    (EXECTIME NIL)
    (LASTBLK NIL)))
  (RPAQQ VXADVICE ([PRINTHISTORY] BEFORE NIL
    (COND ((PRINTVXHIST2 ED X VAL NOVALUES <INPUT
      VALUEFORM OTHER)
      (RETURN]
    (UNDOPRINT BEFORE NIL (COND ((VXUNDOPRINT X)
      (RETURN (CAR X]
  (DEFINEQ
    (VXSET
      [LAMBDA NIL
        (RELINK (QUOTE PRINTHISTORY))
        (RELINK (QUOTE UNDOLISPBLOCK]))

```

(EXECLP

[LAMBDA (COMDESC XFILE AUXDESC)

(PROG [(FILE (OR XFILE (INPUT)))

(XPROMPT (GETPROMPT (CAR COMDESC)))

KWRECLIST KWLIST BOTHCOMLIST (NOPARSEFN (CAR (CDDDDR COMDESC]

(AUXDESC AUXDESC)

→  
→

L1 (ERSETQ (PROG (~~OLDLINE ECHO~~ (LINE ""))

LOOP (SETQ ECHO (AND (EQ FILE T)

(NEQ (SETQ ECHO (OUTPUT))

T)

ECHO))

(SETQ H (READFIRST FILE))

(COND

→

(EXECTIME (ETIME (EXEC H FILE ECHO)))

(T (EXEC1 ~~FILE ECHO~~))

(GO LOOP)))

(COND

((NEQ FILE T)

(RETURN NIL)))

(TERPRI T)

(GO L1))

(GETPROMPT

[LAMBDA (CHAR)

(PROG [(P (LIST CHAR))

(POS (STKPOS (QUOTE EXECLP]

LP (COND

((SETQ POS (STKPOS (QUOTE EXECLP)

1

(STKNTH -1 POS)))

[COND

((EQ (CAR (EVALV (QUOTE COMDESC)

POS))

CHAR)

(SETQ P (CONS (CAR P)

P]

(GO LP)))

(RETURN (PACK P])

(AUXDESC

[LAMBDA (DESC) (COND ((OR (NEQ DESC AUXDESC) FLAG)

(SETQ KWRECLIST (APPEND (CAR DESC)

(CADR COMDESC)))

(SETQ KWLIST (APPEND (CADR DESC)

(CADDR COMDESC)))

(SETQ BOTHCOMLIST (APPEND (CADR DESC)

(CADDRR COMDESC))

(NIL)) (SETQ AUXDESC DESC]

```

(EXEC1
[LAMBDA (FILE FILE)
(COND
  ((EQ H (QUOTE E))
   (LISPX (LISPXREAD FILE)
           XPROMPT))
  ((FORLISPX? H FILE)
   (LISPX H XPROMPT))
  (T (SETQ LINE (TRIM (EDITLINE (SETQ OLDLINE LINE)
                                FILE)))
      (COND
        ((EQ FILE T)
         (COND
          (ECHO (TERPRI ECHO)
                (PRIN1 XPROMPT ECHO)
                (PRIN2 H ECHO)
                (COND
                 ((NEQ (NCHARS LINE)
                      0)
                  (SPACES 1 ECHO)
                  (PRIN1 LINE ECHO)))
                (TERPRI ECHO)))
          (LISPXUNREAD (LIST (LIST H LINE)))
          (LISPX (QUOTE VX)
                 XPROMPT))
          (T (VX H LINE T]))
         (NEG (OUTPUT)
              T))
        ))))

```

*(\* Uses OLDLINE and LINE)*

*(PROG ((H (READFIRST FILE)))*

```

(TRIM
 [LAMBDA (S)
 (PROG ((M 1)
        (N -1))
  LQ (SELECTQ (NTHCHAR S M)
             (ADDIVAR M)
             (GO LQ))
      (NIL (RETURN (QUOTE "")))
      NIL)
  LP (RETURN (COND
             ((EQ (NTHCHAR S N)
                  (QUOTE %))
              (SUBIVAR N)
              (GO LP))
             ((AND (EQ M 1)
                    (EQ N -1))
              S)
             (T (SUBSTRING S M N]))

```

*(%A %t)*

```

(READFIRST
 [LAMBDA (FILE)
 (COND
  ((EQ FILE T)
   (PROMPTCHAR XPROMPT T LISPXHISTORY)
   (LISPXREAD T))
  (T (READ FILE])

```

*to VEP*

```

(FORLISPX?
 [LAMBDA (X FILE)
  (OR (LISTP X)
    (PROG [(MORE (OR (NEQ FILE T)
                    (LISPXREADP)
    (RETURN (COND
              [(LITATOM X)
               (OR [COND
                    (MORE (AND (GETD X)
                              (SELECTQ (PEEKC FILE)
                                        ((%[ %] %))
                                        T)
                                        NIL)))
              (T (NEQ (CAR X)
                    (QUOTE NOBIND]
    (NOT (FMEMB (FIXWORD X BOTHCOMLIST T)
              KWLIST]
    (T (NOT MORE])

```

```

(VX
 [LAMBDA (HH LINE FLAG)
  (PROG ((LEX (LEXSTAT HH))
        PS)
    (FRPLACD (NLEFT LEX 2)
             (LEXSTAT LINE T))
    [COND
      ((SETQ PS (SELECTQ (CAR LEX)
                        ($ID (OR (FASSOC (FIXWORD (CDR LEX)
                                         KWLIST)
                                     KWRECLIST)
                                (FASSOC (CAR LEX)
                                         KWRECLIST))))
            (FASSOC (CAR LEX)
                    KWRECLIST)))
      (SETQ PS (PARSE LEX (CDR PS)
    [COND
      ([OR PS (AND NOPARSEFN (SETQ PS (APPLY* NOPARSEFN LEX]
        (EVAL PS))
      (T (INPUTERROR1 (QUOTE Error]
    (OR FLAG (RTERM LISPX T])

```

(\* NOTE: PUTIN1 USES LINE FREE,  
INSBLKX USES HH AND LINE)

(PRINTVXHIST)

[LAMBDA (~~ID X VAL NOVALUES VALUEFORM OTHER~~) (X)

(PROG (Y)

(COND

((EQ (CAR X)  
(QUOTE VX))

(RETURN NIL)))

(~~TAB 5 NIL T~~) (PROG ((Y (CAADR X)))

(AND ID (PRIN2 ID T))

(PRIN1 (CAADR X)

T) ← T

(SETQ Y (CADADR X))

(COND

((NEQUAL Y (QUOTE ""))

(SPACES 1 T)

(PRIN1 Y T))) ← T

(TERPRI T)

[COND  
(OTHER (MAPC OTHER (FUNCTION (LAMBDA (Z)

(COND

((NLISTP Z)

(PRIN1 Z T))

(T (APPLY (CAR Z)

(CDR Z)

(RETURN T])

(VXUNDOPRINT

[LAMBDA (X)

(COND

((EQ (CAR (LISTP X))

(QUOTE VX))

(OR (LITATOM (PRIN1 (CAADR X)))

(PRIN1 (QUOTE " --"))])

(FIXWORD

[LAMBDA (W SPLST Q)

(PROG [(WORD (COND

((LISTP W)

(CAR W))

(T W)

(RETURN (COND

((FMEMB WORD SPLST)

WORD)

([SETQ WORD (COND

(Q (CHOOZ WORD 70 SPLST))

(T (FIXSPELL WORD 70 SPLST])

(COND

((LISTP W)

(RPLACA W WORD)))

WORD])

(INPUTERROR

[LAMBDA (MESS1 MESS2)

(\* Uses FILE from EXECLP and HH  
from VX)

(ERRORMESS (LIST 17 (CONS MESS1 MESS2)))

(COND

((AND (NEQ FILE T)  
(NUMBERP HH))

(PRINQ "in line " T)

(PRINT HH T)))

(ERROR!])

(INPUTERROR1

[LAMBDA (MESS1)

(INPUTERROR MESS1 (QUOTE ""))

(GET2A

[LAMBDA (A B NOORD)

(COND

((NULL A)

(INPUTERROR1 (QUOTE First% line% address?)))

((NULL B)

(INPUTERROR1 (QUOTE Second% line% address?)))

((OR (ZEROP (TLNO A))

(ZEROP (TLNO B)))

(INPUTERROR1 (QUOTE Range% must% not% include% header)))

((OR NOORD (ORDCHK A B))

(CONS A B))

(T (INPUTERROR1 (QUOTE Line% addresses% not% in% order]))

(GET1A

[LAMBDA (A)

(OR A (INPUTERROR1 (QUOTE Bad% line% address]))

(GET2A1

[LAMBDA (A)

(CONS (SETQ A (GET1A A))

A))

(GETB2A

[LAMBDA (P)

(PROG [(S (TSUCC (GETBLKA P)

(COND

[S (RETURN (CONS S (LASTST S)

(T (INPUTERROR P (QUOTE "is empty"))]

(GETBLKA

[LAMBDA (P)

(PROG ((P1 (FIXWORD P BLKNAMES)))

(RETURN (COND

(P1 (INBLK P1))

(T (INPUTERROR (QUOTE No% module)

P]))

```
(INBLK
 [LAMBDA (P)
  (SETQ LASTBLK P)])
```

(\* Gets advised in VXL)

```
(GETPATH
 [LAMBDA (P J)
  (PROG ((L (PATHL P)))
    (COND
      ((OR (ILESSP J 1)
           (IGREATERP J (FLENGTH L)))
        (INPUTERROR (QUOTE No% path)
                     J))
      (T (RETURN (CAR (FNTH L J)))))
```

```
(GETNCLAUSE
 [LAMBDA (C N)
  (OR (NUMCLAUSE (CONS (CN C)
                       N)
                (CN C))
      (INPUTERROR (QUOTE No% clause)
                  (CONS C N]))
```

```
(RPLBA
 [LAMBDA (BLK TXA)
  (RPLACA (LAST TXA)
          BLK)
  TXA])
```

```
(TEXTADDR
 [LAMBDA (I P)
  (PROG NIL
    [COND
      ((ZEROP I)
       (RETURN P))
      ((EQ I 1)
       (RETURN (TSUCC P)))
      (L1 (COND
          ((NULL (SETQ P (TSUCC P)))
           (RETURN NIL))
          ((NOT (EQP (TLNO P)
                    I))
           (GO L1)))
        (RETURN P])
```

```
(TXAFIND
 [LAMBDA (S J P)
  (PROG NIL
    S1 (COND
        ((NULL (SETQ P (TSUCC P)))
         (RETURN NIL))
        ((NOT (STRPOS S (TEXT P)))
         (GO S1)))
      (OR J (RETURN P))
    S2 (COND
        ((ZEROP J)
         (RETURN P))
        ((NULL (SETQ P (TSUCC P)))
         (RETURN NIL))
        (T (SETQ J (SUB1 J))
           (GO S2]))
```

(TXALAST  
[LAMBDA (P)  
(AND (TSUCC P)  
(LASTST P])

(KWSET  
[LAMBDA (L AUXP)  
(PROG ([RECLIST (MAPCONC [APPEND L (AND (NOT AUXP)  
(QUOTE (XCOMQ)  
(FUNCTION (LAMBDA (X)  
(MAPCAR (PKWDS X)  
(FUNCTION (LAMBDA (Y)  
(CONS Y X]  
  
WORDLIST)  
(RETURN (LIST RECLIST (SETQ WORDLIST (MAPCAR RECLIST  
(FUNCTION CAR)))  
(AND (NOT AUXP)  
(APPEND WORDLIST LISPXCOMS NIL])

(PKWDS  
[LAMBDA (A XOK)  
(PKWDS1 (PARSEDEF A))

(PKWDS1  
[LAMBDA (D)  
(COND  
((NULL D)  
NIL)  
[(NLISTP (CAR D))  
(SELECTQ (CAR D)  
((\$ID SPUNC)  
(AND (NOT XOK)  
(RTERM PKWDS NIL)))  
(LIST (CAR D)  
(OPTION REPEAT  
(OPTION (UNION (PKWDS1 (CDAR D))  
(PKWDS1 (CDR D))  
(EITHER (MAPREDC (CDAR D)  
(FUNCTION (LAMBDA (X Y)  
(UNION (PKWDS1 X)  
Y]

[LIST (UNION  
(PKWDS1 (CDAR D))  
(PKWDS1 (CDR D))

(COND  
((SETQ D (PARSEDEF (CAAR D)))  
(PKWDS1 D))  
((NOT XOK)  
(RTERM PKWDS NIL])





```

(PTRACE2
 [LAMBDA (Y)
  (COND
   ((NLISTP Y))
   ((NOT (FNTYP (CAR Y)))
    (LISPXPRT (CONS (CAR Y)
                    (QUOTE (is undefined)))
              T))
   (FLAG (APPLY (QUOTE ADVISE)
                Y))
   (T (APPLY* (QUOTE UNADVISE)
              (CAR Y]))
 )
)
(DEFLIST(QUOTE(
 (VXSET (5-AUG-74 . 1236))
 (EXECLP (5-AUG-74 . 1240))
 (GETPROMPT (5-AUG-74 . 1241))
 (AUXDESC (5-AUG-74 . 1237))
 (TRIM (5-AUG-74 . 1242))
 (VXUNDOPRINT (27-MAY-74 . 2333))
 (GET2A (5-AUG-74 . 1243))
 (GETBLKA (5-AUG-74 . 1244))
 (INBLK (5-AUG-74 . 1245))
 (TEXTADDR (5-AUG-74 . 1246))
 (KWSET (5-AUG-74 . 1247))
 (PTRACE (6-JUN-74 . 18))
 (EXPANDGROUPS (6-JUN-74 . 17))
 (PTRACE1 (6-JUN-74 . 18))
 ))(QUOTE EDITDATE))
 (RPAQ TRACELIST NIL)
 (RPAQ EXECUTIVE-NTE)
 (RPAQ LASTBLK NIL)
 (APPLYMAPC VXADVICE (FUNCTION ADVISE))
 (LISPXPRT (QUOTE (VXSET))
           T)
 (VXSET)
 (DECLARE: DONTCOPY
 (FILEMAP (NIL (2516 13481 (VXSET 2528 . 2621) (EXECLP 2625 . 3247) (GETPROMPT
 3251 . 3597) (AUXDESC 3601 . 3824) (EXEC1 3828 . 4711) (TRIM 4715 . 5071)
 (READFIRST 5075 . 5226) (FORLISPX? 5230 . 6031) (VX 6035 . 6742) (PRINTVXHIST1
 6746 . 7517) (VXUNDOPRINT 7521 . 7666) (FIXWORD 7670 . 8195) (INPUTERROR 8199
 . 8529) (INPUTERROR1 8533 . 8599) (GET2A 8603 . 8993) (GET1A 8997 . 9070)
 (GET2A1 9074 . 9141) (GETB2A 9145 . 9311) (GETBLKA 9315 . 9478) (INBLK 9482
 . 9584) (GETPATH 9588 . 9849) (GETNCLAUSE 9853 . 10044) (RPLBA 10048 . 10128)
 (TEXTADDR 10132 . 10422) (TXAFIND 10426 . 10844) (TXALAST 10848 . 10913) (KWSET
 10917 . 11302) (PKWDS 11306 . 11358) (PKWDS1 11362 . 12148) (PTRACE 12152
 . 12777) (EXPANDGROUPS 12781 . 13101) (PTRACE1 13105 . 13228) (PTRACE2 13232
 . 13478))))))
 STOP

```

(FILECREATED "28-AUG-74 20:14:09" VXL.;50 23676

changes to: FULLSWEEP ZAP

previous date: "28-AUG-74 11:47:04" VXL.;48)

*Edutjix*

(LISPXPRINT (QUOTE VXLCOMS)

T)  
[RPAQQ VXLCOMS ((PROP (PARSEDEF SENAME)  
BIKLIST QTOKEN PIVFILE OBLOP BLOP BLOC TRLIST)

(PROP TBSTL \* BTYPES)

[PROP (PARSEDEF SENAME)

\*  
(MAPREDC BTYPES (FUNCTION (LAMBDA (X Y)  
(UNION (TBSTL X)  
Y])

(PROP (PARSEDEF SENAME)  
\* XCOMLIST)

(PROP JFORM JPROC)

(ENDUMP VXL)

~~(DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY (ADDVARS~~

~~(NLAMA  
(NLAML INSBLKX)))~~

(DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS  
(ADDVARS (NLAMA)  
(NLAML INSBLKX])

(DEFLIST(QUOTE(

[BLKLIST ((EITHER (\$ID (\$ 1)  
(REPEAT \$ID)  
(; \$)  
(: QUOTE 1))  
((: NIL])

[QTOKEN ((EITHER (\$ID (: QUOTE 1))  
(\$NUM)  
(\$QUOTE (: QUOTE 1))  
(\$PUNC (: QUOTE 1])

(PIVFILE ((FILENAME)  
(: PIVOT)  
(# EXIFILE 2)  
(: QUOTE 1)))

[OBLOP ((EITHER ((BLOP))  
((: (GETBLKA LASTBLK))  
(: LASTPATH])

[BLOP ((EITHER [(BADDR)  
(EITHER (# \$NUM)  
((: NIL])  
(# (: LASTBLK)  
\$NUM])

[BLOC ((EITHER (\$NUM (: CN 1))  
((OBADDR)  
(: ROOTC 1])

```
(TRLIST ($ID ($ 1)
          (REPEAT $ID)
          (: $)
          (: QUOTE 1)))
))(QUOTE PARSEDEF))
(DEFLIST(QUOTE(
  (BLKLIST <module-list>)
  (QTOKEN <token>)
  (PIVFILE NONE)
  (OBLOP NONE)
  (BLOP <module-or-path>)
  (BLOC <module-or-context>)
  (TRLIST <trace-list>)
))(QUOTE SENAME))
(RPAQQ BTYPES (PROCEDURE DECLARATIONS))
(DEFLIST(QUOTE(
  (PROCEDURE (ISTAT DECQ))
  (DECLARATIONS (DSTAT))
))(QUOTE TBSTL))
(DEFLIST(QUOTE(
  (ISTAT ($NUM (ETTER (: (QID)
                       :))
              (:(NIL)))
          (XSTAT)
          (: QUOTE 1)
          (: PUTIN1 3)
          (: PMARK 1)))
  (DECQ ((DECSTAT)
         (: QUOTE 1)
         (: PUTINDEC 1)
         (: PMARK 1)))
  (DSTAT ((DATASTAT)
         (: QUOTE 1)
         (: PUTINDEC 1)
         (: PMARK 1)))
))(QUOTE PARSEDEF))
(DEFLIST(QUOTE(
  (ISTAT <numbered-statement>)
  (DECQ NONE)
  (DSTAT NONE)
))(QUOTE SENAME))
(RPAQQ XCOMLIST (EDITCOMQ LISTQ EXECCOMQ))
```

```

(DEFLIST(QUOTE(
  [EDITCOMQ ((EITHER [DELETE (EITHER (BLOCK (BADDR)
    (: DELBLKX 1))
    ((T2ADDR)
    (: DELETEx 1]
  (REN (: NIL)
    [EITHER ((T2ADDR))
    ((: (GETB2A LASTBLK]
    (EITHER [FROM $NUM (EITHER (BY $NUM)
    (: 2]
    (: 100)
    (: 10)))
    (: RENUMBX 4))
  ((EITHER (COPY (: T))
    (MOVE (: NIL)))
    (T2ADDR)
    TO $NUM (EITHER (BY $NUM)
    (: 2)))
    (: COPYX 4))
  (CHANGE (QTOKEN)
    (QTOKEN)
    (T2ADDR)
    (: CHANGEx 3))
  ((PROCDEC)
    (: PROCEDURE)
    (: INSBLKX 3))
  ((DECDEC)
    (: DECLARATIONS)
    (: INSBLKX 3))
  (LABEL (TADDR)
    (EITHER ((QID))
    (: NIL)))
    (: LABELX 2]
  [LISTQ ((EITHER [LIST (EITHER ((BLOP)
    (: 1000)
    (: LISTBLKX 3))
    ((T2ADDR)
    (: LISTSTLX 1))
    (: (LISTBLKX LASTBLK NIL 1000]
  [TH (EITHER [$NUM (: CN 1)
    (EITHER (%. $NUM (: CONS 2)
    (: PNUMCL 1))
    (: LISTCLAUSES 1]
    ((OBLOP)
    (: LISTHMX 2]
  (VALUES [EITHER ($NUM (: CN 1))
    (: (CURCTX]
    (: LISTCSL 1))
  ((EITHER (HIST (: NIL))
    (HISTALL (: T)))
  [EITHER ($NUM (: CN 1))
    (: (CURCTX]
    (: HISTX 2))

```

```

(JUSTIFY (BLOC)
  (EITHER (UP (: T))
    ((: NIL)))
  (EITHER (ONLY (: NIL))
    ((: T)))
  (: JDUMP 3))
(EFFORT (BLOC)
  (: (QUOTE TIME))
  (: PLOTCE 2))
(STATS (EITHER ((QID)
  (: LIST 1))
  ((: FILELST)))
  (: STATSX 1))
(TRACE (EITHER (NIL (: (PTRACE NIL)))
  (+ (TRLIST)
    (: (QUOTE ON))
    (: PTRACE 2))
  (- (TRLIST)
    (: (QUOTE OFF))
    (: PTRACE 2))
  ((TRLIST)
    (: PTRACE 1))
  (? (EITHER (? (: T))
    ((: NIL)))
    (: TRACEQX 1))
  ((: (PRINT TRACELIST T)))
[EXECCOMQ ((EITHER (MODULES (: (CHECKBLOCKS BLKNAMES NIL)))
  (REFS (: (CHECKBLOCKS BLKNAMES T)))
  (GET (PIVFILE)
    (: LOADX 1))
  (SAVE (BLKLIST)
    (: DUMPX 1))
  (FILE (PIVFILE)
    (BLKLIST)
    (: FILEX 2))
  (PATHS (OBADDR)
    (: PATHSX 1))
  (VPATHS (OBADDR)
    (: VPATHSX 1))
  (IN (BLOP)
    (: INBLOPX 2))
  ([EITHER [PROVE (: ((B NIL]
    [IPROVE (: ((B (QUOTE (T]
    [COUNTER (: ((B (QUOTE (NIL . T]
    [COUNTER (: ((B (QUOTE (T . T]
  (EITHER ($NUM (: CN 1)
    [(: (OR B (QUOTE (NIL]
    (: DOPROOF 2))
  ((OBLOP)
    (: T)
    (: B)
    (: CHECKX 4)))
  (: PROG 2))

```

↓  
TRACEX

```

      (SETUP (OBADDR)
            (EITHER (# SNUM)
                  ( (: NIL)))
            (: NIL)
            (: CHECKX 3))
      (KILL (OBLOP)
            (: KP 2))
      (WHERE (: (WHEREX)
              ))(QUOTE PARSEDEF))
      (DEFLIST(QUOTE(
        (EDITCOMQ <edit-command>)
        (LISTQ <print-command>)
        (EXECCOMQ <exec-command>)
      ))(QUOTE SENAME))
      (DEFLIST(QUOTE(
        (JPROC ("for " PJPROC))
      ))(QUOTE JFORM))
      (RPAQQ VXLPROPS ((ROOTC)
        (BLKU (CLR)
              HASH
              (SETHASHQ BLKU 50 2.0))
        (PFILE)
        (TBSTL (NOSET))
        (TBSD)))
      (RPAQQ VXLFNS
        (LISTBLKX LISTSTLX VXLSET XLSET TGSET EXEC INBLK1 STKWFIX ZAP HRECLAIM
          TFXTSWEEP FULLSWEEP HSWEAP KP CHECKX LISTHMX BLOPL PATHSX
          VPATHSX THEX LISTHML LISTHL THPATHS TPARSE GPATHS DOPATHS
          SHOWPATH BLOCKREFS UNITFILEP BLKREF CHECKBLOCKS TEBLKS
          RENUMBX COPYX CHANGEX LABELX PUTIN1 PUTIN3 PUTINDEC NEWLNO
          PMARK DELETEx DELBLKX INSBKX GENPROOT PJPROC LOADX DUMPX
          FILEX INBLOPX WHEREX STATSX HISTX TRACEQX))
      (RPAQQ VXLVARS (BTYPES FTLNO XCOMLIST))
      [RPAQQ VXLADVICE ((INBLK AFTER NIL (INBLK1 P]
      (APPLYMAPC VXLPROPS (FUNCTION NEWPROP))
      [DECLARE: EVAL@COMPILE
      (ADDTOVAR GLOBALVARS BLKU)
      ]
      (DEFINEQ
      (LISTBLKX
        [LAMBDA (P J LEVELS)
          (COND
            (J (TEXTPATH (GETPATH P J)))
            (T (LISTBLK P))
          )
        ]
      (LISTSTLX
        [LAMBDA (A)
          (PROG ((LEVELS 1))
            (LISTSTL (CAR A)
                    (CDR A))
          )
        ]

```

```

(VXLSET
 [LAMBDA NIL
  (SETQ BLK NAMES NIL)
  (SETQ LASTPATH NIL) ← (XPSET)
  (SFSET)
  (TGSET)
  (XLSET)
  (TLSET)])

```

```

(XLSET
 [LAMBDA NIL
  [MAPC BTYPES (FUNCTION (LAMBDA (X)
    (SETTBSD X (KWSET (TBSTL X)
      T))
    (SETQ STKWDS (PKWDS (QUOTE XSTAT)
      T))
    (CAR (SETQ XCOMDESC (CONS (QUOTE #)
      (NCONC1 (KWSET XCOMLIST)
        (FUNCTION STKWFIX]))

```

```

(XPSET
 [X NIL
 (MAPC (GETP 'CTX 'NEWPROPS)
 (F/X (INITPROP (CAR X]))

```

```

(TGSET
 [LAMBDA NIL
  (SETQ TRACEGROUPS (MAPCONC FILELST (FUNCTION (LAMBDA (X)
    (APPEND (FPROPSAVES X (QUOTE TRACEGROUP))

```

```

(EXEC
 [LAMBDA (XFILE)
  (PROG ((LISTLEVEL 1))
  (RETURN (EXECLP XCOMDESC XFILE (TBSD (TBTYPE LASTBLK))

```

```

(INBLK1
 [LAMBDA (P)
  (AUXDESC (TBSD (TBTYPE P))

```

```

(STKWFIX
 [LAMBDA (LEX)
  (PROG ((L (CDDR LEX)))
  (RETURN (COND
    ((AND (EQ (CAR LEX)
      (QUOTE $NUM))
      (EQ (CAR (COND
        [(EQ (CAR L)
          (QUOTE :))
          (AND (EQ (CADR L)
            (QUOTE $ID))
            (EQ (CADDR L)
              (QUOTE :))
            (SETQ L (CADDR L)
              (T L)))
          (QUOTE $ID))
        (FIXWORD (CDR L)
          STKWDS))
      (PARSE LEX (QUOTE ISTAT]))

```



```

(ZAP
[ LAMBDA (ALL ZAPFN) [MAPC (GETP 'MHASH 'NEWPROPS)
(COND (FX (CLRHASH (CAAR (OR (CDDR X) X])
((NULL ALL)
(HRECLAIM (FUNCTION FULLSWEEP)))
(T (SELECTQ ALL
((ALL T)
(CLRCTX)
[MAPC PROPERTIES (FUNCTION (LAMBDA (X)
(CLRHASH (CAR X])
(SELECTQ ALL
(ALL (HZAP)
(TEXTSWEEP))
(HRECLAIM (FUNCTION TEXTSWEEP)))
(SFSET))
(QUOTE ?)]

```

```

(HRECLAIM
[LAMBDA (MAPFN)
(OR (FN1YP MAPFN)
(HELP (QUOTE HRECLAIM)
(QUOTE "requires a functional argument"))))
(GSTART)
(APPLY* MAPFN (FUNCTION RMARK))
(GSWEEP])

```

```

(TEXTSWEEP
[LAMBDA (SFN) (* Uses ALL free.)
(MAPC BLKNAMES (FUNCTION (LAMBDA (Z)
(SETPATHL Z NIL)
(GENPROOT Z)
(MAPSTL Z NIL (FUNCTION (LAMBDA (X)
(SELECTQ ALL
(ALL (SETTRANSL X NIL)
(SETSEXP X NIL))
[T (SETTRANSL X (COND
((MEMBTRANSL (QUOTE TCANON)
X)
(APPLY* SFN (SEXP X))
(LIST (QUOTE TCANON]
(HELP])

```

(FULLSWEEP

[LAMBDA (SFN)

(\* Scan source program.)

[MAPC BLKNAMES (FUNCTION (LAMBDA (Z)  
[MAPSTL Z NIL (FUNCTION (LAMBDA (X)

(COND  
((MEMBTRANSL (QUOTE TCANON)  
X)

(APPLY\* SFN (SEXP X])

(APPLY\* SFN (PATHL Z]) (\* Scan data bases.)

[MAPC PROPERTIES (FUNCTION (LAMBDA (X)  
(HSWEEP (CAR X)  
SFN]

(\* Scan the stack.)

[PROG ((POS (STKPOS (QUOTE FULLSWEEP)))  
(TOP (STKARG 1 (STKNTH 1)))

P  
(J 0))

LP (COND  
((NEQ (SETQ P (FSTKARG J POS))  
TOP)  
(APPLY\* SFN (CAR P))  
(SETQ J (SUB1 J))  
(GO LP]

(\* Scan hash arrays. Assumes that HCARRAY and HCNILARRAY are  
not defined with NEWPROP!)

[MAPC (APPEND (GETP (QUOTE CHASH)  
✓ (QUOTE NEWPROP)  
(GETP (QUOTE HASH)  
(QUOTE NEWPROPS)))  
(FUNCTION (LAMBDA (X)  
(HSWEEP (CAAR (OR (CDDR X)  
X))

(\* Miscellaneous.)

SFN]  
[MAPC BASEMODES (FUNCTION (LAMBDA (X)  
(APPLY\* SFN (CAADR X])  
(AND ZAPFN (APPLY\* ZAPFN SFN])

(\* Do caller's scan.)

(HSWEEP

[LAMBDA (A SFN)

✓ (MAPHASH A (FUNCTION (LAMBDA (X)  
✓ (APPLY\* SFN (CDDR X])  
✓ (APPLY\* SFN (CDDR X])

```
(KP
[LAMBDA (P J)
[COND
(J (PROG [(Y (CAR (LAST (CADR (GETPATH P J)
[COND
((NEQ (CAR Y)
(QUOTE NOBIND))
(MAPC (CADAR Y)
(FUNCTION KILLTREE))
(MAPC (CAAR Y)
(FUNCTION KILLTREE))
(RPLACA Y (QUOTE NOBIND]
(RETURN J)))
(T (KILLSONS (ROOTC P))
(SETPATHL P NIL)
(DELTRANSL P (QUOTE GPATHS]
(GCCTX])
```

```
(CHECKX
[LAMBDA (P J FLAG B)
(ETIME (THPATHS P))
(PRINTQ "Paths ready.")
(THX P (BLOPL P J)
(COND
(FLAG B)
(T (QUOTE NONE)))
T])
```

```
(LISTHMX
[LAMBDA (P J)
(THX P (BLOPL P J)
T NIL])
```

```
(BLOPL
[LAMBDA (P J)
(COND
(J (LIST (GETPATH P J)))
(T (PATHL P])
```

```
(PATHSX
[LAMBDA (P)
(ETIME (THPATHS P))
(DOPATHS P NIL (PATHL P])
```

```
(VPATHSX
[LAMBDA (P)
(ETIME (THPATHS P))
(VERTPATHS P])
```

```

(THEx
 [LAMBDA (P L NODO GEN)
  (PROG ((TOP (ROOTC P)))
    (MAPC
      L
      (FUNCTION (LAMBDA (X)
        (TERPRI)
        (SHOWPATH X P)
        [COND
          (GEN (PRINT (MAPCAR (CADR (ETIME (CGEN (CADR X)
            TOP))))
            (FUNCTION CI]
          (COND
            ((NEQ NODO (QUOTE NONE))
              (PROG ([L (CADAAR (FLAST (CADR X)
                (ANY NIL))
                (COND
                  ((NULL L)
                    (PRINTQ "No proofs"))
                  (T [COND
                    ((EQ NODO T)
                      (SETQ ANY (LISTHML L)))
                    (T (MAPC L (FUNCTION (LAMBDA (C)
                      (COND
                        ((CAR (PROOFDONE C)))
                        (T (SETQ ANY T)
                          (DOPROOF C NODO)
                          (OR ANY (PRINTQ "All proofs done"))
                    (LISTHML
 [LAMBDA (CL)
  (COND
    ((NULL (CDR CL))
      (LISTHML CL NIL))
    (T (PRINTQ Common% clauses:)
      (PROG [(XL (GNCLAUSES (CAR CL)
        [MAPC (CDR CL)
          (FUNCTION (LAMBDA (C)
            (SETQ XL (DFILTER XL (FUNCTION (LAMBDA (X)
              (EQ (CLAUSENO (CAR X)
                C)
              (CDR X]
            (COND
              (XL (LISTCLL XL (CAR CL)))
              (T (PRINTQ % % none)))
            (RETURN (LISTHML CL XL])

```

```
(LISTHL
 [LAMBDA (CL XL)
  (PROG (ANY)
   [MAPC CL (FUNCTION (LAMBDA (C)
    (TERPRI)
    (PCTX C)
    (SPACES 1)
    (PJUST C)
    (TERPRI)
    (COND
     ((CAR (PROOFDONE C))
      (PCEFF C))
     (T (SETQ ANY T)
        (LISTCLAUSES C XL]
    (RETURN ANY]))
```

```
(THPATHS
 [LAMBDA (P)
  [MAPSTL P NIL (FUNCTION (LAMBDA (X)
   (DOTRANS X (QUOTE TPARSE]
  (DOTRANS P (QUOTE GRAFTREE))
  (SETCTX (ROOTC P))
  (DOTRANS P (QUOTE DECSCAN))
  [MAPSTL P NIL (FUNCTION (LAMBDA (X)
   (DOTRANS X (QUOTE TCANON]
  (DOTRANS P (QUOTE IOACHECK))
  (DOTRANS P (QUOTE GPATHS])
```

```
(TPARSE
 [LAMBDA (A)
  (PROG ((E (LEXSTAT (TEXT A)
   T))
   (LINE (TEXT A))
   (LN (TLNO A)))
  [SETQ E (COND
   ((GEO LN FTLNO)
    (PARSE (CONS (QUOTE $NUM)
     (CONS LN E))
     (QUOTE ISTAT)))
   ((NEQ LN 0)
    (AND (PARSE E (QUOTE DECQ)) (* *** Should store)
     (RETURN)))
   (T (RETURN]
  (COND
   (E (EVAL E))
   (T (SCANERROR1 (QUOTE Syntax% error]))
```

```
(GPATHS
 [LAMBDA (P)
  (PROG ((OLD (PATHL P))
   (NEW (FOLLOWPATHS P)))
  (SEGMRG OLD NEW)
  (SETPATHL P NEW])
```

(DOPATHS

```
[LAMBDA (P FN L)
  (MAPC L (FUNCTION (LAMBDA (X)
    (SHOWPATH X P)
    (COND
      (FN (APPLY* FN X]))
```

(SHOWPATH

```
[LAMBDA (X P)
  (PRINQ "Path #")
  (PRIN2 (INDEX X (PATHL P)))
  (PRINQ :%)
  (PRINTPATH (CADR X))
  (TERPRI)]
```

(BLOCKREFS

```
[LAMBDA (L)
  (PROG [(FL (SORT (DISTRIB L (FUNCTION PFILE))
    (FUNCTION (LAMBDA (X Y)
      (ALPHORDER (CAR X)
        (CAR Y)]
    [MAPC FL (FUNCTION (LAMBDA (Z)
      (COND
        ((UNITFILEP Z)
          (BLKREF (CADR Z])
    (MAPC FL (FUNCTION (LAMBDA (Z)
      (COND
        ((NOT (UNITFILEP Z))
          (PRINQ " File ")
          (PRIN2 (CAR Z))
          (PRINTQ :)
          (MAPC (CDR Z)
            (FUNCTION BLKREF])
```

(UNITFILEP

```
[LAMBDA (Z)
  (AND (NOT (CDDR Z))
    (EQ (CAR Z)
      (PACK (FRPLACA (QUOTE (NIL .PIVOT))
        (CADR Z]))
```

(BLKREF

```
[LAMBDA (X)
  (LISTBN X)
  (PROG ((R (BLKU X)))
    (COND
      ((NULL (TSUCC X))
        (PRINQ " (EMPTY)"))))
  (TERPRI)
  (COND
    ((LISTP R)
      (PRINQ " (from")
        [MAPC R (FUNCTION (LAMBDA (X)
          (PROG ((LASTPGN 0))
            (SPACES 1)
            (LISTLCN X)
          (PRINTQ %]))]
```

```

(CHECKBLOCKS
 [LAMBDA (LST REFS)
  (CLRBLKU)
  [AND
   REFS
   (MAPC BLKNAMES
    (FUNCTION (LAMBDA (X)
     (COND
      [(TSUCC X)
       (MAPSTL (TSUCC X)
        NIL
        (FUNCTION (LAMBDA (Y)
         (MAPC (TEBLKS Y)
          (FUNCTION (LAMBDA (Z)
           (SETBLKU Z
            (CONS Y (LISTP (BLKU Z]
            ((NULL (BLKU X))
             (SETBLKU X T]
    (BLOCKREFS LST])

```

```

(TEBLKS
 [LAMBDA (A)
  (UCALLS (TEXP A)
   NIL CGOPS])

```

```

(RENUMBX
 [LAMBDA (KEEP A N K)
  (COND
   [(AND (GEQ (TLNO (CAR A))
    FTLNO)
    (GEQ N FTLNO)
    (GREATERP K 0))
   (PROG ((A1 (CAR A))
    (A2 (CDR A))
    (P (TPGN (CAR A)))
    S)
    (COND
     ((NOT KEEP)
      (TDELT A1 A2)))
    LP (SETQ S (TSUCC A1))
    (PUTIN2 (TEXT A1)
     N P (COND
      (KEEP NIL)
      (T A1))
     (TLABEL A1)
     (TEXP A1))
    (COND
     ((NEQ A1 A2)
      (SETQ A1 S)
      (SETQ N (IPLUS N K))
      (GO LP]
  (T (INPUTERROR1 (QUOTE Bad% parameters.]

```

(COPYX

```
[LAMBDA (KEEP A N K)
 (PROG ((P (TPGN (CAR A)))
 (M N)
 (A1 (CAR A))
 (A2 (CDR A)))
```

```
LC [COND
 ((NEQ A1 A2)
 (SETQ A1 (TSUCC A1))
 (SETQ M (IPLUS M K))
 (GO LC))
 ((AND (LEQ (TLNO A1)
 M)
 (GEQ (TLNO A2)
 N))
 (INPUTERROR1 (QUOTE Illegal% move])
```

```
LP [COND
 ((AND (SETQ P (TSUCC P))
 (ILESSP (TLNO P)
 N))
 (GO LP))
 ((EQP (TLNO P)
 N)
 (INPUTERROR1 (QUOTE Illegal% move)))
 ((AND (TSUCC P)
 (LEQ (TLNO (TSUCC P))
 M))
 (INPUTERROR1 (QUOTE Not% enough% room])
 (RENUMBX KEEP A N K])
```

(CHANGEX

```
[LAMBDA (R S A)
 (PROG ((N O))
```

```
[MAPSTL (CAR A)
 (CDR A)
 (FUNCTION (LAMBDA (X)
 (PROG ((LN (TEXT X))
 (LNO (TLNO X))
 LN1 E)
 (COND
```

```
([NEQ N (SETQ N (IPLUS N (NSTROCC R LN])
 (SETQ LN1 (STRSUBST S R LN))
 [COND
```

```
([NULL (SETQ E (PARSE LN1
 (COND
 ((GEQ LNO FTLNO)
 (QUOTE XSTAT))
 (T (QUOTE DECQ))
```

```
(LISPXPRIHQ Error% in% line% )
```

```
(LISPXPRIHQ (TLNO X])
```

```
(PMARK (PUTIN2 LN1 (TLNO X)
 (TPGN X)
```

```
X
 (TLABEL X)
 E))
```

```
(LISTST X])
```

```
(PRIN2 N)
 (PRINTQ % changes])
```



```

(LABELX
[LAMBDA (A LB)
(COND
  ((ILESSP (TLNO A)
    FTLNO)
  (INPUTERROR (QUOTE Labels% not% allowed% below% line)
    FTLNO))
(T (SETTLABEL A LB]))

(PUTIN1
[LAMBDA (N LB E)
(COND
  ((ILESSP N FTLNO)
  (INPUTERROR (QUOTE Minimum% line% number% is)
    FTLNO))
(T (SETQ LASTLCN (PUTIN3 LINE N LASTBLK NIL LB E))
  (* USES LINE FROM VX)

(PUTIN3
[LAMBDA (LN N P A LB EXP)
(PUTIN2 (SUBSTRING LN
(COND
  (LB (STRPOS (QUOTE (% ))
    LN
    (STRPOS (QUOTE (:)
    LN
    (STRPOS (QUOTE (:)
    LN NIL NIL NIL T)
    NIL NIL T)
    T))
(T (STRPOS (QUOTE (% 48 49 50 51 52 53 54 55 56 57))
  LN NIL T)))
  -1)
  N P A LB EXP]]

(PUTINDEC
[LAMBDA (E)
(PUTIN2 (CONCAT H (QUOTE % )
  LINE)
  (NEWLNO LASTBLK 1 (SUB1 FTLNO))
  LASTBLK NIL NIL E])
  (* LINE is bound in VX)

(NEWLNO
[LAMBDA (P LO HI)
(PROG ((A P))
  L1 [COND
    ((NULL (SETQ A (TSUCC A)))
    (RETURN LO))
    ((IGREATERP (TLNO A)
    LO)
    (RETURN LO))
    ((EQP (TLNO A)
    LO)
    (COND
      ((IGREATERP (SETQ LO (ADD1 LO))
    HI)
      (INPUTERROR1 (QUOTE No% free% line% numbers]
    (GO L1]))

```





(INBLOPX  
[LAMBDA (P J)  
(INBLK P)  
(SETQ LASTPATH (COND  
(J (GETPATH P J)  
J))

(WHEREX  
[LAMBDA NIL  
(PRINQ "In ")  
(PRIN2 LASTBLK)  
(COND  
(LASTPATH (PRINQ #)  
(PRIN2 LASTPATH)))  
(TERPRI))

(STATSX  
[λ (L)  
(STATSPL L)  
(COND  
((EQ (OUTPUT) T)  
(APPENDFILE STATSFILE)  
(PRIN1 (USERNAME))  
(SPACES 2)

→ (STATSPL  
→ [LAMBDA (L) (TERPRI)  
→ (MAPC L (FUNCTION (LAMBDA (X) (COND ((FSTATS X)  
→ [PS (FSTATS X)] ← (TERPRI)

→ (PRINT1 (DATE)) ← (PRINT BLK NAMES)  
→ (STATSPL L)  
→ (CLOSEF (OUTPUT])

(HISTX  
[LAMBDA (ALL CTX)  
(COND  
[ALL (MAPREDC CTX (FUNCTION (LAMBDA (X Y)  
(LISTPH X])  
(T (LISTPH CTX])

(TRACEQX  
[LAMBDA (FLAG)  
(PRINT (COND  
[FLAG (MAPCAR TRACEGROUPS  
(FUNCTION (LAMBDA (X)  
(CONS X (MAPCAR (GETP X (QUOTE TRACEGROUP))  
(FUNCTION (LAMBDA (Y)  
(COND  
((LISTP Y)  
(CAR Y))  
(T Y])  
(T TRACEGROUPS])

)  
(DEFLIST(QUOTE(  
(VXLSET (5-AUG-74 . 118))  
(XLSET (5-AUG-74 . 1253))  
(TGSET (6-JUN-74 . 32))  
(EXEC (5-AUG-74 . 1254))  
(INBLK1 (5-AUG-74 . 1254))  
(ZAP (28-AUG-74 . 1932))  
(TEXTSWEEP (28-AUG-74 . 1124))  
(FULLSWEEP (28-AUG-74 . 1317))  
(KP (5-AUG-74 . 1257))  
(BLOCKREFS (18-JUN-74 . 1543))  
(UNITFILEP (18-JUN-74 . 1531))

(TRACEX  
[λ (L FLAG)  
(PTRACE L FLAG (F' TRACEDAP])

(TRACEDAP  
[λ (X) (PROG ((Z (OPMAKE X)) L) (RETURN  
(COND  
(Z (SETR L (ARGLIST (CAR Z)))  
← (CAR X) 'AFTER NIL  
← 'OPTRACE (KQUOTE X)  
(SELECTR (CADR Z)  
(ARB (CAR L))  
(FORM (CADR L))  
(CONS 'LIST L)  
'!VALUES)))

```

(BLKREF (1-AUG-74 . 2348))
(CHECKBLOCKS (18-JUN-74 . 1615))
(TEBLKS (18-JUN-74 . 1531))
(PMARK (26-AUG-74 . 1427))
(DELBLKX (26-AUG-74 . 1428))
(INSBLKX (26-AUG-74 . 1517))
(GENPROOT (26-AUG-74 . 1428))
(PJPROC (9-AUG-74 . 939))
(INBLOPX (5-AUG-74 . 1258))
(HISTX (26-JUL-74 . 131))
(TRACEQX (6-JUN-74 . 33))
)) (QUOTE EDITDATE))
(RPAQQ BTYPES (PROCEDURE DECLARATIONS))
(RPAQQ FTLNO 100)
(RPAQQ XCOMLIST (EDITCOMQ LISTQ EXECCOMQ))
(APPLYMAPC VXLADVICE (FUNCTION ADVISE))
(LISPPRINT (QUOTE (VXLSET))
T)

```

STATSFILE = PIVOT.STATS

```

(VXLSET)
[DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
(ADDOVAR NLAMA)
(ADDOVAR NLAML INSBLKX)
]

```

```

[DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVERS
(ADDOVAR NLAMA)
(ADDOVAR NLAML INSBLKX)
]

```

```

[DECLARE: DONTCOPY
(FILEMAP (NIL (5949 22516 (LISTBLKX 5961 . 6064) (LISTSTLX 6068 . 6173) (VXLSET
6177 . 6302) (XLSET 6306 . 6565) (TGSET 6569 . 6702) (EXEC 6706 . 6823) (INBLK1
6827 . 6881) (STKWFIX 6885 . 7728) (ZAP 7732 . 8085) (HRECLAIM 8089 . 8282)
(TEXTSWEEP 8286 . 8732) (FULLSWEEP 8736 . 9904) (HSWEEP 9908 . 10025) (KP
10029 . 10414) (CHECKX 10418 . 10613) (LISTHM 10617 . 10686) (BLOPL 10690
. 10777) (PATHSX 10781 . 10859) (VPATHSX 10863 . 10931) (THEX 10935 . 12152)
(LISTHML 12156 . 12805) (LISTHL 12809 . 13263) (THPATHS 13267 . 13587) (TPARSE
13591 . 14022) (GPATHS 14026 . 14168) (DOPATHS 14172 . 14322) (SHOWPATH 14326
. 14465) (BLOCKREFS 14469 . 14909) (UNITFILEP 14913 . 15039) (BLKREF 15043
. 15385) (CHECKBLOCKS 15389 . 15775) (TEBLKS 15779 . 15841) (RENUMBX 15845
. 16383) (COPYX 16387 . 17262) (CHANGEX 17266 . 17934) (LABELX 17938 . 18142)
(PUTIN1 18146 . 18381) (PUTIN3 18385 . 18769) (PUTINDEC 18773 . 18966) (NEWLNO
18970 . 19431) (PMARK 19435 . 19589) (DELETX 19593 . 19690) (DELBLKX 19694
. 19987) (INSBLKX 19991 . 20596) (GENPROOT 20600 . 20712) (PJPROC 20716 .
20803) (LOADX 20807 . 21005) (DUMPX 21009 . 21512) (FILEX 21516 . 21734) (
INBLOPX 21738 . 21832) (WHEREX 21836 . 21987) (STATSX 21991 . 22079) (HISTX
22083 . 22209) (TRACEQX 22213 . 22513))))))
STOP

```

[resbool (op l z) (resfix) op l z (mkor ~~list~~ (cons (caddr z) (mkor ~~list~~ (bxor

~~[resquant (op l z) (resfix) op l z (mkor (makefa (car l) t p) (makeex (car l) t (cond ((eq z (cdr l)) (list (caddr z) (bxor (caddr l) c j))) (t (list (cadr l) (bxor (caddr z) c j))~~

[resfu (op l z) (resfix) op l z (makefu (car l) t (caddr z)]

[rescond1 (op l z) (resfix) op l z (prog ((l1 l) pl) (r (caddr z)))

lp [cond ((eq (setg l1 (cdr l1)) z) (return (mkor (cons r (conc pl (mapcar (caddr l1) (if i) (f' caddr)) (setg pl (cons (car l1) pl))

[cond ((eq (setg l1 (cdr l1)) z)

(return (mkor (cons (rootof (car pl)) (conc (and lf (cdr pl)) (fplacd (fplaca pl r) nil)]

(go lp]

[rescond (op l z) (rescond1 op l z t]

[rescond (op l z) (rescond1 op l z nil]

~~[insproc (name) (bindctx (rootctx) (setmode name (mk1 'funmode nil)) (mode) (chproc name) (seteval name (mk1 'equal (makeconst name)]~~

[addproc (name) (resproc name) (cond ((eq (totype name) 'procedure)

[resproc (name) (bindctx (rootctx) (setmode name (mk1 'funmode nil))

[seteval name (mk1 'procedure) (makeconst name) (setoro name t)]

[deleteproc (name) (bindctx (rootctx) (setmode name

(removal name) (remoro name)]

NOTRES

A == 'EQUAL

A == 'NEQUAL

[xresval (e) (cond [(eql e 'restrict) ~~...~~

(prog ((c (curctx)) (r (caddr e))) (return  
(prog ((c2 (newctx c 'jresd (list ~~...~~ r)))  
(cl (xply (list (newctx c 'jresd (list r)))  
~~...~~)))

(setg goalist (nconc goalist cl))  
[x (putgoal (xresval (caddr x] e))

(putassert r c2)  
(~~...~~ fyplaca thlist c2)

(return (cadr e] (t e]

Utility stuff:  
1) End arg for DREV  
2) write SETHASH, remember names  
3) PRINTLEVEL in NPRINT: &, --

~~[xresval (e) (putgoal (caddr e)) ~~...~~~~

[makefunction (br e) (mk) \* function br e]

[altrel (l ~~...~~ r1 c1 r2 c2) (mk 'or (altrel l ~~...~~ nil even]

[altrel (l ~~...~~ r1) (cond [(null l) (and ~~...~~ (list (mk 'and l1]

~~(t (nconc (altrel (cdr l) (cons (makerel r1 (car l) c1) l))  
(altrel (car l) (cons (makerel r2 (car l) c2) l)) (not odd]~~

(t ([λ (z) (nconc (altrel (cdr l) z even)  
(altrel (car l) (fyplaca z (makerel<sup>r2</sup> (car l) c2)) (not even]  
(cons (makerel r1 (car l) c1) l1]

[reldiv (e u) (\* U is (UNITFACTORS L))

(λ (u1) (cond ((eql u1 1) u) (t (cons u1 u))))  
(unitwide e u]

XEROX

XEROX

E A R S

Filename: <DEUTSCH>VBOOL.;9

Creation Date: FRIDAY, 21 MAR 1975 16:36-PDT

Printed by: DEUTSCH

XEROX

XEROX



(FILECREATED "21-MAR-75 16:22:53" &lt;DEUTSCH&gt;VBOOL.;9 14017

changes to: NOTBOOL VBOOLVARS VBOOLPROPS BOOLSEQ BSUBSM BSUBREL BITEMS  
VBOOLGLOBALS

previous date: " 1-SEP-74 16:43:08" &lt;DEUTSCH&gt;VBOOL.;7)

```

(LISPXPRINT (QUOTE VBOOLCOMS)
  T T)
(RPAQQ VBOOLCOMS ((PROP (OPMAKE NOTOP OPMODE MCFN)
  NOT AND OR EQV NEQV)
  (PROP NOTOP RESTRICT)
  (PROP (EQUALPRED SFDEF)
  EQV NEQV)
  (ADDVARS (TPLIST (21 - (EVBOOL CLAUSE)
  C$1 EQV)
  (22 - (EVBOOL CLAUSE)
  C$1 NEQV))))
  (DECLARE: DOEVAL@COMPILE (ADDVARS (NOLINKFNS BOOLSEQ)))
  (ENDUMP VBOOL)))
(DEFLIST(QUOTE(
  (NOT (MAKENOT 1))
  [AND (MKAND ARB ((RESTRICT . RESBOOL]
  [OR (MKOR ARB ((RESTRICT . RESBOOL]
  (EQV (MAKEQV 2))
  (NEQV (MAKENEQV 2))
  ))(QUOTE OPMAKE))
(DEFLIST(QUOTE(
  (NOT (NOTNOT))
  (AND (NOTBOOL OR))
  (OR (NOTBOOL AND))
  (EQV NEQV)
  (NEQV EQV)
  ))(QUOTE NOTOP))
(DEFLIST(QUOTE(
  (NOT BOOL)
  (AND BOOL)
  (OR BOOL)
  (EQV BOOL)
  (NEQV BOOL)
  ))(QUOTE OPMODE))
(DEFLIST(QUOTE(
  (NOT (BOOL))
  (AND (BOOL . BOOL))
  (OR (BOOL . BOOL))
  (EQV (BOOL BOOL))
  (NEQV (BOOL BOOL))
  ))(QUOTE MCFN))
(DEFLIST(QUOTE(
  (RESTRICT (NOTRES))
  ))(QUOTE NOTOP))
(DEFLIST(QUOTE(
  (EQV T)
  (NEQV NIL)
  ))(QUOTE EQUALPRED))
(DEFLIST(QUOTE(
  [EQV ((VAR #1 (MODEX BOOL))
  (VAR #2 (MODEX BOOL)))
  (OR (AND #1 #2)
  (AND (NOT #1)
  (NOT #2]
  [NEQV ((VAR #1 (MODEX BOOL))
  (VAR #2 (MODEX BOOL)))
  (OR (AND #1 (NOT #2))
  (AND #2 (NOT #1]
  ))(QUOTE SFDEF))
  (ADDTOVAR TPLIST (21 - (EVBOOL CLAUSE)
  C$1 EQV)
  (22 - (EVBOOL CLAUSE)
  C$1 NEQV))

```

```

[DECLARE: DOEVAL@COMPILE
 (ADDTOVAR NOLINKFNS BOOLSEQ)
]
(RPAQQ VBOOLSTATS
 (BLO# BL1# BL2# BLMORE# BLOP# BLNOP# BLREL# BSUB# BSUBREL# BRLO# BRL1#
  BRL2# (BRLMORE#))
 (RPAQQ VBOOLBLKFNS (BC))
 [RPAQQ VBOOLPROPS ((NOTOP (NOSET MACRO))
 (NOTEXP (MACRO SETMACRO))
 HASH
 (SETHASHQ NOTEXPA 100 2.0)
 MHASH)
 (EQUALPRED (NOSET])
 (RPAQQ VBOOLFNS
 (MAKENOT NOTBOOL NOTQUANT NOTNOT NOTREQ NOTRES NOTOF BXOR BXORL MKAND
  MKOR BOOLSEQ BSUBSM BSUBREL BRELP BITEMS UIMPLY RIMPLY
  RCLASHAND RCLASHOR UCLASH RESBOOL MAKEQV MAKENEQV BC CONVNF
  CCONVNF CONVNF NFSIZE))
 (RPAQQ VBOOLVARS (CJROPS (BLENGTHS NIL)
 (BRLNGTHS NIL)))
 (RPAQQ VBOOLGLOBALS (CJROPS BLENGTHS BRLNGTHS))
 [RPAQQ VBOOLBLOCKS ((BOOLSEQ BOOLSEQ BITEMS BSUBSM BSUBREL BRELP
 (LOCALFREEVARS CJ NCJ OP NOP SL CL TL]
 (CS VBOOLSTATS)
[DECLARE: EVAL@COMPILE
 (ADDTOVAR GLOBALVARS BRLMORE# BRL2# BRL1# BRLO# BSUBREL# BSUB# BLREL# BLNOP#
  BLOP# BLMORE# BL2# BL1# BLO#)
]
[DECLARE: DOEVAL@COMPILE
 (ADDTOVAR BLKLIBRARY BC)
]
[DECLARE: DOEVAL@COMPILE
 (DEFLIST(QUOTE(
 [BC (LAMBDA (CJ)
 (COND (CJ (QUOTE AND))
 (T (QUOTE OR)
 ))(QUOTE BLKLIBRARYDEF))
 ])
 (DEFINEQ
 (NOTOP
 [LAMBDA (X)
 (FGETP X (QUOTE NOTOP])
 (NOTEXP
 [LAMBDA (X)
 (GETHASH X NOTEXPA])
 (SETNOTEXP
 [LAMBDA (X V)
 (PUTHASH X V NOTEXPA])
 (EQUALPRED
 [LAMBDA (X)
 (FGETP X (QUOTE EQUALPRED])
 )
 (DEFLIST(QUOTE(
 (NOTOP (NOTOP))
 (NOTEXP (NOTEXP SETNOTEXP))
 (EQUALPRED (EQUALPRED))
 ))(QUOTE NEWPROP))
 [DECLARE: DOEVAL@COMPILE
 (DEFLIST(QUOTE(
 [NOTOP ((X)
 (FGETP X (QUOTE NOTOP]
 (NOTEXP ((X)
 (GETHASH X NOTEXPA)))
 (SETNOTEXP ((X V)
 (PUTHASH X V NOTEXPA)))
 ))(QUOTE MACRO))
 ]

```

```

(/ADDPROP (QUOTE PROP)
  (QUOTE NEWPROPS)
  (QUOTE (NOTOP (NOSET MACRO))))
T)
(SETHASHQ NOTEXPA 100 2.0)
(/ADDPROP (QUOTE MHASH)
  (QUOTE NEWPROPS)
  (QUOTE (NOTEXP (MACRO SETMACRO)
    NOTEXPA)))
T)
(/ADDPROP (QUOTE PROP)
  (QUOTE NEWPROPS)
  (QUOTE (EQUALPRED (NOSET))))
T)
[DECLARE: EVAL@COMPILE
  (ADDTOVAR GLOBALVARS NOTEXPA)
]
(DEFINEQ
(MAKENOT
  [LAMBDA (E)
    (COND
      ((CONSTP E)
       (NOT E))
      ((NLISTP E)
       (MK1*(QUOTE NOT)
        E))
      ((NOTEXP E)
       (T (PROG ((OP (NOTOP (CAR E)))
                 (A (CDR E)))
                [SETNOTEXP E (SETQ A (COND
                  ((LISTP OP)
                   (BLKAPPLY* (CAR OP)
                              (CADR OP)
                              A))
                  (OP (MK1 OP A))
                  (T (MK1*(QUOTE NOT)
                    E))
                (RETURN A]))))
      (NOTBOOL
        [LAMBDA (OP L)
          (MK1 OP (FSORT (MAPCAR L (FUNCTION MAKENOT))
            (FUNCTION EXPORDER]))
          (* edited (21-MAR-75 . 1607))
      (NOTQUANT
        [LAMBDA (OP A)
          (MK1* OP (CAR A)
            (CADR A)
            (MAKENOT (CADDR A))
      (NOTNOT
        [LAMBDA (Z A)
          (CAR A)]
      (NOTREQ
        [LAMBDA (Z A)
          (MK1*(CAR Z)
            (CAR A)
            (IPLUS (CDR Z)
              (CADR A))
      (NOTRES
        [LAMBDA (Z A)
          (MK1*(QUOTE RESTRICT)
            (MAKENOT (CAR A)
              (CADR A))
          (* edited (1-SEP-74 . 1638))
      (NOTOF
        [LAMBDA (E)
          (MAKENOT E)]
          (* Formerly a MEMO)

```

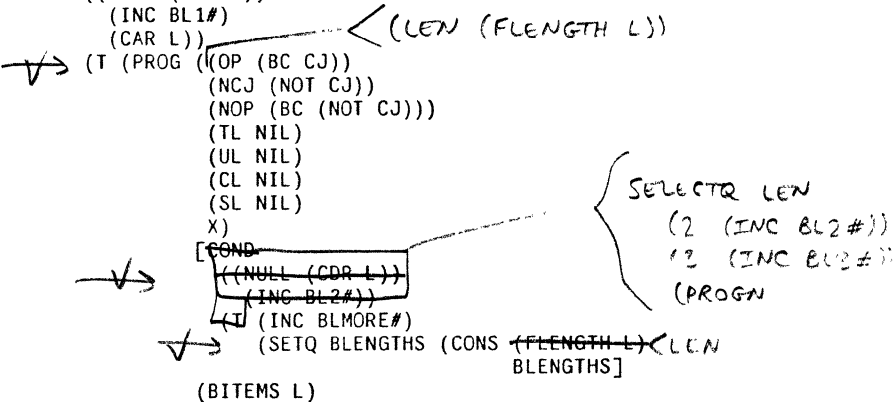
```
(BXOR
 [LAMBDA (E CJ) (* edited (24-AUG-74 . 2152))
 (COND
 (CJ (MAKENOT E))
 (T E]))
```

```
(BXORL
 [LAMBDA (L CJ) (* edited (24-AUG-74 . 2152))
 (COND
 (CJ (MAPRPC L (FUNCTION MAKENOT)))
 (T L]))
```

```
(MKAND
 [LAMBDA (L)
 (BOOLSEQ L T)])
```

```
(MKOR
 [LAMBDA (L)
 (BOOLSEQ L NIL)])
```

```
(BOOLSEQ
 [LAMBDA (L CJ) (* edited (21-MAR-75 . 1611))
 (COND
 ((NULL L)
 (INC BLO#)
 CJ)
 ((NULL (CDR L))
 (INC BL1#)
 (CAR L))
 (T (PROG ((OP (BC CJ))
 (NCJ (NOT CJ))
 (NOP (BC (NOT CJ)))
 (TL NIL)
 (UL NIL)
 (CL NIL)
 (SL NIL)
 X)
 [COND
 ((NULL (CDR L))
 (INC BL2#))
 (T (INC BLMORE#))
 (SETQ BLENGTHS (CONS (LENGTH L)
 BLENGTHS])
 (BITEMS L)
```



(\* SL is the list of items headed by NOP, CL is the list of lists of arithmetic relations with the same non-constant part, and TL is the list of all other items.)

```
L1 (COND
 (CL (SETQ X (CAR CL))
 [COND
 ((NULL (CDR X))
 (SETQ UL (FRPLACD X UL)))
 ((EQ (SETQ X (CJREL X CJ))
 NCJ) (* Can't be CJ)
 (RETURN NCJ))
 ((NEQ (CAR X)
 OP)
 (SETQ UL (CONS X UL)))
 (T (SETQ UL (APPEND (CDR X)
 UL])
 (SETQ CL (CDR CL))
 (GO L1)))
 (* UL now contains the items
 from CL after possible
 reductions by CJREL.)
 (SETQ CL (COND
 (SL (BSUBSM UL TL SL CJ))
 (TL (NCONC TL UL))
 (T UL)))
```



```

(BSUBREL
  [LAMBDA (R UL CJ)
    (INC BSUBREL#)
    (PROG ((A1 (CADR R))
           U1 U2 S)
      [COND
        [[AND (FMEMB (CAR R)
                    CJROPS)
              (SETQ U1 (SOME UL (FUNCTION (LAMBDA (X)
                                           (BRELP X A1)]
                                           (* Relevant units appear
                                           together on UL)
              [SETQ U2 (SOME (CDR U1)
                             (FUNCTION (LAMBDA (X)
                                         (NOT (BRELP X A1)]
                                         (* Can't be CJ)
              (SETQ S (CJREL (CONS R (LDIFF U1 U2))
                             CJ))
              (COND
                ((EQ S (NOT CJ))
                 (RTFRM BOOLSEQ S))
                ((EQ (CAR S)
                     (BC CJ))
                 (SETQ S (APPEND (CDR S)
                                 U2))
                 (FRPLACA U1 (CAR S))
                 (FRPLACD U1 (CDR S)))
                (T (FRPLACA U1 S)
                   (FRPLACD U1 U2]
                (T (SETQ UL (CONS R UL]
                  (RETURN UL]))

```

```

(BRELP
  [LAMBDA (X A)
    (AND (LISTP X)
         (EQ (CADR X)
              A)
         (FMEMB (CAR X)
                 CJROPS]))

```

```

(BITEMS
  [LAMBDA (L)
    (PROG (D A1 P X (Z L))
      LO [COND
        [(CONSTP (SETQ X (CAR Z)))
         (COND
           ((NEQ (CONSTVAL X)
                  CJ)
            (RTFRM BOOLSEQ NCJ]
          ((EQ (SETQ P (CAR (LISTP X)))
               OP)
           (INC BLOP#)
           (BITEMS (CDR X)))
          [(EQ P NOP)
           (INC BLNOP#)
           (OR (FMEMB X SL)
               (SETQ SL (CONS X SL]
          [(FMEMB P CJROPS)
           (INC BLREL#)
           (SETQ A1 (CADR X))
           (COND
             [[AND CL (SETQ D (SOME CL (FUNCTION (LAMBDA (Y)
                                                   (EQ (CADAR Y)
                                                       A1]
                 (OR (FMEMB X (CAR D))
                     (FRPLACA D (CONS X (CAR D]
                 (T (SETQ CL (CONS (LIST X)
                                   CL]

```

```

(T (COND
  ((FMEMB X TL))
  ((FMEMB (NOTOF X)
    TL)
  (RTFRM BOOLSEQ NCJ))
  ([OR (NULL TL)
    (NLISTP X)
    (NOTANY TL (FUNCTION (LAMBDA (Y)
      (AND (LISTP Y)
        (COND
          ((UCLASH Y X CJ)
            (RTFRM BOOLSEQ NCJ))
          (T (UIMPLY Y X CJ]
      )
    )
  )
  (SETQ TL (CONS X TL])
(COND
  ((SETQ Z (CDR Z))
  (GO LO])

```

```

(UIMPLY
  [LAMBDA (A B CJ)
    (OR (EQ A B)
      (AND (EQ (CADR A)
        (CADR B))
      (COND
        (CJ (RIMPLY (CAR A)
          (CADDR A)
          (CAR B)
          (CADDR B)))
        (T (RIMPLY (CAR B)
          (CADDR B)
          (CAR A)
          (CADDR A))
      )
  )

```

(INC UI#)  
(COND  
(\* (INC UIT#)  
T))

```

(RIMPLY
  [LAMBDA (R1 C1 R2 C2)
    (SELECTQ R1
      (EQUAL (APPLY* R2 C1 C2))
      (GEQ (SELECTQ R2
        (NEQUAL (LESSP C2 C1))
        (GEQ (LEQ C2 C1))
        NIL))
      (LEQ (SELECTQ R2
        (NEQUAL (GREATERP C2 C1))
        (LEQ (GEQ C2 C1))
        NIL))
      (NEQUAL (SELECTQ R2
        (NEQUAL (EQP C1 C2))
        NIL))
      (AND (CONSTP C1)
        (CONSTP C2)
        (EQUALPRED R1)
        (COND
          ((EQ R2 R1)
            (EQ C2 C1))
          ((EQ R2 (NOTOP R1))
            (NEQ C2 C1])
    )
  )

```

(\* edited (2-AUG-74 . 1437))

```

(RCLASHAND
  [LAMBDA (R1 C1 R2 C2)
    (RIMPLY R1 C1 (SELECTQ R2
      (LEQ (SETQ C2 (ADD1 C2))
        (QUOTE GEQ))
      (GEQ (SETQ C2 (SUB1 C2))
        (QUOTE LEQ))
      (AND (LITATOM (SETQ R2 (NOTOP R2)))
        R2))
    )
  )

```

(\* edited (2-AUG-74 . 1453))

C2])

```
(RCLASHOR
[LAMBDA (R1 C1 R2 C2) (* edited (2-AUG-74 . 1455))
(RIMPLY (SELECTQ R1
(LEQ (SETQ C1 (ADD1 C1))
(QUOTE GEQ))
(GEQ (SETQ C1 (SUB1 C1))
(QUOTE LEQ))
(AND (LITATOM (SETQ R1 (NOTOP R1)))
R1))
C1 R2 C2])
```

```
(UCLASH
[LAMBDA (A B CJ)
(AND (EQ (CADR A)
(CADR B))
(COND
(CJ (RCLASHAND (CAR A)
(CADDR A)
(CAR B)
(CADDR B)))
(T (RCLASHOR (CAR A)
(CADDR A)
(CAR B)
(CADDR B))
( INC UC#)
(COND
(* (INC UCT#)
T))
```

```
(RESBOOL
[LAMBDA (OP L Z) (* edited (25-AUG-74 . 912))
(RESFIX1 OP L Z (MKOR (CONS (CADDAR Z)
(BXORL (NCONC (LDIFF L Z)
(APPEND (CDR Z)))
(EQ OP (QUOTE AND]))
```

```
(MAKEQV
[LAMBDA (X Y) (* edited (13-AUG-74 . 1036))
(COND
((EQ X Y))
((EXPORDER Y X)
(MAKEQV Y X))
[(CONSTP X) (* T or NIL)
(COND
(X Y)
(T (MAKENOT Y]
(T (MK1*(QUOTE EQV)
X Y])
```

```
(MAKENEQV
[LAMBDA (X Y) (* edited (19-JUN-74 . 116))
(MAKENOT (MAKEQV X Y])
```

```
(BC
[LAMBDA (CJ)
(COND
(CJ (QUOTE AND))
(T (QUOTE OR])
```



```

(CONVNF
  [LAMBDA (E CJ)
    (PROG ((BOOLOP (BC CJ)))
      (RETURN (COND
        ((NLISTP E)
          E)
        [(EQ (CAR E)
          BOOLOP)
          (MKN E (MAPCARN (CDR E)
            (FUNCTION (LAMBDA (X)
              (CONVNF X CJ))
            (AND (EQ (CAR E)
              (BC (NOT CJ)))
            (FASSOC BOOLOP (CDR E)))
          (MK BOOLOP
            (MAPCAR [XXPROD (CDR E)
              (FUNCTION (LAMBDA (X)
                (COND
                  ((EQCAR (SETQ X
                    (CONVNF X CJ))
                    BOOLOP)
                  (CDR X))
                (T (LIST X]
              (FUNCTION (LAMBDA (X)
                (MK (CAR E)
                  X]
            (T E])
  ]

```

```

(CCONVNF
  [LAMBDA (E CJ)
    (COND
      ((CONVNF E CJ)
      (CONVNF E CJ))
    (T E])

```

```

(CONVNFP
  [LAMBDA (E CJ)
    (AND (LISTP E)
      (LEQ (NFSIZE E CJ)
        (ITIMES 2 (FLENGTH E))

```

```

(NFSIZE
  [LAMBDA (E CJ)
    (* edited (8-JUN-74 . 1922))
    (COND
      ((EQCAR E (BC CJ))
        (MAPREDC (CDR E)
          (FUNCTION [LAMBDA (X Y)
            (IPLUS (NFSIZE X CJ)
              Y])
          0))
      ((EQCAR E (BC (NOT CJ)))
        (MAPREDC (CDR E)
          (FUNCTION [LAMBDA (X Y)
            (ITIMES (NFSIZE X CJ)
              Y])
          1))
    (T 1])
  )
  (RPAQQ CJROPS (EQUAL NEQUAL LEQ GEQ))
  (RPAQ BLENGTHS NIL)
  (RPAQ BRLENGTHS NIL)
  [DECLARE: EVAL@COMPILE
  (ADDTOVAR GLOBALVARS CJROPS BLENGTHS BRLENGTHS)
  ]
  [DECLARE: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY
  (BLOCK: NIL MAKENOT NOTBOOL NOTQUANT NOTNOT NOTTREQ NOTRES NOTOF BXOR BXORL
    MKAND MKOR UIMPLY RIMPLY RCLASHAND RCLASHOR UCLASH RESBOOL MAKEQV
    MAKENEQV BC CONVNF CCONVNF CONVNFP NFSIZE)
  (BLOCK: BOOLSEQ BOOLSEQ BITEMS BSUBSM BSUBREL BRELP
    (LOCALFREEVARS CJ NCJ OP NOP SL CL TL))
  ]

```

(DECLARE: DONTCOPY

(FILEMAP (NIL (2857 3095 (NOTOP 2869 . 2919) (NOTEXP 2923 . 2971) (SETNOTEXP  
2975 . 3030) (EQUALPRED 3034 . 3092)) (3805 13478 (MAKENOT 3817 . 4204) (NOTBOOL  
4208 . 4372) (NOTQUANT 4376 . 4462) (NOTNOT 4466 . 4504) (NOTREQ 4508 . 4604)  
(NOTRES 4608 . 4762) (NOTOF 4766 . 4859) (BXOR 4863 . 4996) (BXORL 5000 .  
5152) (MKAND 5156 . 5197) (MKOR 5201 . 5243) (BOOLSEQ 5247 . 6781) (BSUBSM  
6785 . 7946) (BSUBREL 7950 . 8738) (BRELP 8742 . 8841) (BITEMS 8845 . 9922)  
(UIMPLY 9926 . 10173) (RIMPLY 10177 . 10734) (RCLASHAND 10738 . 11021) (RCLASHOR  
11025 . 11313) (UCLASH 11317 . 11536) (RESBOOL 11540 . 11763) (MAKEQV 11767  
. 12058) (MAKENEQV 12062 . 12177) (BC 12181 . 12257) (CONVNF 12261 . 12911)  
(CCONVNF 12915 . 13006) (CONVNFP 13010 . 13109) (NFSIZE 13113 . 13475))))))  
STOP

(FILECREATED "18-MAR-75 19:35:47" &lt;DEUTSCH&gt;VDS.;41 10910

changes to: VDSCOMS MAKEL MKSETEL SETELMERGE SETELCONST ELENUM ELCAT  
 ELCASE ELCASE1 MAKELENGTH MAKESUBSEQ SUBENUM SUBCONCAT SUBCAT1 SUBSETEL XLSUBSEQ  
 INDEXCHECK MKCONCAT CONCATITEM MODEL MODECONCAT MODESETEL MODESUBSEQ OMCONCAT  
 OMSUBSEQ OMCSEQ MAKEUSEQ OMCSEQ\* OMCSTRUCT MKCONSTX

previous date: " 1-SEP-74 16:47:14" &lt;DEUTSCH&gt;VDS.;40)

```
(LISPXPRT (QUOTE VDSCOMS)
  T T)
(RPAQQ VDSCOMS ((PROP (OPMAKE OPMODE MCFN XLFN)
  MD VAL ALLOC EQUALP NEQUALP FNMODE SEQMODE STRUCMODE
  PTRMODE ATMODE REF)
  (PROP (EQUALPRED NOTOP)
  EQUALP NEQUALP)
  (ADDVARS (CONSTATOMS NULL)
  (TPLIST (25 (- (& (LENGTH -)
  (? TGEQL))
  -)
  NIL C$1 LEQ)
  (95 (? EQSPLITP)
  (MATCHMIN CLAUSE)
  C$1 NEQUALP EQSPLIT)
  (97 (? EQSPLITP)
  (MATCHMIN CLAUSE)
  C$1 EQUALP EQSPLIT)))
  (ENDUMP VDS)))
(DEFLIST(QUOTE(
  (MD (MD 1 NIL T))
  (VAL (MAKEVAL 1 NIL T))
  (ALLOC (MAKEALLOC 1 NIL T))
  (EQUALP (MAKEEQUALP 2))
  (NEQUALP (MAKENEQUALP 2))
  (FNMODE (MK1 FORM))
  (SEQMODE (MK1 FORM))
  (STRUCMODE (MK1 FORM))
  (PTRMODE (MK1 FORM))
  (ATMODE (MK1 FORM))
  (REF (MK1 FORM))
  ))(QUOTE OPMAKE))
(DEFLIST(QUOTE(
  (MD MODE)
  (VAL (MODEVAL 1))
  (ALLOC (MODEALLOC 1))
  (EQUALP BOOL)
  (NEQUALP BOOL)
  (FNMODE MODE)
  (SEQMODE MODE)
  (STRUCMODE MODE)
  (PTRMODE MODE)
  (ATMODE MODE)
  (REF (MODEREF 1))
  ))(QUOTE OPMODE))
(DEFLIST(QUOTE(
  (MD NIL)
  (VAL OMVAL)
  (ALLOC OMCOMP)
  (EQUALP OMEQUALP)
  (NEQUALP OMEQUALP)
  (FNMODE (MODE . MODE))
  (SEQMODE (MODE))
  (STRUCMODE (MODE . MODE))
  (PTRMODE (MODE))
  (ATMODE NIL)
  (REF NIL)
  ))(QUOTE MCFN))
```

```

(DEFLIST(QUOTE(
  (MD NIL)
  (VAL XLVALL)
  (ALLOC XLALLOC)
  (EQUALP NIL)
  (NEQUALP NIL)
  (FNMODE NIL)
  (SEQMODE NIL)
  (STRUCMODE NIL)
  (PTRMODE NIL)
  (ATMODE NIL)
  (REF NIL)
)))(QUOTE XLFN))
(DEFLIST(QUOTE(
  (EQUALP T)
  (NEQUALP NIL)
)))(QUOTE EQUALPRED))
(DEFLIST(QUOTE(
  (EQUALP NEQUALP)
  (NEQUALP EQUALP)
)))(QUOTE NOTOP))
(ADDTOVAR CONSTATOMS NULL)
(ADDTOVAR TPLIST (25 (- (& (LENGTH -)
  (? TGEQL))
  -)
  NIL C$1 LEQ)
  (95 (? EQSPLITP)
  (MATCHMIN CLAUSE)
  C$1 NEQUALP EQSPLIT)
  (97 (? EQSPLITP)
  (MATCHMIN CLAUSE)
  C$1 EQUALP EQSPLIT))
(RPAQQ VDSPROPS ((OPMODE (NOSET MACRO))
  (MGNL NIL CTX)
  (MANL NIL CTX)
  (CONT NIL CTX)
  (MNMODE NIL CTX)))
(RPAQQ VDSFNS
  (TGEQL MAKEVAL XLVALL MAKEQUALP MK2C EQUALCAT EQUALSETEL EQSPLITP
  EQSPLIT EQSPLIT1 EQSUBXY MAKENEQUALP XLALLOC GENCONT AGTR
  BOMBEXP BADOBJECT NULOBJECT MODEALLOC MODEVAL MODEREF OMCOMP
  OMEQUALP OMVAL OMCONSTX MKCONSTX))
(RPAQQ VDSVARS NIL)
(DEFINEQ
(OPMODE
  [LAMBDA (X)
    (FGETP X (QUOTE OPMODE))]
(MGNL
  [LAMBDA (E CTX)
    (GETX E (QUOTE MGNL)
    CTX)]
(SETMGNL
  [LAMBDA (E V CTX)
    (PUTX E (QUOTE MGNL)
    V CTX)]
(MANL
  [LAMBDA (E CTX)
    (GETX E (QUOTE MANL)
    CTX)]
(SETMANL
  [LAMBDA (E V CTX)
    (PUTX E (QUOTE MANL)
    V CTX)]

```

```

(CONT
  [LAMBDA (E CTX)
    (GETX E (QUOTE CONT)
      CTX)])

(SETCONT
  [LAMBDA (E V CTX)
    (PUTX E (QUOTE CONT)
      V CTX)])

(MNMODE
  [LAMBDA (E CTX)
    (GETX E (QUOTE MNMODE)
      CTX)])

(SETMNODE
  [LAMBDA (E V CTX)
    (PUTX E (QUOTE MNMODE)
      V CTX)])
)
(DEFLIST(QUOTE(
  (OPMODE (OPMODE))
  (MGNL (MGNL SETMGNL))
  (MANL (MANL SETMANL))
  (CONT (CONT SETCONT))
  (MNMODE (MNMODE SETMNODE)))
)) (QUOTE NEWPROP))
[DECLARE: DOEVAL@COMPILE
(DEFLIST(QUOTE(
  [OPMODE ((X)
    (FGETP X (QUOTE OPMODE]
  )) (QUOTE MACRO))
])
  (/ADDPROP (QUOTE PROP)
    (QUOTE NEWPROPS)
    (QUOTE (OPMODE (NOSET MACRO))))
    T)
  (/ADDPROP (QUOTE CTX)
    (QUOTE NEWPROPS)
    (QUOTE (MGNL))
    T)
  (/ADDPROP (QUOTE CTX)
    (QUOTE NEWPROPS)
    (QUOTE (MANL))
    T)
  (/ADDPROP (QUOTE CTX)
    (QUOTE NEWPROPS)
    (QUOTE (CONT))
    T)
  (/ADDPROP (QUOTE CTX)
    (QUOTE NEWPROPS)
    (QUOTE (MNMODE))
    T)
(DEFINEQ
(TGEQL
  [LAMBDA (X)
    (PROG [(E (CTXEV (MAKEGEQ X 0]
      (COND
        ((NEQ E T)
         (CVASSERT E)
         T])
(MAKEVAL
  [LAMBDA (X)
    (COND
      ((EQCAR X (QUOTE ALLOC))
       (CADR X))
      (T (MK1* (QUOTE VAL)
        X)])

```

```

(XLVALL
 [LAMBDA (E)
  (PROG [(E1 (XLX (CADR E)
    (RETURN (COND
      [(NEQCAR E1 (QUOTE REF))
       (SETQ E1 (GENCONT (BOMBEXP E1)))
       (RESTRICT (CONT (CADR E1))
        (MAKEEQUALP E1 (QUOTE NULL])
       (T (CONT (CADR E1]))

(MAKEQUALP
 [LAMBDA (X Y)
  (OR (EQ X Y)
    (PROG ((M (MD X)))
      (RETURN (COND
        ((EQ M INTMODE)
         (MAKEQUAL X Y))
        ((EQ M BOOLMODE)
         (MAKEQV X Y))
        ((EQCAR Y (QUOTE CONCAT))
         (EQUALCAT X (CDR Y)))
        ((EQCAR Y (QUOTE SETEL))
         (EQUALSETEL Y X))
        ((NLISTP X)
         (MK2C (QUOTE EQUALP)
          X Y))
        ((EQ (CAR X)
         (QUOTE CONCAT))
         (EQUALCAT Y (CDR X)))
        ((EQ (CAR X)
         (QUOTE SETEL))
         (EQUALSETEL X Y))
        ((EQCAR Y (CAR X))
         (SELECTQ
          (CAR X)
          [(CONSTSEQ* CONSTRUCT)
           (AND (EQ (FLENGTH X)
            (FLENGTH Y))
            (MKAND (MAP2CAR (CDDR X)
              (CDDR Y)
              (FUNCTION MAKEQUALP]
           [CONSTSEQ (MKAND (MAKEQUAL (CADDR X)
            (CADDR Y))
            (MAKEQUALP (CADDR X)
              (CADDR Y))

          (QUOTE NIL)
          (REF (MAKEQUAL (CADR X)
            (CADR Y)))
          (MK2C (QUOTE EQUALP)
            X Y)))
        (T (MK2C (QUOTE EQUALP)
          X Y])

(MK2C
 [LAMBDA (OP X Y)
  (COND
    ((EXPORDER X Y)
     (MK1* OP Y X))
    (T (MK1* OP X Y]))
  (* Put constant second)

```

```

(EQUALCAT
  [LAMBDA (X L)
    (PROG ((P (LIST 0 0))
      E)
      [SETQ E
        (MAPCAR L
          (FUNCTION (LAMBDA (Y)
            (PROG [(N (MKPLUS (FRPLACA P (MAKELENGTH Y))
              (RETURN (MK*(QUOTE EQUALP)
                Y
                (MAKESUBSEQ X (MKPLUS (FRPLACA P 1))
                  (PROG1 N
                    (FRPLACA
                      (CDR P)
                      N]
                (RETURN (MKAND (CONS (MK (QUOTE EQUAL)
                  (FRPLACA P (MAKELENGTH X)))
                    E]))

```

```

(EQUALSETEL
  [LAMBDA (X Y)
    (PROG ([R (MKSETEL (CONS (CADR X)
      (CDDDDR X]
      (J (CADDR X))
      (V (CADDDR X))
      J1
      (L1 (MAKELENGTH (CADR X)))
      (L2 (MAKELENGTH Y)))
    (RETURN (MK* (QUOTE AND)
      (MAKEQUAL L1 L2)
      (MAKEQUALP V (MAKEL Y J))
      (MAKEQUALP (MAKESUBSEQ R 1 (SETQ J1 (MAKEPLUS J -1)))
        (MAKESUBSEQ Y 1 J1))
      (MAKEQUALP (MAKESUBSEQ R (SETQ J1 (MAKEPLUS J 1))
        L1)
        (MAKESUBSEQ Y J1 L1]))

```

```

(EQSPLITP
  [LAMBDA (E)
    (PROG [(M (MD (CADR E]
      (RETURN (SELECTQ (CAR M)
        (ATMODE (EQ M BOOLMODE))
        ((SEQMODE STRUCMODE)
          T)
        NIL]))

```

```

(EQSPLIT
  [LAMBDA (E)
    (PROG [(E1 (SELECTQ (CAR E)
      (EQUALP (EQSPLIT1 E))
      [NEQUALP (NOTOF (EQSPLIT1 (NOTOF E]
        (FAIL]
      (COND
        ((NEQ E1 E)
          (REPCLAUSE E E1 (QUOTE EQSPLIT)
            CTX))
        (T (FAIL]))

```

```

(EQSPLIT1
 [LAMBDA (E)
  (PROG ((X (CADR E))
        (Y (CADDR E))
        M)
  (RETURN
   (SELECTQ
    (CAR (SETQ M (MD X)))
    [STRUCMODE
     (PROG ((J 0))
      (RETURN (MK (QUOTE AND)
                  (MAPCAR (CDR M)
                          (FUNCTION (LAMBDA (Z)
                                    (MAKEQUALP (MAKEL X
                                                (SETQ J
                                                  (ADD1 J))))
                                                (MAKEL Y J]
          (SEQMODE (EQSUBXY X Y (QUOTE *SEQUAL))))
    E])

```

```

(EQSUBXY
 [LAMBDA (X Y FN)
  (SFCALL FN (LIST X Y])

```

```

(MAKENEQUALP
 [LAMBDA (X Y)
  (MAKENOT (MAKEQUALP X Y])

```

```

(XLALLOC
 [LAMBDA (E)
  (PROG [(E1 (XLX (CADR E)
                (RETURN (GENCONT E1 T])

```

```

(GENCONT
 [LAMBDA (E ALLOC)
  (* Generates a cell to hold E
  and returns a REF expression)

  (PROG ((M (MD E))
        (N (GENNAME))
        A)
  (SETMNODE N M)
  (SETVMODE N INTMODE)
  (SETCONT N E)
  (SETQ A (MANL M))
  [COND
   [ALLOC (COND
            (A (AGTR N (CAR A)))
            (T (MAPC (MGNL M)
                    (FUNCTION (LAMBDA (X)
                              (AGTR N X]
            (T (COND
                 (A (AGTR (CAR (FLAST A))
                          N)))
                (SETMGNL M (CONS N (MGNL M]
            (RETURN (MK1* (QUOTE REF)
                          N])

```

```

(AGTR
 [LAMBDA (X Y)
  (CVASSERT (MAKEGEQ (MAKEDIFF X Y)
                    1])

```



```

(BOMBEXP
  [LAMBDA (E)
    (PROG ((M (MD E))
      (PL (QUOTE (X @ NIL)))
      (N E)
      E1)
      (COND
        ((NEQ (CAR M)
          (QUOTE STRUCMODE))
          (RETURN E))
        ((EQCAR E (QUOTE CONSTRUCT))
          (RETURN E)))
      [COND
        ((NOT (VARP E))
          (SETQ E (GENNAME)]
        (FRPLACA PL N)
        [SETQ E1 (MK (QUOTE CONSTRUCT)
          (CONS M (MAPCAR (CDR M)
            (FUNCTION (LAMBDA (X)
              (FRPLACA (CDDR PL)
                (CADR X))
              (PROG ((Y (PACK PL)))
                (SETVMODE Y (CADDR X))
                (RETURN Y]
          (CVASSERT (MAKEQUALP E E1))
          (RETURN E1])

(BADOBJECT
  [LAMBDA (M)
    (RESTRICT (NULLOBJECT M)
      NIL])

(NULLOBJECT
  [LAMBDA (M)
    (SELECTQ (CAR M)
      [STRUCMODE (MK1 (QUOTE CONSTRUCT)
        (CONS M (MAPCAR (CDR M)
          (FUNCTION (LAMBDA (X)
            (NULLOBJECT (CADDR X]
          (SEQMODE (MK1 (QUOTE CONSTSEQ*)
            M))
          (PTRMODE (QUOTE NULL))
          (ATMODE (SELECTQ (CADADR M)
            (INT 0)
            (CHAR (MAKECONST (QUOTE % )))
            (REAL 0.0)
            (MODE NULLMODE)
            (HELP)))
          (HELP])

(MODEALLOC
  [LAMBDA (X)
    (MK1*(QUOTE PTRMODE)
      (MD X])

(MODEVAL
  [LAMBDA (X)
    (CADR (MD X])

(MODEREF
  [LAMBDA (X)
    (MNMODE X])

(OMCOMP
  [LAMBDA (L)
    (SELECTQ (CAR (MD (CAR L)))
      ((SEQMODE STRUCMODE)
        T)
      NIL])

```

```

(OMEQUALP
  [LAMBDA (L)
    (PROG [(M1 (MD (CAR L)))
          (M2 (MD (CADR L])
          (RETURN (OR (EQ M1 M2)
                     (AND (EQ M1 NULLMODE)
                          (EQ (CAR M2)
                              (QUOTE PTRMODE)))
                     (AND (EQ M2 NULLMODE)
                          (EQ (CAR M1)
                              (QUOTE PTRMODE]))))
    ]))

(OMVAL
  [LAMBDA (L)
    (EQCAR (MD (CAR L))
            (QUOTE PTRMODE))]

(OMCONSTX
  [LAMBDA (M L)
    (SELECTQ (CAR (LISTP M))
              (SEQMODE (OMCSEQ* (CONS M L)))
              (STRUCMODE (OMCSTRUCT (CONS M L)))
              [PTRMODE (AND L (NULL (CDR L))
                           (EQ (MD (CAR L))
                               (CADR M])
              NIL]))
    (* Called from OMCALL if FN is a
    mode)

(MKCONSTX
  [LAMBDA (M L)
    (SELECTQ (CAR (LISTP M))
              (SEQMODE (MK1 (QUOTE CONSTSEQ*)
                           (CONS M L)))
              (STRUCMODE (MK1 (QUOTE CONSTRUCT)
                           (CONS M L)))
              (PTRMODE (MK1 (QUOTE ALLOC)
                           L))
              (HELP]))
    (* edited (18-MAR-75 . 1935))
    (* Called from MKCALL if
    function is a mode)

)
(DECLARE: DONTCOPY
  (FILEMAP (NIL (2768 3383 (OPMODE 2780 . 2832) (MGNL 2836 . 2896) (SETMGNL
2900 . 2967) (MANL 2971 . 3031) (SETMANL 3035 . 3102) (CONT 3106 . 3166) (
SETCONT 3170 . 3237) (MNMODE 3241 . 3305) (SETMNMODE 3309 . 3380)) (4068 10886
(TGEQL 4080 . 4204) (MAKEVAL 4208 . 4327) (XLVALL 4331 . 4579) (MAKEQUALP
4583 . 5555) (MK2C 5559 . 5710) (EQUALCAT 5714 . 6153) (EQUALSETEL 6157 .
6645) (EQSPLITP 6649 . 6822) (EQSPLIT 6826 . 7062) (EQSPLIT1 7066 . 7480)
(EQSUBXY 7484 . 7540) (MAKENEQUALP 7544 . 7604) (XLALLOC 7608 . 7696) (GENCONT
7700 . 8308) (AGTR 8312 . 8387) (BOMBEXP 8391 . 9030) (BADOBJECT 9034 . 9103)
(NULLOBJECT 9107 . 9546) (MODEALLOC 9550 . 9616) (MODEVAL 9620 . 9662) (MODEREF
9666 . 9706) (OMCOMP 9710 . 9818) (OMEQUALP 9822 . 10091) (OMVAL 10095 . 10163)
(OMCONSTX 10167 . 10518) (MKCONSTX 10522 . 10883))))).
STOP

```

(FILECREATED "20-MAR-75 01:19:58" &lt;DEUTSCH&gt;VFAC.;7 7075

changes to: UDIVEXPT UNITDIVIDEP

previous date: "18-MAR-75 19:22:30" &lt;DEUTSCH&gt;VFAC.;6)

```

(LISPXPRINT (QUOTE VFACCOMS)
  T T)
(RPAQQ VFACCOMS ((PROP SFDEF *EXPT1P)
  (ENDUMP VFAC)))
(DEFLIST(QUOTE(
  [*EXPT1P ((VAR #1 (MODEX INT))
    (VAR #2 (MODEX INT)))
    (OR (EQUAL #1 1)
      (AND (NEQUAL #1 0)
        (EQUAL #2 0))
      (AND (EQUAL #1 -1)
        (EQUAL (REMAINDER #2 2)
          0)]
    ))(QUOTE SFDEF))
  (RPAQQ VFACFNS (UNITFACTORS UFACEXPT UNITDIVIDE UNITDIVIDEP UDIVEXPT TERMGCD
    FACTORS POWERFACP MAPALLFACTORS MAPALLFAC1))
  [RPAQQ VFACBLOCKS ((MAPALLFACTORS MAPALLFACTORS MAPALLFAC1 (LOCALFREEVARS
    MAPFN])
(DEFINEQ
(UNITFACTORS
  [LAMBDA (E)
    (COND
      ((NLISTP E)
        NIL)
      ((EQ (CAR E)
        (QUOTE TIMES))
        (CDDR E))
      ((AND (EQ (CAR E)
        (QUOTE PLUS))
        (ZEROP (CADR E)))
        (PROG ((L (CDDR E))
          U X XF N X1 X2 X3)
          .L1 [SETQ X
            (COND
              [(NULL L)
                (RETURN (COND
                  (XF (UFACEXPT U))
                  (T U))]
                ((EQCAR (SETQ X (CAR L))
                  (QUOTE TIMES))
                  (SETQ X1 (SETQ X3 NIL))
                  [MAPC
                    (SETQ X (CDDR X))

```

(\* edited (18-MAR-75 . 1922))

```

(FUNCTION (LAMBDA (Y)
  (COND
    ((EQCAR Y (QUOTE IEXPT))
     (PROG ((N (CADDR Y)))
      (COND
        ((NUMBERP N)
         (SETQ X1 (CONS Y X1)))
        ((EQCAR N (QUOTE PLUS))
         (COND
           ((IGEQ (CADR N)
                  0)
            [COND
              ((NEQ (CADR N)
                   0)
               (SETQ X1
                    (CONS (MK1*(QUOTE IEXPT)
                          (CADR N)
                          (CADR Y))
                          X1])
              [MAPC
                (CDDR N)
                (FUNCTION (LAMBDA (Z)
                  (SETQ X3
                       (CONS (MK1*(QUOTE IEXPT)
                             Z
                             (CADR Y))
                             X3])
                  (SETQ X2 (CONS Y X2])
                ]
              ]
            ]
          ]
        ]
      ]
    ]
    [(OR X1 X2)
     (NCONC [MAPCONC (SETQ XF X1)
                    (FUNCTION (LAMBDA (Y)
                      (REXPT (CADR Y)
                            (CADDR Y]
                    (NCONC X3 (NREML (NCONC X2 X1)
                                     X]
                      (T X)))
     ([AND (EQCAR X (QUOTE IEXPT))
          (NUMBERP (SETQ N (CADDR X]
          (SETQ XF (REXPT (CADR X)
                        N)))
     (T (LIST X]
    (COND
      ((NULL U)
       (SETQ U X))
      ((SETQ U (INTERSECTION U X)))
      (T (RETURN NIL)))
    (SETQ L (CDR L))
    (GO L1])
(UFACEXPT
 [LAMBDA (L)
  (PROG ((M L)
        N)
    L1 (COND
      ((NULL (CDR M))
       (RETURN L))
      ((NEQ (CAR M)
            (CADR M))
       (SETQ M (CDR M))
       (GO L1)))
      (SETQ N 2)
    L2 (FRPLACD M (CDDR M))
    (COND
      ((EQ (CAR M)
           (CADR M))
       (ADDIVAR N)
       (GO L2)))
    [SETQ M (CDR (FRPLACA M (MK1* (QUOTE IEXPT)
                                   (CAR M)
                                   N]
    (GO L1])

```



```
(TERMGCD
[LAMBDA (L)
  (PROG (D X)
    L1 (COND
      [(EQCAR (SETQ X (CAR L))
        (QUOTE TIMES))
        (SETQ X (ABS (CADR X)))
        (SETQ D (COND
          (D (GCD X D))
          (T X]
        (T (RETURN 1)))
      (COND
        ((SETQ L (CDR L))
          (GO L1)))
      (RETURN D])
```

```
(FACTORS
[LAMBDA (Y)

(* Y is a non-zero integer. Returns sorted list of positive
prime factors, smallest first.)
```

```
(PROG ((X Y)
  (J 3)
  L)
LX [COND
  ((ZEROP (LOGAND X 1))
    (SETQ L (CONS 2 L))
    (SETQ X (RSH X 1))
    (GO LX))
  ((MINUSP X)
    (SETQ X (IMINUS X]
  (COND
    ((EQ X 1)
      (RETURN L)))
  (SETQ J 3)
LP (COND
  [(IGREATERP (ITIMES J J)
    X)
    (RETURN (DREV (CONS X L]
  ((ZEROP (IREMAINDER X J))
    (SETQ L (CONS J L))
    (SETQ X (IQUOTIENT X J))
    (GO LP))
  (T (SETQ J (IPLUS J 2))
    (GO LP])
```

```
(POWERFACP
[LAMBDA (L)

(* L is a sorted list of prime factors.
If the product of L is A*B with B#1, returns
(A . B), otherwise NIL.)
```

```

(AND (EQ (CAR L)
         (CADR L))
      (PROG ((W L)
             A B C D)
            L1 (SETQ C 1)
            L2 (COND
                ((EQP (CADR W)
                      (CAR W))
                 (SETQ C (ADD1 C))
                 (SETQ W (CDR W))
                 (GO L2))
                ((NULL A)
                 (SETQ A (CAR W))
                 (SETQ B C)
                 )
                [(EQP B C)
                 (SETQ A (ITIMES A (CAR W))
                 ((EQ (SETQ D (GCD B C))
                      1)
                  (RETURN NIL))
                 (T [SETQ A (ITIMES (IEXPT A (IQUOTIENT B D))
                                   (IEXPT (CAR W)
                                           (IQUOTIENT C D))
                   (SETQ B D))])
            (COND
              ((SETQ W (CDR W))
               (GO L1)))
            (RETURN (CONS A B]))

```

```

(MAPALLFACTORS
 [LAMBDA (MAPL MAPFN)

```

```

  (* MAPL is a sorted list of prime factors.
  Applies MAPFN to all unique subproducts of L.)

```

```

  (MAPALLFAC1 MAPL 1])

```

```

(MAPALLFAC1
 [LAMBDA (MAPL MAPK)
  (PROG (MAPR)
   LP [COND
      ((NULL MAPL)
       (RETURN (CONS (APPLY* MAPFN MAPK)
                     MAPR])
       (SETQ MAPR (NCONC (MAPALLFAC1 (CDR MAPL)
                                     MAPK)
                         MAPR))
       LP1 (SETQ MAPK (ITIMES MAPK (CAR MAPL)))
       (COND
         ([EQP (CAR MAPL)
              (CAR (SETQ MAPL (CDR MAPL))
              (GO LP1)))
         (GO LP])
      )
  ]
[DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
 (BLOCK: NIL UNITFACTORS UFACEXPT UNITDIVIDE UNITDIVIDEP UDIVEXPT TERMGCD
  FACTORS POWERFACP)
 (BLOCK: MAPALLFACTORS MAPALLFACTORS MAPALLFAC1 (LOCALFREEVARS MAPFN))
]
[DECLARE: DONTCOPY
 (FILEMAP (NIL (736 6826 (UNITFACTORS 748 . 2372) (UFACEXPT 2376 . 2803) (
  UNITDIVIDE 2807 . 2934) (UNITDIVIDEP 2938 . 3879) (UDIVEXPT 3883 . 4484) (
  TERMGCD 4488 . 4786) (FACTORS 4790 . 5455) (POWERFACP 5459 . 6251) (
  MAPALLFACTORS 6255 . 6437) (MAPALLFAC1 6441 . 6823))))))
STOP

```

(FILECREATED "28-AUG-74 20:27:07" VIA.;57 15619

changes to: VIACOMS

previous date: "26-AUG-74 14:55:59" VIA.;56)

```
(LISXPRT (QUOTE VIACOMS)
  T)
(RPAQQ VIACOMS ((PROP (OPMAKE OPMODE MCFN)
  PLUS TIMES IEXPT MINUS DIFFERENCE QUOTIENT REMAINDER)
  (PROP SFDEF ABS SIGN QUOTIENT REMAINDER MAX MIN)
  (ENDUMP VIA)))
(DEFLIST(QUOTE(
  (PLUS (MKPLUS ARB))
  (TIMES (MKTIMES ARB))
  (IEXPT (MAKEIEXPT 2))
  (MINUS (MAKENEG 1))
  (DIFFERENCE (MAKEDIFF 2))
  (QUOTIENT (MAKEQUOTIENT 2))
  (REMAINDER (MK1 FORM))
  ))(QUOTE OPMAKE))
(DEFLIST(QUOTE(
  (PLUS INT)
  (TIMES INT)
  (IEXPT INT)
  (MINUS INT)
  (DIFFERENCE INT)
  (QUOTIENT INT)
  (REMAINDER INT)
  ))(QUOTE OPMODE))
(DEFLIST(QUOTE(
  (PLUS (INT . INT))
  (TIMES (INT . INT))
  (IEXPT (INT INT))
  (MINUS (INT))
  (DIFFERENCE (INT INT))
  (QUOTIENT (INT INT))
  (REMAINDER (INT INT))
  ))(QUOTE MCFN))
(DEFLIST(QUOTE(
  [ABS (((VAR #1 (MODEX INT)))
  (DCOND (GEQ #1 0)
  #1 T (MINUS #1]
  (SIGN (((VAR #1 (MODEX INT)))
  (DCOND (GEQ #1 0)
  1 T -1)))
```



```

(QUOTIENT (((VAR #1 (MODEX INT))
            (VAR #2 (MODEX INT)))
           (CHOICE (BV (VAR Q (MODEX INT)))
                   [OR (AND (GEQ #2 0)
                             (GEQ #1 (TIMES Q #2))
                             (LEQ #1 (PLUS #2 (TIMES Q #2)
                                           -1)))
                     (AND (LEQ #2 -1)
                           (GEQ #1 (TIMES Q #2))
                           (LEQ #1 (PLUS (DIFFERENCE (TIMES Q #2)
                                                       #2)
                                           -1)
                               -1]
                               Q)))
           [REMAINDER (((VAR #1 (MODEX INT))
                        (VAR #2 (MODEX INT)))
                     (DIFFERENCE #1 (TIMES (QUOTIENT #1 #2)
                                           #2)
                               #2]
                     (MAX (((VAR #1 (MODEX INT))
                            (VAR #2 (MODEX INT)))
                          (DCOND (GEQ #1 #2)
                                  #1 T #2)
                          T))
                     (MIN (((VAR #1 (MODEX INT))
                            (VAR #2 (MODEX INT)))
                          (DCOND (GEQ #1 #2)
                                  #2 T #1)
                          T))
                     ))(QUOTE SFDEF))
(RPAQQ VIABLKFN (LPLUS LTIMES GETFACS GETBASE))
(RPAQQ VIAPROPS ((NEGEXP (MACRO SETMACRO
                        HASH
                        → (SETHASHQ NEGEXPA 100 2.0) ←MHASH
                        (RPAQQ VIAFNS
                        (LPLUS SAMETERM CONSTERM GETTERMS FIXPLUS MAKEPLUS MKPLUS TLESS FLESS
                        GETBASE CONSTFAC GETFACS COPYFACS FACMUL LTIMES LSCMUL
                        MAKETIMES MKTIMES FIXTIMES SAMEFAC FIRSTERM REXPT LINEXP
                        MAKEIEXPT MAKECMUL MAKENEG MAKEDIFF NEGOF MAKEQUOTIENT MAKECDIV
                        CDIVTEST FLOORQUOTIENT))
(RPAQQ VIAVARS NIL)
(RPAQQ VIABLOCKS ((MKPLUS)
                  (FIXPLUS FIXPLUS SAMETERM)
                  (MKTIMES MKTIMES FACMUL CONSTFAC COPYFACS FIXTIMES SAMEFAC)))
(RPAQQ VIASTATS (FASTPLUS# SLOWPLUS# CANCEL# COMBINE# FASTTIMES# SLOWTIMES#
                  FACMUL# LSCMUL#))
(ADDTOVAR BLKLIBRARY LPLUS LTIMES GETFACS GETBASE)
[DECLARE: DOEVAL@COMPILE
(DEFLIST(QUOTE(
[LPLUS (LAMBDA (C L)
        (COND ((NULL L)
                C)
              ((AND (ZEROP C)
                    (NULL (CDR L)))
                (CAR L))
              (T (MK1 (QUOTE PLUS)
                      (CONS C L)

```

```

[LTIMES (LAMBDA (C L)
  (SELECTQ C (0 0)
    [1 (COND ((CDR L)
      (MK1 (QUOTE TIMES)
        (CONS C L)))
      (T (CAR L]
      (MK1 (QUOTE TIMES)
        (CONS C L]
    ]
  ]
[GETFACS (LAMBDA (X)
  (COND ((EQCAR X (QUOTE TIMES))
    (CDDR X))
    (T (LIST X]
  ]
[GETBASE (LAMBDA (X)
  (COND ((EQCAR X (QUOTE IEXPT))
    (CADR X))
    (T X]
  ]
)))(QUOTE BLKLIBRARYDEF))
]
(APPLYMAPC VIAPROPS (FUNCTION NEWPROP))
[DECLARE: EVAL@COMPILE
  (ADDTOVAR GLOBALVARS NEGEXPA)
]
(DEFINEQ
(LPLUS
  [LAMBDA (C L)
    (COND
      ((NULL L)
        C)
      ((AND (ZEROP C)
        (NULL (CDR L)))
        (CAR L))
      (T (MK1 (QUOTE PLUS)
        (CONS C L])
    ]
  ]
(SAMETERM
  [LAMBDA (X Y)
    (COND
      [(EQCAR X (QUOTE TIMES))
        (COND
          [(EQCAR Y (QUOTE TIMES))
            (AND (EQ (CDDR X)
              (CDDR Y))
              (PLUS (CADR X)
                (CADR Y]
            ((AND (EQ Y (CADDR X))
              (NULL (CDDDR X)))
              (ADD1 (CADR X]
          [(EQCAR Y (QUOTE TIMES))
            (AND (EQ X (CADDR Y))
              (NULL (CDDDR Y))
              (ADD1 (CADR Y]
          ((EQ X Y)
            2])
    ]
  ]

```

```
(CONSTERM
 [LAMBDA (X)
  (COND
   ((EQCAR X (QUOTE PLUS))
    (CADR X))
   ((NUMBERP X)
    X)
   (T 0])
```

```
(GETTERMS
 [LAMBDA (X)
  (COND
   ((EQCAR X (QUOTE PLUS))
    (CDDR X))
   ((NOT (NUMBERP X))
    (LIST X])
```

```
(FIXPLUS
 [LAMBDA (C Z)
  (PROG ((X Z)
        (Y (CDR Z))
        S)
    L1 (COND
       ((NULL Y)
        (RETURN (LPLUS C Z)))
       ([NULL (SETQ S (SAMETERM (CAR X)
                                (CAR Y))
        (SETQ Y (CDR (SETQ X Y)))
        (GO L1))
       [(EQ S 0)
        (INC CANCEL#)
        (COND
         ((SETQ Y (CDR Y))
          (FRPLACA X (CAR Y))
          (FRPLACD X (SETQ Y (CDR Y)))
          (GO L1))
         ((EQ Z X)
          (RETURN C))
         (T (FRPLACD (NLEFT Z 1 X)
                     NIL)
          (RETURN (LPLUS C Z]
       (T (INC COMBINE#)
          [FRPLACA X (LTIMES S (GETFACS (CAR X)
                                        (FRPLACD X (SETQ Y (CDR Y)))
                                        (GO L1])
```

```
(MAKEPLUS
 [LAMBDA (X Y)
  (MKPLUS (LIST X Y))
```

```

(MKPLUS
 [LAMBDA (L)
  (PROG ((C 0)
        (Z L)
        M M1 X)
  L1 [COND]
      [(NULL Z)
       (RETURN (COND
                [M1 (INC SLOWPLUS#)
                    (FIXPLUS C (MERGE (APPEND M) (FSORT (APPEND M M1)
                    (FUNCTION TLESS]
                (T (INC FASTPLUS#)
                    (LPLUS C M]
                ((NUMBERP (SETQ X (CAR Z)))
                 (SETQ C (PLUS X C)))
                [(AND (LISTP X)
                     (EQ (CAR X)
                         (QUOTE PLUS)))
                 (SETQ C (PLUS (CADR X)
                               C))
                (COND
                 ((NULL M)
                  (SETQ M (CDDR X)))
                 [M1 (SETQ M1 (MERGE (APPEND (CDDR X))
                 (FUNCTION TLESS]
                 (T (SETQ M1 (APPEND (CDDR X)
                 ((NULL M)
                  (SETQ M (LIST X)))
                 [M1 (SETQ M1 (MERGE (LIST X)
                 (FUNCTION TLESS]
                 (T (SETQ M1 (LIST X))
                   (SETQ Z (CDR Z))
                   (GO L1])
                   (CONS X M1)

```

```

(TLESS
 [LAMBDA (X Y)
  (EXPORDL (GETFACS X)
           (GETFACS Y))

```

```

(FLESS
 [LAMBDA (X Y)
  (EXPORDER (GETBASE X)
           (GETBASE Y))

```

```

(GETBASE
 [LAMBDA (X)
  (COND
   ((EQCAR X (QUOTE IEXPT))
    (CADR X))
   (T X])

```

```
(CONSTFAC
 [LAMBDA (X)
  (COND
   ((EQCAR X (QUOTE TIMES))
    (CADR X))
   (T 1]))
```

```
(GETFACS
 [LAMBDA (X)
  (COND
   ((EQCAR X (QUOTE TIMES))
    (CDDR X))
   (T (LIST X]))
```

```
(COPYFACS
 [LAMBDA (X Y)
  (COND
   ((EQCAR X (QUOTE TIMES))
    (APPEND (CDDR X)
             Y))
   (T (CONS X Y]))
```

```
(FACMUL
 [LAMBDA (X Y)
  (INC FACMUL#)
  (PROG [(C (TIMES (CONSTFAC X)
                  (CONSTFAC Y)))
         (L (MERGE (COPYFACS X)
                   (COPYFACS Y)
                   (FUNCTION FLESS])
         (RETURN (FIXTIMES C L]))
```

```
(LTIMES
 [LAMBDA (C L)
  (SELECTQ C
   (0 0)
   [1 (COND
       ((CDR L)
        (MK1 (QUOTE TIMES)
              (CONS C L)))
       (T (CAR L)
          (MK1 (QUOTE TIMES)
                (CONS C L]))
```

```
(LSCMUL
 [LAMBDA (L N)
  (INC LSCMUL#)
  (SELECTQ N
   (0 NIL)
   (1 (APPEND L))
   (MAPCAR L (FUNCTION (LAMBDA (X)
                        (MAKECMUL X N))
```

```

(MAKETIMES
  [LAMBDA (X Y)
    (MKTIMES (LIST X Y))

(MKTIMES
  [LAMBDA (L)
    (PROG ((C1 1)
           (L1 NIL)
           C2 L2 X)
      LP [COND
        [(NULL L)
          (RETURN (COND
                   [(CDR L1)
                    (INC SLOWTIMES#)
                    (FIXPLUS C1 (SORT L1 (FUNCTION TLESS])
                    (T (INC FASTTIMES#)
                       (LPLUS C1 L1]
                   [(NUMBERP (SETQ X (CAR L)))
                    (SETQ C1 (TIMES C1 X))
                    (AND L1 (SETQ L1 (LSCMUL L1 X)
                    ((NEOCAR X (QUOTE PLUS))
                    [AND L1 (MAPRPC L1 (FUNCTION (LAMBDA (Y)
                                                    (FACMUL Y X]
                    (SETQ L1 (SELECTQ C1
                                     (0 L1)
                                     (1 (CONS X L1))
                                     (CONS (COND
                                             ((EQCAR X (QUOTE TIMES))
                                              (LTIMES (TIMES C1 (CADR X))
                                                         (CDDR X)))
                                             (T (MK1*(QUOTE TIMES)
                                                C1 X)))
                                             L1)))
                    (SETQ C1 0))
          (T (SETQ C2 (CADR X))
            (SETQ L2 (CDDR X))
            [SETQ L1 (NCONC (LSCMUL L2 C1)
                           (LSCMUL L1 C2)
                           (MAPCONC L1 (FUNCTION (LAMBDA (Y)
                                                  (MAPCAR L2 (FUNCTION (LAMBDA (Z)
                                                                           (FACMUL Y Z]
                    (SETQ C1 (TIMES C1 C2]
          (SETQ L (CDR L))
          (GO LP])

```

```

(FIXTIMES
[LAMBDA (C Z)
  (PROG ((X Z)
        (Y (CDR Z))
        S)
    L1 (COND
        ((NULL Y)
         (RETURN (LTIMES C Z)))
        ([NULL (SETQ S (SAMEFAC (CAR X)
                                (CAR Y))
          (SETQ Y (CDR (SETQ X Y)))
          (GO L1))
         [(EQ S 0)
          (COND
            ((SETQ Y (CDR Y))
             (FRPLACA X (CAR Y))
             (FRPLACD X (SETQ Y (CDR Y)))
             (GO L1))
            ((EQ Z X)
             (RETURN C))
            (T (FRPLACD (NLEFT Z 1 X)
                       NIL)
              (RETURN (LTIMES C Z))
            (T (FRPLACA X (MAKEIEXPT (GETBASE (CAR X))
                                     S))
              (FRPLACD X (SETQ Y (CDR Y)))
              (GO L1]))

```

```

(SAMEFAC
[LAMBDA (X Y)
  (COND
    [(EQCAR X (QUOTE IEXPT))
     (COND
       [(EQCAR Y (QUOTE IEXPT))
        (AND (EQ (CADR X)
                 (CADR Y))
              (MKPLUS (CONS (CADDR X)
                             (CDDR Y))
                ((EQ (CADR X)
                     Y)
                 (MKPLUS (CONS 1 (CDDR X))
                [(EQCAR Y (QUOTE IEXPT))
                 (AND (EQ (CADR Y)
                         X)
                      (MKPLUS (CONS 1 (CDDR Y))
                ((EQ X Y)
                 2])

```

```

(FIRSTERM
[LAMBDA (X)
  (COND
    ((EQCAR X (QUOTE PLUS))
     (CADDR X))
    ((NOT (NUMBERP X))
     X])

```

```
(REXPT
 [LAMBDA (E N)
 (COND
 ((NEQ N 0)
 (CONS E (REXPT E (SUB1 N]))
```

```
(LINEXP
 [LAMBDA (E FLAG)
 (OR (NLISTP E)
 (SELECTQ (CAR E)
 (EL FLAG)
 [(PLUS TIMES)
 (EVERY (CDR E)
 (FUNCTION (LAMBDA (X)
 (LINEXP X FLAG)
 NIL]))]
```

```
(MAKEIEXPT
 [LAMBDA (X Y)
 (SELECTQ Y
 (0 1)
 (1 X)
 (SELECTQ X
 ((0 1)
 X)
 (COND
 ((EQCAR X (QUOTE IEXPT))
 (MAKEIEXPT (CADR X)
 (MK* (QUOTE TIMES)
 (CADDR X)
 Y)))
 ([AND (NUMBERP Y)
 (OR (EQCAR X (QUOTE PLUS))
 (EQCAR X (QUOTE TIMES))
 (MK (QUOTE TIMES)
 (REXPT X Y)))
 (T (MK1*(QUOTE IEXPT)
 X Y]))]
```

```
(MAKECMUL
 [LAMBDA (E C)
 (SELECTQ C
 (0 0)
 (1 E)
 (COND
 ((CONSTP E)
 (TIMES C (CONSTVAL E)))
 ((NLISTP E)
 (MK1*(QUOTE TIMES)
 C E))
 (T (SELECTQ (CAR E)
 (TIMES (LTIMES (TIMES C (CADR E))
 (CDDR E)))
 [PLUS (MK1 (QUOTE PLUS)
 (CONS (TIMES C (CADR E))
 (LSCMUL (CDDR E)
 C]
 (MK1*(QUOTE TIMES)
 C E]))]
```



(MAKENEG  
[LAMBDA (X)  
 (MAKECMUL X -1)])

(MAKEDIFF  
[LAMBDA (X Y)  
 (MAKEPLUS X (MAKECMUL Y -1))])

(NEGOF  
[LAMBDA (E)  
 (OR (NEGEXP E)  
 (PROGN (SETNEGEXP E (SETQ E ([LAMBDA (EVALARGS)  
 (MK (QUOTE MINUS)  
 (LIST E)  
 NIL))))  
 E]))]

(MAKEQUOTIENT  
[LAMBDA (X Y)  
 (COND  
 ((EQ X Y)  
 1)  
 ((EQ X 0)  
 0)  
 ((NUMBERP Y)  
 (MAKECDIV X Y))  
 ([AND ~~ZEROP (CONSTERN X)~~ (LISTP X)  
 ~~EVERY (GETTERMS X)~~ (UNITDIVIDEP X  
 (FUNCTION (LAMBDA (Z)  
 (OR (EQ Y Z)  
 (AND (EQCAR Z (QUOTE TIMES))  
 (MEMB Y (CDR Z))  
 (FRPLACA (QUOTE (T))  
 Y]))  
 (UNITDIVIDE X (LIST Y)))  
 (T (MK1\*(QUOTE QUOTIENT)  
 X Y]))]

(MAKECDIV  
[LAMBDA (X Y)  
 (COND  
 ((MINUSP Y)  
 (MAKECDIV (MAKENEG X)  
 (MINUS Y)))  
 ((CDIVTEST X Y))  
 (T (MK1\*(QUOTE QUOTIENT)  
 X Y]))]

```

(CDIVTEST
 [LAMBDA (X Y)
  (COND
   ((EQ Y 1)
    X)
   ((NUMBERP X)
    (FLOORQUOTIENT X Y))
   [(EQCAR X (QUOTE TIMES))
    (AND (ZEROP (REMAINDER (CADR X)
                           Y))
          (LTIMES (QUOTIENT (CADR X)
                             Y)
                  (CDDR X)
                  ((EQCAR X (QUOTE PLUS))
                   (LPLUS (FLOORQUOTIENT (CADR X)
                                           Y)
                           (MAPCAR (CDDR X)
                                    (FUNCTION (LAMBDA (Z)
                                              (OR (CDIVTEST Z Y)
                                                  (RTFRM CDIVTEST))))))
                   (RTFRM CDIVTEST)))]
  )
(FLOORQUOTIENT
 [LAMBDA (X Y)
  (COND
   ((MINUSP Y)
    (FLOORQUOTIENT (IMINUS X)
                    (IMINUS Y)))
   (T (IQUOTIENT (COND
                  ((MINUSP X)
                   (IDIFFERENCE (ADD1 X)
                                 Y))
                  (T X))
                 Y))
  )
(DECLIST(QUOTE(
))(QUOTE EDITDATE))
[DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
 (BLOCK: NIL LPLUS CONSTERM GETTERMS MAKEPLUS TLESS FLESS GETBASE GETFACS
  LTIMES LSCMUL MAKETIMES FIRSTERM REXPT LINEXP MAKEIEXPT MAKECMUL
  MAKENEG MAKEDIFF NEGOF MAKEQUOTIENT MAKECDIV CDIVTEST FLOORQUOTIENT)
 (BLOCK: MKPLUS)
 (BLOCK: FIXPLUS FIXPLUS SAMETERM)
 (BLOCK: MKTIMES MKTIMES FACMUL CONSTFAC COPYFACS FIXTIMES SAMEFAC)
]
 (CS VIASTATS)
 (LISPXPRINT (QUOTE VIACOMS)
  T)
 (RPAQQ VIACOMS ((PROP (OPMAKE OPMODE MCFN)
                       PLUS TIMES IEXPT MINUS DIFFERENCE QUOTIENT REMAINDER)
                 (PROP SFDEF ABS SIGN QUOTIENT REMAINDER MAX MIN)
                 (ENDUMP VIA)))
(DECLARE: DONTCOPY
 (FILEMAP (NIL (3262 14924 (LPLUS 3274 . 3462) (SAMETERM 3466 . 3965) (CONSTERM

```

3969 . 4100) (GETTERMS 4104 . 4234) (FIXPLUS 4238 . 5107) (MAKEPLUS 5111 .  
5162) (MKPLUS 5166 . 6408) (TLESS 6412 . 6488) (FLESS 6492 . 6570) (GETBASE  
6574 . 6673) (CONSTFAC 6677 . 6777) (GETFACS 6781 . 6886) (COPYFACS 6890 .  
7028) (FACMUL 7032 . 7291) (LTIMES 7295 . 7579) (LSCMUL 7583 . 7781) (MAKETIMES  
7785 . 7838) (MKTIMES 7842 . 9517) (FIXTIMES 9521 . 10375) (SAMEFAC 10379  
. 10875) (FIRSTERM 10879 . 11004) (REXPT 11008 . 11096) (LINEXP 11100 . 11388)  
(MAKEIEXPT 11392 . 12174) (MAKECMUL 12178 . 12910) (MAKENEG 12914 . 12959)  
(MAKEDIFF 12963 . 13023) (NEGOF 13027 . 13280) (MAKEQUOTIENT 13284 . 13765)  
(MAKECDIV 13769 . 13961) (CDIVTEST 13965 . 14578) (FLOORQUOTIENT 14582 . 14921)))  
\*\*)

)  
STOP

(FILECREATED "28-AUG-74 07:29:36" VPROC.;33 14594

changes to: MODEPROCDEF PROCMODE VPROCCOMS XLFUNCTION IOACHECK MODECALL  
 OMCALL MKCALL PROCDEF SFSUBP SFSUB1

previous date: "26-AUG-74 20:08:45" VPROC.;29)

```
(LISPXPRINT (QUOTE VPROCCOMS)
  T)
(RPAQQ VPROCCOMS ((PROP (OPMAKE OPMODE MCFN XLFN)
  CALL FUNCTION PROCDEF)
  (ENDUMP VPROC)))
(DEFLIST(QUOTE(
  (CALL (MKCALL ARB))
  [FUNCTION (MAKEFUNCTION 2 ((LCOND
    (CHOICE)
    (RESTRICT]
    (PROCDEF (MK1 FORM))
  ))(QUOTE OPMAKE))
(DEFLIST(QUOTE(
  (CALL (MODECALL ARB))
  (FUNCTION (MODEFUNCTION 2))
  (PROCDEF (MODEPROCDEF 1))
  ))(QUOTE OPMODE))
(DEFLIST(QUOTE(
  (CALL OMCALL)
  (FUNCTION NIL)
  (PROCDEF NIL)
  ))(QUOTE MCFN))
(DEFLIST(QUOTE(
  (CALL XLCALL)
  (FUNCTION XLFUNCTION)
  (PROCDEF NIL)
  ))(QUOTE XLFN))
[RPAQQ VPROCPROPS ((FNVAL NIL CTX)
  ((FVPAT FVCOND FVVAL . FVHIST]
(RPAQQ VPROCFNS
  (MAKEFUNCTION MODEFUNCTION XLFUNCTION ADDPROC RESETPROC DELETEPROC
  IOACHECK IOACK IOFREE OMCALL EVERY2 MKCALL MODECALL
  SFSUBP SFSUB SFSUB1 FNSUBP FNSUBP1 FNSUB FNSUB1 XLCALL
  PROCDEF RESULTNAME MODEPROCDEF PROCMODE SUBIODEC SFCALL
  PARSUB PARMATCH MODEMATCH MODEMATCH1 PROCINDECS
  PROCOUTDECS FVALUE FNVRET FNHISTP ADDFNVALUE
  ADDFNATTERN))
(RPAQQ VPROCVARS NIL)
[RPAQQ VPROCBLOCKS ((SFSUBP)
  (SFSUB SFSUB SFSUB1 (LOCALFREEVARS ALL))
  (FNSUBP FNSUBP FNSUBP1 (LOCALFREEVARS PFLAG))
  (FNSUB FNSUB FNSUB1 (LOCALFREEVARS CL COND PFLAG]
  (APPLYMAPC VPROCPROPS (FUNCTION NEWPROP))
(DEFINEQ
```

```
(MAKEFUNCTION
 [LAMBDA (BV E)
  (MK1* (QUOTE FUNCTION)
   BV E])
```

```
(MODEFUNCTION
 [LAMBDA (V E)
  ([LAMBDA (BV MODES)
   (MK1 (QUOTE FNMODE)
    (CONS (MD E)
     (MAPCAR (CDR V)
      (FUNCTION CADDR]
   (ADDBV V BV MODES])
```

```
(XLFUNCTION
 [LAMBDA (XLE)
  (PROG ((XLBV XLBV)
   (BV MODES BV MODES)
   (BV (CADR XLE)))
  (SETQ XLBV (APPEND (BVL BV)
   XLBV))
  (SETQ BV MODES (ADDBV BV BV MODES))
  (RETURN (MAKEFUNCTION BV ([LAMBDA (NCASEFLAG)
   (XLX (CADR XLE]
   T]))
```

```
(ADDPROC
 [LAMBDA (NAME)
  (RESETPROC NAME)]
```

```
(RESETPROC
 [LAMBDA (NAME)
  (COND
   ((EQ (TBTYPENAME)
    (QUOTE PROCEDURE))
   (PROG ((CTX (ROOTCTX)))
    (SETVMODE NAME (MK1 (QUOTE FNMODE)
     NIL)
     CTX)
    (SETCVAL NAME (MK1* (QUOTE PROCDEF)
     (MAKECONST NAME))
     CTX)
    (SETVRO NAME T CTX])
```

```
(DELETEPROC
 [LAMBDA (NAME)
  (PROG ((CTX (ROOTCTX)))
   (REVMODE NAME CTX)
   (REMCVAL NAME CTX)
   (REMVRO NAME CTX])
```

```

(IOACHECK
 [LAMBDA (P)
  (COND
   ((EQ (TBTYPE P)
        (QUOTE PROCEDURE))
    (PROG ((CTX (ROOTC P))
           (OUT (PROCOUTDECS P)))
          (MAPC (PROCINDECS P)
                (FUNCTION IOACK))
          (COND
           (OUT)
           ((OR (NEQCAR (SETQ OUT (SEXP (TLAST P)))
                    (QUOTE SETQ))
                (NEQ (CADR OUT)
                     (RESULTNAME P)))
            (SCANERROR1
             (COND
              ((PROCOUTDECS P T)
               (QUOTE "All output assertions involve local variables"))
              (T (QUOTE "No output assertions"))))
             (TLAST P)))
          ((SETQ OUT (IOFREE (CADR OUT)
                            CTX))
           (SCANERROR OUT (QUOTE "not allowed in final assignment")
                       (TLAST P]))
   ))

```

```

(IOACK
 [LAMBDA (A L)
  (PROG ((VL (IOFREE (CADR (SEXP A))
                    CTX L)))
        (* Uses CTX free)
        (COND
         (VL (SCANERROR VL (QUOTE "not allowed in I/O assertion")
                          A))

```

```

(IOFREE
 [LAMBDA (E CTX L)
  (SUBSET (FREEV E NIL L)
          (FUNCTION (LAMBDA (X)
                    (NOT (VRO X CTX]))

```

```

(OMCALL
 [LAMBDA (L)
  (PROG ((FN (CAR L))
        (ARGS (CDR L))
        D)
        (SETQ D (MD FN))
        (RETURN (COND
                 [(EQCAR D (QUOTE FNMODE))
                  (OR (NULL (CDR D))
                      (EVERY2 ARGS (CDDR D)
                                   (FUNCTION (LAMBDA (X Y)
                                             (EQ (MD X)
                                                  Y))
                  ((EQ D MODEMODE)
                   (OMCONSTX FN ARGS))
                  (T (SCANERROR FN (QUOTE "improperly used as function"))

```

```
(EVERY2
 [LAMBDA (EVL1 EVL2 EVFN)
 (EVERY EVL1 (FUNCTION (LAMBDA (EVX)
 (PROG1 (APPLY* EVFN EVX (CAR EVL2))
 (SETQ EVL2 (CDR EVL2]))
```

```
(MKCALL
 [LAMBDA (L)
 (PROG ((D (CAR L)))
 [COND
 ((EQCAR D (QUOTE PROCDEF))
 (COND
 ((EQCAR (SETQ D (EVAL D))
 (QUOTE FUNCTION))
 (PARMATCH D (CDR L)
 (CADADR (CAR L]
 (RETURN (COND
 ((EQ (MD D)
 MODEMODE)
 (MKCONSTX D (CDR L)))
 ((NEQCAR D (QUOTE FUNCTION))
 (MK1 (QUOTE CALL)
 L))
 ((EQ (CADDR D)
 0)
 (* D is (FUNCTION vars exp flag)
 (PARSUB D (CDR L)
 (CAR L)))
 (T (MK1 (QUOTE CALL)
 (CONSN L D (CDR L])
```

```
(MODECALL
 [LAMBDA (L)
 (CADR (MD (CAR L])
```

```
(SFSUBP
 [LAMBDA (E)
 (AND (LISTP E)
 (OR (SFDEF (CAR E))
 (AND (EQ (CAR E)
 (QUOTE CALL))
 (EQCAR (CADR E)
 (QUOTE FUNCTION))))
 (SOME (CDR E)
 (FUNCTION SFSUBP]))
```

```
(SFSUB
 [LAMBDA (E ALL)
 (SFSUB1 E)]
```

```
(SFSUB1
 [LAMBDA (E)
  (COND
   ((NLISTP E)
    E)
   ((AND (EQ (CAR E)
             (QUOTE CALL))
          (EQCAR (CADR E)
                 (QUOTE FUNCTION))))
   (PARSUB (CADR E)
            (CDDR E)
            (CADR E)))
  (T (PROG [(D (SFDEF (CAR E)
                    (RETURN (COND
                             ((AND D (OR ALL (CADDR D)))
                              (SFCALL (CAR E)
                                       (CDR E)))
                             (T (MKN E (MAPCARN (CDR E)
                                                (FUNCTION SFSUB1)))]
```

```
(FNSUBP
 [LAMBDA (E PFLAG)
  (FNSUBP1 E)]
```

```
(FNSUBP1
 [LAMBDA (E)
  (AND (LISTP E)
        (OR [COND
              (PFLAG (FNVAL (CAR E)))
              (T (CAR (FNVAL (CAR E)
                            (SOME (CDR E)
                                   (FUNCTION FNSUBP1)))]
```

```
(FNSUB
 [LAMBDA (E CL PFLAG)
  (PROG ((COND
         T))
        (RETURN (FNSUB1 E)]
```



```

(FNSUB1
 [LAMBDA (E)
  (COND
   ((NLISTP E)
    E)
   (T
    (SELECTQ
     (CAR E)
     (QUOTE E)
     [(FA EX FU CHOICE FUNCTION)
      (PROG ((BVMODES (ADDBV (CADR E)
                           BVMODES))
             (COND
              (COND
               P)
              (RETURN (MKN E (CONSN (CDR E)
                                   (CADR E)
                                   (CONSN (CDDR E)
                                       (SETQ P (FNSUB1 (CADDR E)))
                                       (COND
                                        ((NEQ (CAR E)
                                         (QUOTE CHOICE))
                                         (SETQ COND
                                          (MKAND (LIST COND P)))
                                          (CONSN (CDDDR E)
                                              (FNSUB1 (CADDR
                                                       E)
                                                       E]
                                          (FNVALUE (MKN E (MAPCARN (CDR E)
                                                                (FUNCTION FNSUB1)))
                                                    PFLAG NIL CL COND])
    ]

```

```

(XLCALL
 [LAMBDA (E)
  (PROG ((E1 (XLIS E))
        V)
   (COND
    ((NEQCAR (CADR E1)
             (QUOTE FUNCTION))
     (RETURN E1))
    ((NEQ (SETQ V (FNVALUE E1 T))
          E1)
     (RETURN V)))
   (SETQ V (PARSUB (CADR E1)
                  (CDDR E1)
                  (CADR E)))
   (AND (NULL XLBV)
        (ADDFNVALUE E1 V))
   (RETURN V])

```

```

(PROCDEF
 [LAMBDA (FN)
  [COND
   ((NEQ (TBTYPE FN)
    (QUOTE PROCEDURE))
    (HELP FN (QUOTE "is not a defined PROCEDURE"))
  (PROG ((IN (PROCINDECS FN))
   (OUT (PROCOUTDECS FN))
   (VL (CDR (SEXP FN)))
   (BVMODES BVMODES)
   BV DEF)
  (SETQ BVMODES (ADDBV (SETQ BV (MK1 (QUOTE STRUCMODE)
   (APPEND (CDR VL)
    NIL))))
   BVMODES))

 [SETQ DEF
  (MK1*
   (QUOTE FUNCTION)
   BV
   (RESTRICT
    [COND
     (OUT (PROG ((G (GENFROM FN))
      (M (CAR VL))
      (BVMODES BVMODES)
      RL)
      (SETQ BVMODES
       (ADDBV [SETQ RL
        (MKBV (LIST (MK1* (QUOTE VAR)
         G M])
         BVMODES))
       (RETURN (MAKECHOICE
        RL
        (SUBIODEC (LIST (CONS (RESULTNAME FN)
         G))
         OUT FN)
        G]
        (T (CADDR (SEXP (TLAST FN)
         (SUBIODEC NIL IN FN)
         (SETCVAL FN DEF (ROOTCTX))
         (SETVMODE FN (MD DEF)
         (ROOTCTX))
         (RETURN DEF]))

 (RESULTNAME
 [LAMBDA (FN)
  (QUOTE RESULT)])

 (MODEPROCDEF
 [LAMBDA (X)
  (PROCMODE (CADR X])

```

```
(PROCMODE
[LAMBDA (FN)
  (PROG ((M (VMODE FN)))
    (COND
      ((NEQUAL M (QUOTE (FNMODE)))
        (RETURN M)))
      (SETQ M (CDR (SEXP FN)))
      (SETVMODE FN [SETQ M (MK1 (QUOTE FNMODE)
                                (CONS (CAR M)
                                      (MAPCAR (CDR M)
                                             (FUNCTION CADDR)
                                             (ROOTCTX)))
                                (RETURN M]))
```

```
(SUBIODEC
[LAMBDA (AL L P)
  ([LAMBDA (EVALARGS)
    (MK (QUOTE AND)
      (MAPCAR L (FUNCTION (LAMBDA (A)
                          (PROG [(E (CADR (SEXP A)]
                                [BINDCTX (ROOTC P)
                                  (COND
                                    ((SFSUBP E)
                                      (SETQ E (SFSUB E NIL])
                                    (RETURN (FSUBMAKE AL E]
      NIL]))
```

```
(SFCALL
[LAMBDA (FN ARGS)
  (PARSUB (OR (CVAL FN (ROOTCTX))
            (HELP))
    ARGS FN])
```

```
(PARSUB
[LAMBDA (DEF ARGS FN)
  (FSUBMAKE (PARMATCH DEF ARGS FN)
    (CADDR DEF])
```

```

(PARMATCH
[LAMBDA (DEF ARGS FN)
  (PROG ((A ARGS)
        (V (CDADR DEF))
        L)
    L1 [COND
      [(NLISTP V)
       (COND
        ((NLISTP A)
         (RETURN (DREVERSE L)))
        (T (SCANERROR (QUOTE Too% many% args% for)
                      FN])
         ((NLISTP A)
          (SCANERROR (QUOTE Too% few% args% for)
                    FN))
        (T (MODEMATCH FN (CAR V)
                  (MD (CAR A)
                      (SETQ L (CONS (CONS (CADAR V)
                                       (CAR A))
                                   L))
                  (SETQ V (CDR V))
                  (SETQ A (CDR A))
                  (GO L1])

```

```

(MODEMATCH
[LAMBDA (FN VAR M)

```

(\* Should also try for a match  
between virtual and actual  
modes)

```

  (OR (MODEMATCH1 M (CADDR VAR))
      (SCANERROR (CADDR VAR)
                  (CONCAT (QUOTE "arg has wrong mode in call of ")
                          FN])

```

```

(MODEMATCH1
[LAMBDA (M M1)
  (OR (EQ M M1)
      (EQ M1 ANYMODE)
      (SELECTQ (CAR M)
                [(SEQMODE PTRMODE)
                 (AND (EQ (CAR M1)
                          (CAR M))
                     (MODEMATCH1 (CADR M)
                                   (CADR M1))
                NIL])

```

```

(PROCINDECS
[LAMBDA (FN)
  (PROG (L (A FN))
    LP (SELECTQ [CAR (SEXP (SETQ A (TSUCC A)
                          ((LET (DECVAR START))
                           ((ASSERT DECLARE)
                            (SETQ L (CONS A L)))
                           (RETURN (DREVERSE L)))
                        (GO LP])

```

(PROCOUTDECS

```

[LAMBDA (FN ALL)
  (PROG (L (A (TLAST FN)))
    LP (SELECTQ (CAR (SEXP A))
      (STOP)
      [(ASSERT DECLARE)
        (COND
          [(OR ALL (NULL (IOFREE (CADR (SEXP A))
            NIL
            (LIST (RESULTNAME FN)
              (SETQ L (CONS A L])
              (RETURN L))
            (SETQ A (TPRED A))
            (GO LP])

```

(FNVALUE

```

[LAMBDA (E PFLAG CTX CL COND)
  (COND
    ((NLISTP E)
      E)
    (T
      (PROG ((Z (FNVAL (CAR E)
        CTX)))
        (RETURN
          (COND
            [Z
              (PROG ([FL (AND BVMODES (MAPCAR BVMODES
                (FUNCTION CAR]
                R D AL HH)
                (COND
                  ((NULL COND)
                    (SETQ COND T)))
                (RETURN
                  (COND
                    ((AND (CAR Z)
                      (SETQ D (FASSOC E (CAR Z)))
                      (NOT (OCCURSFREE FL (CADR D)))
                      (FNHISTP (CDDR D)
                        CL CTX))
                    (FNVRET (CADR D)
                      (CDDR D)))
                    ((AND
                      (CDR Z)
                      PFLAG
                      [SETQ D
                        (MAPCONC
                          (CDR Z)
                          (FUNCTION (LAMBDA (Y)
                            (AND
                              (FNHISTP (FVHIST Y)
                                CL CTX)
                              [SETQ AL
                                (QMATCH (FVPAT Y)
                                  E
                                  (QUOTE (((S1 . NOBIND)
                                    (S2 . NOBIND)
                                    (S3 . NOBIND)
                                    (S4 . NOBIND)
                                    (S5 . NOBIND))

```

0]

```

[NOT (OCCURSFREE FL (SETQ D
(FVCOND Y)
[SETQ D
(CTXEV
(FSUBMAKE
(SETQ AL (CAAR AL))
(COND
(R (MKOR (FRPLACA R D)))
[(NEQ COND T)
(MKOR (SETQ R
(LIST D COND]
(T D]
(NOT (OCCURSFREE FL (FVVAL Y)))
(PROG [(V (FSUBMAKE AL
(FVVAL Y)
(RETURN
(COND
[(EQ D T)
(RTFRM
FNVALUE
(FNVRET V
(FVHIST
Y]
(T (SETQ HH
(APP2 HH
(FVHIST Y)))
(LIST D V]
(NEQ PFLAG (QUOTE NOCOND)))
(FNVRET (MKCOND (NCONC D (LIST T E)))
HH))

```

(T E]

(T E])

```

(FNVRET
[LAMBDA (VAL H)
VAL])

```

(\* For tracing)

```

(FNHISTP
[LAMBDA (L CL CTX)
(EVERY L (FUNCTION (LAMBDA (X)
(AND (NOT (FMEMB X CL))
(THCLAUSEP X CTX]))

```

```

(ADDFNVALUE
[LAMBDA (E VAL H CTX)
(PROG ((Z (FNVAL (CAR E)
CTX)))
(SETFNVAL (CAR E)
(CONS (CONS (CONS E (CONS VAL H))
(CAR Z))
(CDR Z))
CTX)
(RETURN VAL])

```

```

(ADDFNPATTERN
[LAMBDA (E VARS COND VAL H CTX)
  (PROG [(Z (FNVAL (CAR E)
                  CTX))
        (AL (MAP2CAR VARS (QUOTE ($1 $2 $3 $4 $5))
                        (FUNCTION CONS]
        (SETFNVAL (CAR E)
                  (CONS (CAR Z)
                        (CONS (CONS (GVSUBST AL E)
                                   (CONS (GVSUBST AL COND)
                                         (CONS (GVSUBST AL VAL)
                                               H))))
                        (CDR Z)))
                  CTX)
        (RETURN VAL)])
)
(DEFLIST(QUOTE(
(MAKEFUNCTION (25-AUG-74 . 2206))
(MODEFUNCTION (26-AUG-74 . 715))
(XLFUNCTION (27-AUG-74 . 746))
(ADDPROC (25-AUG-74 . 2206))
(RESETPROC (26-AUG-74 . 1431))
(DELETEPROC (26-AUG-74 . 1431))
(IOACHECK (27-AUG-74 . 928))
(IOACK (4-JUN-74 . 1015))
(IOFREE (4-JUN-74 . 1015))
(OMCALL (27-AUG-74 . 1042))
(MKCALL (27-AUG-74 . 1052))
(MODECALL (27-AUG-74 . 953))
(SFSUBP (27-AUG-74 . 2141))
(SFSUB1 (27-AUG-74 . 2141))
(FNSUBP (27-MAY-74 . 2254))
(FNSUBP1 (27-MAY-74 . 2254))
(FNSUB (15-JUN-74 . 56))
(FNSUB1 (12-AUG-74 . 956))
(XLCALL (25-AUG-74 . 2225))
(PROCDEF (27-AUG-74 . 2046))
(RESULTNAME (26-AUG-74 . 1432))
(MODEPROCDEF (27-AUG-74 . 713))
(PROCMODE (27-AUG-74 . 714))
(SFCALL (25-AUG-74 . 2230))
(PARSUB (25-AUG-74 . 2230))
(PARMATCH (25-AUG-74 . 2231))
(MODEMATCH (4-JUN-74 . 1036))
(MODEMATCH1 (4-JUN-74 . 1035))
(PROCOUTDECS (26-AUG-74 . 1525))
(FNVALUE (13-AUG-74 . 1035))
(FNVRET (27-MAY-74 . 1021))
(FNHIISTP (4-JUN-74 . 1332))
(ADDFNVALUE (26-MAY-74 . 2253))
(ADDFNPATTERN (26-MAY-74 . 2254))
)))(QUOTE EDITDATE))

```

```
[DECLARE: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY
  (BLOCK: NIL MAKEFUNCTION MODEFUNCTION XLFUNCTION ADDPROC RESETPROC DELETEPROC
    IOACHECK IOACK IOFREE OMCALL EVERY2 MKCALL MODECALL XLCALL PROCDEF
    RESULTNAME MODEPROCDEF PROCMODE SUBIODEC SFCALL PARSUB PARMATCH
    MODEMATCH MODEMATCH1 PROCINDECS PROCOUTDECS FNVALUE FNVRET FNHISTP
    ADDFNVALUE ADDFNATTERN)
  (BLOCK: SFSUBP)
  (BLOCK: SFSUB SFSUB SFSUB1 (LOCALFREEVARS ALL))
  (BLOCK: FNSUBP FNSUBP FNSUBP1 (LOCALFREEVARS PFLAG))
  (BLOCK: FNSUB FNSUB FNSUB1 (LOCALFREEVARS CL COND PFLAG))
]
(DECLARE: DONTCOPY
  (FILEMAP (NIL (1584 12864 (MAKEFUNCTION 1596 . 1666) (MODEFUNCTION 1670 .
1838) (XLFUNCTION 1842 . 2133) (ADDPROC 2137 . 2186) (RESETPROC 2190 . 2482)
(DELETEPROC 2486 . 2635) (IOACHECK 2639 . 3282) (IOACK 3286 . 3509) (IOFREE
3513 . 3619) (OMCALL 3623 . 4020) (EVERY2 4024 . 4177) (MKCALL 4181 . 4746)
(MODECALL 4750 . 4798) (SFSUBP 4802 . 5005) (SFSUB 5009 . 5051) (SFSUB1 5055
. 5439) (FNSUBP 5443 . 5489) (FNSUBP1 5493 . 5661) (FNSUB 5665 . 5755) (FNSUB1
5759 . 6393) (XLCALL 6397 . 6757) (PROCDEF 6761 . 7778) (RESULTNAME 7782 .
7830) (MODEPROCDEF 7834 . 7886) (PROCMODE 7890 . 8218) (SUBIODEC 8222 . 8505)
(SFCALL 8509 . 8601) (PARSUB 8605 . 8696) (PARMATCH 8700 . 9204) (MODEMATCH
9208 . 9559) (MODEMATCH1 9563 . 9766) (PROCINDECS 9770 . 9997) (PROCOUTDECS
10001 . 10340) (FNVALUE 10344 . 12043) (FNVRET 12047 . 12130) (FNHISTP 12134
. 12259) (ADDFNVALUE 12263 . 12480) (ADDFNPATTERN 12484 . 12861))))))
STOP
```



(FILECREATED "24-AUG-74 22:06:32" VQC.;2 8791

changes to: MKLCOND MKCOND TVP CONDIR CONDIT1 MAKECHOICE MAKEFA MAKEEX  
MAKEFU MKBV QU QUANT NESUBL VARSUBL QUSUB ADDBV BVREM1 DOMPRED DOMSATL DOMSTEP  
VQCCOMS VQCBLOCKS

previous date: "24-AUG-74 21:47:26" VQC.;1)

(LISPXPRT (QUOTE VQCCOMS)

((ADDVARS (SUPEROPS  
(LCOND . CONDFIX)  
(CHOICE . CHOICEFIX)))

→ (RPAQQ VQCCOMS ((PROP (OPMAKE NOTOP) FA EX FU LCOND DCOND CHOICE BV) OPMODE MCFN

→ (DECLARE: DOEVAL@COMPILE COMPILERVARS (ADDVARS (NOLINKFNS QUSUB)))  
(ENDUMP VQC))

(DEFLIST(QUOTE(  
(FA (MAKEFA 3)) ((RESTRICT . RESQUANT))  
(EX (MAKEEX 3)) ((RESTRICT . RESFU))  
(FU (MAKEFU 3))  
(LCOND (MKLCOND ARB 3)) ((LCOND) (RESTRICT . RESLCOND))  
(DCOND (MKCOND ARB 3)) ((LCOND) (RESTRICT . RESCOND))  
(CHOICE (MAKECHOICE 3 (CHOICE)))  
(BV (MKBV ARB))  
)) (QUOTE OPMAKE))  
((LCOND) (CHOICE))

(DEFLIST(QUOTE(  
(FA (NOTQUANT EX))  
(EX (NOTQUANT FA))  
(FU (NOTQUANT FU))  
(LCOND NIL)  
(DCOND NIL)  
(CHOICE NIL)  
(BV NIL)

→ Stuff from VQC

)) (QUOTE NOTOP))  
[DECLARE: DOEVAL@COMPILE COMPILERVARS  
(ADDTOVAR NOLINKFNS QUSUB)

] (RPAQQ VQCBLOCKFNS (QU))  
(RPAQQ VQCFNS  
(MKLCOND MKCOND TVP CONDIR CONDIT1 MAKECHOICE MAKEFA MAKEEX MAKEFU  
MKBV QU QUANT NESUBL VARSUBL QUSUB ADDBV BVREM1 DOMPRED  
DOMSATL DOMSTEP))

[RPAQQ VQCBLOCKS ((QUANT QUANT BVREM1 DOMPRED DOMSATL DOMSTEP (SPECVARS  
V BVMODES)

(ADDTOVAR BLKLIBRARY QU)  
[DECLARE: DOEVAL@COMPILE  
(DEFLIST(QUOTE(  
[QU (LAMBDA (CJ)  
(COND (CJ (QUOTE FA))  
(T (QUOTE EX]

)) (QUOTE BLKLIBRARYDEF))

] (DEFINEQ



```

(CONDIT1
 [LAMBDA (C V R)
  (PROG (D)
   L2 (RETURN (COND
    ((CONSTP C)
     (COND
      ((CONSTVAL C)
       (RTFRM CONDIT V))
      (T R)))
     ((EQCAR V (QUOTE LCOND))
      (SETQ D (LIST T C))
      (MAP (CDR V)
       [FUNCTION (LAMBDA (X)
        (SETQ R (CONDIT1 (MK (QUOTE AND)
         (FRPLACA D (CAR X)))
         (CADR X)
         R])
        (FUNCTION CDDR))
       R]
      ((SETQ D (FASSOC V R))
       (SETQ R (DREMOVE D R))
       (SETQ C (MK* (QUOTE OR)
        (CDR D)
        C))
       (GO L2))
      (T (CONS (CONS V C)
       R]))
  (MAKECHOICE
   [LAMBDA (VL E F)
    (PROG ((S (NESUBL (CDR VL)
     E NIL)))
     (RETURN (COND
      (S (FSSUB (CDR S)
       (CAR S)
       F))
      (T (MK1* (COND
       ([[LAMBDA (BVMODES)
        (TVP F]
        (ADDBV VL BVMODES))
        (QUOTE FU))
        (T (QUOTE CHOICE)))
        VL E F))
  (MAKEFA
   [LAMBDA (VL R E)
    (QUANT VL R E T BVMODES)]
  (MAKEEX
   [LAMBDA (VL R E)
    (QUANT VL R E NIL BVMODES)]

```

OMCOND } from VTS  
MODECOND }  
CONDFIX } from VEU  
CONDFIXI }  
RESLCOND  
RESCOND  
RESCOND1

OMCHOICE } from VTS  
MODECHOICE }  
CHOICEFIX } from VEU

```

(MAKECHOICE
 [LAMBDA (VL E F)
  (PROG ((S (NESUBL (CDR VL)
   E NIL)))
   (RETURN (COND
    (S (FSSUB (CDR S)
     (CAR S)
     F))
    (T (MK1* (COND
     ([[LAMBDA (BVMODES)
      (TVP F]
      (ADDBV VL BVMODES))
      (QUOTE FU))
      (T (QUOTE CHOICE)))
      VL E F))
  (MAKEFA
   [LAMBDA (VL R E)
    (QUANT VL R E T BVMODES)]
  (MAKEEX
   [LAMBDA (VL R E)
    (QUANT VL R E NIL BVMODES)]

```

```
(MAKEFU
 [LAMBDA (VL P Q)
 (COND
 ((NULL (CDR VL))
 (MKAND (LIST P Q)))
 (T (MK1* (QUOTE FU)
 VL P Q))
```

```
(MKBV
 [LAMBDA (L)
 (MK1 (QUOTE BV)
 (COND
 [(CDR L)
 (SORTCOPY L (FUNCTION (LAMBDA (X Y)
 (ALPHORDER (CADR X)
 (CADR Y)
 (T L]))
```

```
(QU
 [LAMBDA (CJ)
 (COND
 (CJ (QUOTE FA))
 (T (QUOTE EX]))
```

```
(QUANT
 [LAMBDA (VL R E CJ OLDB)
 (PROG [D L D1 L1 VARS (BVMODES OLDB)
 (Q (QU CJ))
 (NCJ (NOT CJ))
 (OP (BC CJ))
 (NOP (BC (NOT CJ))
 L0 (SETQ BVMODES (ADDBV VL OLDB))
 (SETQ VARS (BVL VL))
 L1 (SETQ E (CONVNF E CJ))
 L2 (SETQ R (CONVNF R NIL))
 (RETURN
 (COND
 ((NULL VARS)
 (BOOLSEQ (LIST (BXOR R CJ)
 E)
 NCJ))
 ((NULL R)
 CJ)
 ((EQCAR E Q)
 [SETQ VARS (BVL (SETQ VL (MKBV (APPEND (CDR VL)
 (CDADR E)
 NIL]
 (SETQ BVMODES (ADDBV (CADR E)
 BVMODES))
 [SETQ R (MKAND (LIST R (CADR E)
 (SETQ E (CADDDR E))
 (GO L2))
```



(MK\* OP (QUANT (MKBV (BVREM1 (CAR L)  
(CDR VL))))

R  
(QUSUB L E)  
CJ OLDB)

(QUANT VL R (BOOLSEQ  
(CONS (BOOLSEQ (REMV1 (CAR L1)  
(CDAR D1))

CJ)  
(REMV1 (CAR D1)  
(CDR E)))

NCJ)  
CJ OLDB)))

(T (GO LX]  
((DOMPRED VARS E CJ)  
[SETQ R (MKAND (LIST R (BXOR E CJ]  
(SETQ E NCJ)  
(GO L1))  
((OCCURSFREE VARS E)  
(GO LX))

~~(CONSTP E)  
(COND  
((EQ E CJ)  
E)  
(T (BXOR (DOMSATL VARS R)  
CJ]  
((OCCURSFREE VARS R)  
(GO LX))  
(T L)))~~

((OCCURSFREE VARS  
(SETQ D (DOMSATL VARS R)))  
(COND  
((NEG R (SETQ R D)) (GO L2))  
(T (GO LX))))  
(T (BOOLSEQ (LIST (BXOR D CJ) E)  
NCJ))

LE (SETQ R (QUSUB L R))  
LE1 [SETQ VL (MKBV (BVREM1 (CAR L)  
(CDR VL]

(SETQ E (QUSUB L E))  
(GO L0)

LX (RETURN (MK1\* Q VL R E])

(NESUBL  
[LAMBDA (L X CJ)  
(AND (EQCAR X (EF (NOT CJ)))  
(VARSUBL L X])

(VARSUBL  
[LAMBDA (L X)  
(PROG ((Z L))  
LP (RETURN (AND Z (OR (VARSUBP (CAR Z)  
X)  
(PROGN (SETQ Z (CDR Z))  
(GO LP]))

(QUSUB  
[LAMBDA (D E)  
(FSUBMAKE (LIST D)  
E])

```
(ADDBV
 [LAMBDA (BV OLD)
 [MAPC (CDR BV)
 (FUNCTION (LAMBDA (X)
 (SETQ OLD (CONS (CDR X) OLD])
 OLD])])
```

```
(BVREM1
 [LAMBDA (V VL)
 (COND
 ((EQ V (CADAR VL))
 (CDR VL))
 (T (CONS (CAR VL)
 (BVREM1 V (CDR VL))
```

```
(DOMPRED
 [LAMBDA (V X CJ)
 (COND
 ([SELECTQ (CAR X)
 ((LEQ GEQ)
 T)
 (EQ (CAR X)
 (EF (NOT CJ))
 (COND
 ((LISTP V)
 (VARSUBL V X))
 (V (VARSUBP V X))
```

```
(DOMSATL
 [LAMBDA (VARS P)
 (PROG ((L VARS)
 (R P))
 LP (COND
 ((NULL L))
 (RETURN R))
 ((NILISTP (SETQ R (DOMSTEP (CAR L)
 R)))
 (RETURN R))
 (TV (SETQ L (CDR L))
 (GO LP])
```

*(OR \* (NEQCAR R 'AND))*

←PP RESQUANT

(RESQUANT

[LAMBDA (OP L Z)

(PROG ((BY (CAR L))

B M E P)

[SETQ B (LAMBDA (BYMODES)

(COND

[(EQ OP (QUOTE FA))

(MKOR (CONS (MAKENOT (CADR L))

(CDDR L])

(T (MKAND (CDR L])

(SETQ M (ADDBY BY BYMODES])

(RETURN

(COND

((NEOCAR B (QUOTE RESTRICT))

(MK♦ OP BY T B))

(T

(RESTRICT

(MK♦ OP BY T (SETQ E (CADR B)))

(MKOR (LIST (MAKEFA BY T (SETQ P (CADDR B)))

(MAKEEX BY T

(LAMBDA (BYMODES)

(MKAND (LIST P (MAKENOT

B])

M])

RESQUANT

←



(DOMSTEP  
[LAMBDA (V R)

(\* R IS A DOMPRED OR A  
CONJUNCTION THEREOF)

(COND  
((DOMPRED V R)  
T)

(T (PROG (M (DISTRIB (CDR R) (M R) X P  
(FUNCTION [LAMBDA (X) NON  
(RPARITY V X])  
(COPY (QUOTE ((1)  
(-1])

POS NEG)

[SETQ POS (MAPRPC (CDR M)  
(FUNCTION (LAMBDA (X)  
(CDR (VARSUBP V X))  
[SETQ NEG (MAPRPC (CDADR M)  
(FUNCTION (LAMBDA (X)  
(CDR (VARSUBP V X))

(COND  
((OR (CADDR M)  
(CAADDR M))  
(HELP)))

(RETURN (MKAND (NCONC (CADDDR M) NON  
(MAPROD POS NEG (FUNCTION MAKELEQ)))

LP  
[COND ((SETQ M (CDR M))  
[SELECTQ [SETQ P  
(RPARITY V  
(SETQ X (CDR M))  
[1 (SETQ POS (CONS  
(CDR (VARSUBP V X)) POS]  
[-1 (SETQ NEG (CONS  
(CDR (VARSUBP V X)) NEG]  
[NIL (SETQ NON (CONS X NON)]  
(HELP)] (GO LP]

(DEFLIST(QUOTE(  
(MAKECHOICE (19-JUN-74 . 114))  
(QUANT (24-AUG-74 . 2152))  
(DOMPRED (2-AUG-74 . 2319))  
))(QUOTE EDITDATE))

[DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY  
(BLOCK: NIL MKLCOND MKCOND TVP CONDIT CONDIT1 MAKECHOICE MAKEFA MAKEEX MAKEFU  
MKBV QU NESUBL VARSUBL QUSUB ADDBV)  
(BLOCK: QUANT QUANT BVREM1 DOMPRED DOMSATL DOMSTEP (SPECVARS V BVMODES))

(LISPXPRT (QUOTE VQCCOMS)  
T)

(RPAQQ VQCCOMS ((PROP (OPMAKE NOTOP)  
FA EX FU LCOND DCOND CHOICE BV)  
(DECLARE: DOEVAL@COMPILE COMPILERVARS (ADDVARS (NOLINKFNS QUSUB)))  
(ENDUMP VQC)))

(DECLARE: DONTCOPY  
(FILEMAP (NIL (1392 8176 (MKLCOND 1404 . 1446) (MKCOND 1450 . 1493) (TVP 1497  
. 1545) (CONDIT 1549 . 2287) (CONDIT1 2291 . 2877) (MAKECHOICE 2881 . 3210)  
(MAKEFA 3214 . 3272) (MAKEEX 3276 . 3336) (MAKEFU 3340 . 3469) (MKBV 3473  
. 3639) (QU 3643 . 3718) (QUANT 3722 . 6505) (NESUBL 6509 . 6588) (VARSUBL  
6592 . 6748) (QUSUB 6752 . 6811) (ADDBV 6815 . 6936) (BVREM1 6940 . 7068)  
(DOMPRED 7072 . 7259) (DOMSATL 7263 . 7497) (DOMSTEP 7501 . 8173))))))  
STOP

OM QUANT  
from VTS

RESFU

Redone, see insert

XEROX

XEROX

E A R S

Filename: <DEUTSCH>VREL.;50

Creation Date: SUNDAY, 23 MAR 1975 22:40-PDT

Printed by: DEUTSCH

XEROX

XEROX

(FILECREATED "23-MAR-75 22:32:40" &lt;DEUTSCH&gt;VREL.;50 13137

previous date: " 1-SEP-74 02:05:46" &lt;DEUTSCH&gt;VREL.;49)

```

(LISPXPRT (QUOTE VRELCOMS)
  T T)
(RPAQQ VRELCOMS ((PROP (OPMAKE NOTOP OPMODE MCFN)
  GEQ LEQ EQUAL NEQUAL)
  (PROP EQUALPRED EQUAL)
  (ENDUMP VREL)))
(DEFLIST(QUOTE(
  (GEQ (MAKEGEQ 2))
  (LEQ (MAKELEQ 2))
  (EQUAL (MAKEEQUAL 2))
  (NEQUAL (MAKENEQUAL 2))
  ))(QUOTE OPMAKE))
(DEFLIST(QUOTE(
  (GEQ (NOTREQ (LEQ . -1)))
  (LEQ (NOTREQ (GEQ . 1)))
  (EQUAL NEQUAL)
  (NEQUAL EQUAL)
  ))(QUOTE NOTOP))
(DEFLIST(QUOTE(
  (GEQ BOOL)
  (LEQ BOOL)
  (EQUAL BOOL)
  (NEQUAL BOOL)
  ))(QUOTE OPMODE))
(DEFLIST(QUOTE(
  (GEQ (INT INT))
  (LEQ (INT INT))
  (EQUAL (INT INT))
  (NEQUAL (INT INT))
  ))(QUOTE MCFN))
(DEFLIST(QUOTE(
  (EQUAL T)
  ))(QUOTE EQUALPRED))
(RPAQQ VRELBLKFNS (EF))
(RPAQQ VRELFNS
  (RPARITY VPARITY LINVARS LINEAR MAKEREL RELQUOP RELQUO1 ALTREL ALTREL1
  EPRODREL RELDIV EEXPTREL EXPTR1 MAKEGEQ MAKELEQ MAKEQUAL
  MAKENEQUAL EF CJREL ANDREL ORREL CJREL1 ENUMREL VARTERMS
  VARTERMS1 VTERMP RELTERMS VARSUBP))
(RPAQQ VRELVAR ((ENUMMAX 5)))
(RPAQQ VRELGLOBS (ENUMMAX))
(RPAQQ VRELBLOCKS ((MAKEREL MAKEREL RELQUOP RELQUO1)
  (CJREL CJREL ANDREL ORREL CJREL1 ENUMREL)
  (VARTERMS VARTERMS VARTERMS1)))
[DECLARE: DOEVAL@COMPILE
  (ADDTVAR BLKLIBRARY EF)
]
[DECLARE: DOEVAL@COMPILE
(DEFLIST(QUOTE(
  [EF (LAMBDA (CJ)
    (COND (CJ (QUOTE EQUAL))
    (T (QUOTE NEQUAL]
  ))(QUOTE BLKLIBRARYDEF))
]
(DEFINEQ
(RPARITY
  [LAMBDA (X R)
    (PROG (S P)
      (SETQ S (SELECTQ (CAR R)
        (GEQ 1)
        (LEQ -1)
        (RETURN NIL)))
      (RETURN (AND (SETQ P (VPARITY X (CADR R)))
        (ITIMES P S])

```

```

(VPARITY
 [LAMBDA (X E)
 (COND
 ((EQ X E)
 1)
 ((LISTP E)
 (SELECTQ (CAR E)
 [PLUS (FIRSTV (CDDR E)
 (FUNCTION (LAMBDA (Z)
 (VPARITY X Z)
 [TIMES (COND
 ((AND (EQ X (CADDR E))
 (NULL (CDDDR E)))
 (CADR E]
 NIL])
 (LINVARS
 [LAMBDA (E L)
 (COND
 ((CONSTP E)
 L)
 ((ATOM E)
 (ADDELT E L))
 (T (SELECTQ (CAR E)
 (PLUS (MAPREDC (CDDR E)
 (FUNCTION LINVARS)
 L))
 (TIMES (COND
 ((NULL (CDDDR E))
 (LINVARS (CADDR E)
 L))
 (T L)))
 (ADDELMT E L])
 (LINEAR
 [LAMBDA (E PFN)
 (OR (APPLY* PFN E)
 (AND (LISTP E)
 (SELECTQ (CAR E)
 [PLUS (EVERY (CDDR E)
 (FUNCTION (LAMBDA (X)
 (LINEAR X PFN]
 [TIMES (AND (NULL (CDDDR E))
 (APPLY* PFN (CADDR E)
 NIL])
 (MAKEREL
 [LAMBDA (OP X Y)
 (PROG (W D L C U (E X)
 (INC 0))
 [COND
 ((CONSTP Y)
 (SETQ INC (CONSTVAL Y)))
 (T (SETQ E (MAKEDIFF X Y)
 TEST[COND
 ((CONSTP E)
 (RETURN (APPLY* OP (CONSTVAL E)
 INC]
 (COND
 ((SETQ W (RELQUOP E))
 (SETQ U (CAR W))
 (SETQ E (CDR W))
 (SETQ INC (TIMES U INC))

```

```

(SELECTQ OP
  (GEQ)
  (LEQ (SETQ INC (PLUS INC U -1)))
  [EQUAL (RETURN (MK*(QUOTE AND)
    (MAKEREL (QUOTE GEQ)
      E INC)
    (MAKEREL (QUOTE LEQ)
      E
      (PLUS INC U -1]
  [NEQUAL (RETURN (MK*(QUOTE OR)
    (MAKEREL (QUOTE LEQ)
      E
      (PLUS INC -1))
    (MAKEREL (QUOTE GEQ)
      E
      (PLUS INC U]
    (HELP))
  (GO TEST)))
(COND
  ((AND (EQCAR (SETQ W (FIRSTERM E))
    (QUOTE TIMES))
    (MINUSP (CADR W)))
    (SETQ E (NEGOF E))
    (SETQ INC (MINUS INC))
    (SETQ OP (SELECTQ OP
      (LEQ (QUOTE GEQ))
      (GEQ (QUOTE LEQ))
      OP))
    (GO TEST)))
[SETQ D (TERMGCD (SETQ L (GETTERMS E]
(SETQ C (DIFFERENCE INC (CONSTERM E)))
(SETQ L (LPLUS 0 L))
[COND
  ((NEQ D 1)
    (SETQ L (MAKECDIV L D))
    (COND
      ((ZEROP (REMAINDER C D))
        (SETQ C (QUOTIENT C D)))
      (T (SELECTQ OP
        [(EQUAL NEQUAL)
          (RETURN (EQ OP (QUOTE NEQUAL)
            (LEQ (SETQ INC (FLOORQUOTIENT C D))
              (SETQ E L)
              (GO TEST))
            (GEQ (SETQ INC (MINUS (FLOORQUOTIENT (MINUS C)
              D)))
              (SETQ E L)
              (GO TEST))
          (HELP]

```

```

(RETURN (SELECTQ OP
  ((EQUAL NEQUAL)
   (SELECTQ (CAR (LISTP L))
    [(PLUS TIMES)
     (COND
      ((SETQ U (UNITFACTORS L))
       (EPRODREL OP L C U))
      (T (MK1* OP L C])
     (IEXPT (EEXPTREL OP L C))
     (MK1* OP L C)))
   (COND
    [(AND (OR (EQ C 0)
              (EQ C (SELECTQ OP
                (GEQ 1)
                -1)))
          (SETQ U (SELECTQ (CAR (LISTP L))
            ((PLUS TIMES)
             (UNITFACTORS L))
            NIL)))
     (SETQ U (RELDIV L U))
     (SELECTQ C
      (1 (* OP is GEQ)
        (ALTREL U (QUOTE GEQ)
          1
          (QUOTE LEQ)
          -1 T))
      (-1 (* OP is LEQ)
        (ALTREL U (QUOTE GEQ)
          1
          (QUOTE LEQ)
          -1 NIL))
      (ALTREL U (QUOTE GEQ)
        0
        (QUOTE LEQ)
        0
        (EQ OP (QUOTE GEQ]
      (T (MK1* OP L C])

```

```

(RELQUOP
 [LAMBDA (E)
  (AND
   (LISTP E)
   (SELECTQ
    (CAR E)
    ((QUOTIENT TIMES)
     (RELQUO1 E))
   [PLUS
    (PROG ((L (CDDR E))
           X R Q)
      L1 [COND
        ((AND (LISTP (SETQ X (CAR L)))
              (SETQ R (RELQUO1 X)))
         (COND
          (Q (RETURN NIL))
          (T (SETQ Q X]
        (COND
         ((SETQ L (CDR L))
          (GO L1))
         ((NULL Q)
          (RETURN NIL)))
        (RETURN
         (FRPLACD
          R
          (MK (QUOTE PLUS)
            (CONS (TIMES (CADR E)
                    (CAR R))
                (MAPCAR (CDDR E)
                    (FUNCTION (LAMBDA (X)
                        (COND
                         ((EQ X Q)
                          (CDR R))
                         (T (MAKECMUL X (CAR R]

```

```

NIL])

```

```

(RELQUO1
  [LAMBDA (E)
    (SELECTQ
      (CAR E)
      [QUOTIENT (AND (CONSTP (CADDR E))
                     (CONS (CADDR E)
                           (CADR E))
                     [TIMES (AND (EQ (CADR E)
                                     -1)
                                 (NULL (CDDDR E))
                                 (EQCAR (CADDR E)
                                       (QUOTE QUOTIENT))
                                 (PROG ((A (CDADDR E)))
                                       (RETURN (AND (CONSTP (CADR A))
                                                  (CONS (CADR A)
                                                      (MAKEDIFF (SUB1 (CADR A))
                                                                (CAR A))
                                                  NIL]))
                                 NIL])
      (ALTREL
        [LAMBDA (L R1 C1 R2 C2 EVEN)
          (MK (QUOTE OR)
              (ALTREL1 L NIL EVEN])]
      (ALTREL1
        [LAMBDA (L L1 EVEN)
          (COND
            [(NULL L)
              (AND EVEN (LIST (MK (QUOTE AND)
                                L1)
                              (T ([LAMBDA (Z)
                                  (NCONC (ALTREL1 (CDR L)
                                                  Z EVEN)
                                          (ALTREL1 (CDR L)
                                                  (FRPLACA Z (MAKEREL R2 (CAR L)
                                                                C2))
                                          (NOT EVEN]
                                  (CONS (MAKEREL R1 (CAR L)
                                      C1)
                                      L1))]
            ]))
          (* OP is EQUAL or NEQUAL, U1 are
            the unit factors of L.)
          (* Note that U must have at
            least 2 elements.)
          (RETURN
            (COND
              [(ZEROP C)
                (MK BOP (MAPCAR U (FUNCTION (LAMBDA (X)
                                             (MK* OP X 0)
                                             (* Too hard, punt.))
                                  ((CDDR U)
                                   (MK1* OP L C))
                                  (T
                                    (PROG [(A (CAR U))
                                           (B (CADR U))
                                           (BOP1 (BC (EQ OP (QUOTE EQUAL)
                                                       (RETURN
                                                         (MK
                                                           BOP
                                                           (MAPALLFACTORS
                                                           (FACTORS C)
                                                           (FUNCTION (LAMBDA (X)
                                                                     (PROG [(Y (IQUOTIENT C X))
                                                                           (Y1 (COND
                                                                           ((MINUSP C)
                                                                           (IMINUS X))
                                                                           (T X]
                                                           (RETURN
                                                           (MK* BOP (MK* BOP1
                                                           (MK* OP A Y)
                                                           (MK* OP B Y1))
                                                           (MK* BOP1 (MK* OP A (IMINUS Y))
                                                           (MK* OP B (IMINUS Y1])
                ]))
            ]))
          ]))

```

```
(RELDIV
 [LAMBDA (E U)
 ([LAMBDA (U1)
 (COND
 ((EQ U1 1)
 U)
 (T (CONS U1 U]
 (UNITDIVIDE E U])
 (* U is (UNITFACTORS L))
```

```
(EEXPTREL
 [LAMBDA (OP L C)

 (* OP is EQUAL or NEQUAL, L is is
 (IEXPT A B).)
```

```
(PROG ((A (CADR L))
 (B (CADDR L))
 P)
 (RETURN (COND
 ((ZEROP C)
 (MK1* OP A 0))
 ((EQ OP (QUOTE NEQUAL))
 (MAKENOT (EEXPTREL (QUOTE EQUAL)
 L C)))
 ((EQ C 1)
 (SFCALL (QUOTE *EXPT1P)
 (CDR L)))
 [(SETQ P (POWERFACP (FACTORS C)))
 (MK (QUOTE OR)
 (MAPALLFACTORS (FACTORS (CDR P))
 (FUNCTION (LAMBDA (X)
 (EXPTREL1
 A B X
 (IEXPT (CAR P)
 (IQUOTIENT (CDR P)
 X]
 (T (EXPTREL1 A B 1 1]))
```

```
(EXPTREL1
 [LAMBDA (A B X Y)
 (MK*(QUOTE AND)
 (MAKEQUAL B X)
 (COND
 [(ZEROP (LOGAND X 1))
 (MK*(QUOTE OR)
 (MAKEQUAL A Y)
 (MAKEQUAL A (IMINUS Y]
 (T (MAKEQUAL A Y]))
```

```
(MAKEGEQ
 [LAMBDA (X Y)
 (MAKEREL (QUOTE GEQ)
 X Y])
```

```
(MAKELEQ
 [LAMBDA (X Y)
 (MAKEREL (QUOTE LEQ)
 X Y])
```

```
(MAKEQUAL
 [LAMBDA (X Y)
 (MAKEREL (QUOTE EQUAL)
 X Y])
```

```
(MAKENEQUAL
 [LAMBDA (X Y)
 (MAKEREL (QUOTE NEQUAL)
 X Y])
```



```

(EF
  [LAMBDA (CJ)
    (COND
      (CJ (QUOTE EQUAL))
      (T (QUOTE NEQUAL)))

(CJREL
  [LAMBDA (L CJ)
    (COND
      (CJ (ANDREL L))
      (T (ORREL L]))

(ANDREL
  [LAMBDA (L)
    (PROG (LB UB LR UR NE D R X (M L)
          (Y (CADAR L))))
    LO [COND
      (M [SETQ X (CADDR (SETQ R (CAR M)))
          (SETQ M (CDR M))
          (SELECTQ (CAR R)
            [LEQ (OR (AND UB (ILESSP UB X))
                    (PROGN (SETQ UR R)
                          (SETQ UB X))
            [GEQ (OR (AND LB (IGREATERP LB X))
                    (PROGN (SETQ LR R)
                          (SETQ LB X))
            [EQUAL (RETURN
                  (COND
                    ([EVERY L (FUNCTION (LAMBDA (Y)
                                          (OR (EQ Y R)
                                              (APPLY* (CAR Y)
                                                       X
                                                       (CADDR Y])
                    (NEQUAL (OR (FMEMB R NE)
                                (SETQ NE (CONS R NE)
                                (HELP))
                    (GO L0))
                    ((NULL NE)
                     (GO L1A))
                    ((CDR NE)
                     (SETQ NE (SORT NE (FUNCTION (LAMBDA (X Y)
                                                  (IGREATERP (CADDR X)
                                                            (CADDR Y)
                    (COND
                      ((NULL UB)
                       (GO L2A)))
                    L2B [COND
                      ((IGREATERP (CADDAR NE)
                                   UB)
                       (COND
                         ((SETQ NE (CDR NE))
                          (GO L2B))
                         (T (GO L1A))
                    L2 [COND
                      (SETQ UB (CADDAR NE))
                      (SETQ UB (SUB1 UB))
                      (SETQ UR NIL)
                      (COND
                        ((SETQ NE (CDR NE))
                         (GO L2))
                        (T (GO L1A))
                    L2A (SETQ NE (DREV NE))
                      (COND
                        ((NULL LB)
                         (GO L1A)))

```

```

L1B [COND
  ((ILESSP (CADDR NE)
    LB)
  (COND
    ((SETQ NE (CDR NE))
      (GO L1B))
    (T (GO L1A]
  ↗ L1 [COND
  ((EQ LB (CADDR NE))
    (SETQ LB (ADD1 LB))
    (SETQ LR NIL)
    (COND
      ((SETQ NE (CDR NE))
        (GO L1]
  L1A [AND LB UB (COND
    ((MINUSP (SETQ D (IDIFFERENCE UB LB)))
      (RETURN NIL))
    ((ZEROP D)
      (RETURN (MAKEEQUAL Y LB)))
    ↗ ((EQ D ENUMMAX)
      (RETURN (ENUMREL Y LB UB (MAPRPC NE (FUNCTION CADDR))
        T]
    (SETQ M (NCONS (OR LR (AND LB (MAKEGEQ Y LB)))
      (NCONS (OR UR (AND UB (MAKELEQ Y UB)))
        NE)))
    (RETURN (CJREL1 M (QUOTE AND]))

(ORREL
  [LAMBDA (L)
    (NOTOF (ANDREL (MAPCAR L (FUNCTION NOTOF))

(CJREL1
  [LAMBDA (M OP)
    (COND
      ↗ [(CDR M)
        (MK1 OP (SORT M (FUNCTION EXPORDER]
        (T (CAR M])

(ENUMREL
  [LAMBDA (Y LB UB NE CJ)
    (PROG ((EOP (EF CJ))
      (Z (LIST Y T)))
      (MK1 (BC (NOT CJ))
        (MAPCAR (DREML NE (GENLIST (ADD1 (IDIFFERENCE UB LB))
          LB))
          (FUNCTION (LAMBDA (X)
            (FRPLACA (CDR Z)
              X)
            (MK EOP Z]))

(VARTEMS
  [LAMBDA (VTE VTFN)
    (VARTERMS1 VTE 0 (OR VTFN (FUNCTION VTERMP]))

```

```

(VARTERMS1
[LAMBDA (VTX VTE VTFN)
(COND
[(NLISTP VTX)
(AND (APPLY* VTFN VTX)
(LIST (CONS VTX (IMINUS VTE)]
(T (SELECTQ (CAR VTX)
[TIMES (AND (EQ (CADR VTX)
-1)
(NULL (CDDDR VTX))
(APPLY* VTFN (CADDR VTX))
(LIST (CONS (CADDR VTX)
VTE)]
(PLUS (PROG ((VTC (CADR VTX))
(VTZ (CDR VTX))
(VTZ1 (CDDDR VTX))
VTY VTD VTR VTL)
L1 [COND
((NULL (SETQ VTZ (CDR VTZ)))
(RETURN VTL))
((SETQ VTD (VARTERMS1 (SETQ VTY (CAR VTZ))
1 VTFN))
(SETQ VTR (LPLUS VTC (REMV1 VTY VTZ1)))
(COND
((OR (LISTP (CAAR VTD))
(NOT (OCCURSFREE (CAAR VTD)
VTR)))
(FRPLACD (CAR VTD)
(MAKECMUL VTR (CDAR VTD)))
(SETQ VTL (FRPLACD VTD VTL]
(GO L1)))
(AND (APPLY* VTFN VTX)
(LIST (CONS VTX (IMINUS VTE]))

(VTERMP
[LAMBDA (X)
(LITATOM X)]

(RELTERMS
[LAMBDA (VTX VTFN)
(VARTERMS [COND
((ZEROP (CADDR VTX))
(CADR VTX))
(T (MAKEPLUS (CADR VTX)
(MINUS (CADDR VTX)
VTFN])]

(VARSUBP
[LAMBDA (VTV VTX)
(CAR (RELTERMS VTX (FUNCTION (LAMBDA (VTZ)
(EQ VTZ VTV]))
)
(RPAQ ENUMMAX 5)
[DECLARE: EVAL@COMPILE
(ADDTOVAR GLOBALVARS ENUMMAX)
]
[DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
(BLOCK: NIL RPARITY VPARITY LINVARS LINEAR ALTREL ALTREL1 EPRODREL RELDIV
EEXPTREL EXPTREL1 MAKEGEQ MAKELEQ MAKEQUAL MAKENEQUAL EF VTERMP
RELTERMS VARSUBP)
(BLOCK: MAKEREL MAKEREL RELQUOP RELQUO1)
(BLOCK: CJREL CJREL ANDREL ORREL CJREL1 ENUMREL)
(BLOCK: VARTERMS VARTERMS VARTERMS1)
]
(DECLARE: DONTCOPY
(FILEMAP (NIL (1524 12675 (RPARITY 1536 . 1745) (VPARITY 1749 . 2022) (LINVARS
2026 . 2331) (LINEAR 2335 . 2596) (MAKEREL 2600 . 5132) (RELQUOP 5136 . 5790)
(RELQUO1 5794 . 6202) (ALTREL 6206 . 6289) (ALTREL1 6293 . 6616) (EPRODREL
6620 . 7559) (RELDIV 7563 . 7739) (EEXPTREL 7743 . 8394) (EXPTREL1 8398 .
8598) (MAKEGEQ 8602 . 8666) (MAKELEQ 8670 . 8734) (MAKEQUAL 8738 . 8805) (
MAKENEQUAL 8809 . 8879) (EF 8883 . 8965) (CJREL 8969 . 9048) (ANDREL 9052
. 10887) (ORREL 10891 . 10960) (CJREL1 10964 . 11073) (ENUMREL 11077 . 11352)
(VARTERMS 11356 . 11437) (VARTERMS1 11441 . 12356) (VTERMP 12360 . 12400)
(RELTERMS 12404 . 12567) (VARSUBP 12571 . 12672))))))
STOP

```

(FILECREATED "18-MAR-75 19:35:58" &lt;DEUTSCH&gt;VSEQ.;2 12055

changes to: VSEQCOMS MAKEL MKSETEL SETELMERGE SETELCONST ELENUM ELCAT  
 ELCASE ELCASE1 MAKELLENGTH MAKESUBSEQ SUBENUM SUBCONCAT SUBCAT1 SUBSETEL XLSUBSEQ  
 INDEXCHECK MKCONCAT CONCATITEM MODEL MODECONCAT MODESETEL MODESUBSEQ OMCONCAT  
 OMSUBSEQ OMCSEQ MAKEUSEQ OMCSEQ\* OMCSTRUCT

previous date: "18-MAR-75 19:30:58" &lt;DEUTSCH&gt;VSEQ.;1)

```
(LISPXPRT (QUOTE VSEQCOMS)
  T T)
(RPAQQ VSEQCOMS ((PROP (OPMAKE OPMODE MCFN XLFN)
  EL SETEL LENGTH CONCAT SUBSEQ CONSTSEQ UNDEFSEQ
  CONSTSEQ* CONSTRUCT)
  (PROP SFDEF *SEQUAL)
  (ENDUMP VSEQ)))
(DEFLIST(QUOTE(
  (EL (MAKEL 2))
  (SETEL (MKSETEL ARB))
  (LENGTH (MAKELLENGTH 1 NIL T))
  (CONCAT (MKCONCAT ARB NIL T))
  (SUBSEQ (MAKESUBSEQ 3 NIL T))
  (CONSTSEQ (MK1 FORM))
  (UNDEFSEQ (MAKEUSEQ 2))
  (CONSTSEQ* (MK1 FORM))
  (CONSTRUCT (MK1 FORM))
  ))(QUOTE OPMAKE))
(DEFLIST(QUOTE(
  (EL (MODEL 2))
  (SETEL (MODESETEL ARB))
  (LENGTH INT)
  (CONCAT (MODECONCAT ARB))
  (SUBSEQ (MODESUBSEQ 3))
  (CONSTSEQ (I 3))
  (UNDEFSEQ NIL)
  (CONSTSEQ* (CAR ARB))
  (CONSTRUCT (CAR ARB))
  ))(QUOTE OPMODE))
(DEFLIST(QUOTE(
  (EL NIL)
  (SETEL NIL)
  (LENGTH OMCMP)
  (CONCAT OMCONCAT)
  (SUBSEQ OMSUBSEQ)
  (CONSTSEQ OMCSEQ)
  (UNDEFSEQ OMCSEQ)
  (CONSTSEQ* OMCSEQ*)
  (CONSTRUCT OMCSTRUCT)
  ))(QUOTE MCFN))
(DEFLIST(QUOTE(
  (EL XLEL)
  (SETEL NIL)
  (LENGTH NIL)
  (CONCAT NIL)
  (SUBSEQ XLSUBSEQ)
  (CONSTSEQ NIL)
  (UNDEFSEQ NIL)
  (CONSTSEQ* NIL)
  (CONSTRUCT NIL)
  ))(QUOTE XLFN))
(DEFLIST(QUOTE(
  (*SEQUAL ([[VAR #1 (SEQMODE (ATMODE (QUOTE ANY:)
    (VAR #2 (SEQMODE (ATMODE (QUOTE ANY:)
      [AND (EQUAL (LENGTH #1)
        (LENGTH #2))
        (FA (BV (VAR @J (MODEX INT)))
          (AND (GEQ @J 1)
            (LEQ @J (LENGTH #1)))
          (EQUALP (EL #1 @J)
            (EL #2 @J)
            T))
  ))(QUOTE SFDEF))
```

(RPAQQ VSEQFNS

```
(MAKEL MKSETEL SETELMERGE SETELCONST ELENUM ELCAT ELCASE ELCASE1
  MAKELENGTH MAKESUBSEQ SUBENUM SUBCONCAT SUBCAT1 SUBSETEL
  XLSUBSEQ INDEXCHECK MKCONCAT CONCATITEM MODEL MODECONCAT
  MODESETEL MODESUBSEQ OMCONCAT OMSUBSEQ OMCSEQ MAKEUSEQ OMCSEQ*
  OMCSTRUCT))
```

(DEFINEQ

(MAKEL

[LAMBDA (X J) (\* edited (1-SEP-74 . 1640))

(COND

((NLISTP X)

(MK1\*(QUOTE EL)

X J))

(T (SELECTQ (CAR X)

(CONSTSEQ (CADDR X))

(CONSTRUCT (CAR (FNTH (CDDR X)

J))))

[CONSTSEQ\*(COND

((NUMBERP J)

([LAMBDA (R)

(COND

(R (CAR R))

(T (BADOBJECT (CADADR X]

(FNTH (CDDR X)

J))))

((NULL (CDDR X))

(BADOBJECT (CADADR X)))

(T (MK (QUOTE LCOND)

(ELENUM (CDDR X)

J 1]

[SUBSEQ (MAKEL (CADR X)

(MKPLUS (LIST (CADDR X)

J -1]

(CONCAT (ELCAT (CDR X)

J))

(SETEL (ELCASE (CADR X)

J

(CDDR X)))

(MK1\*(QUOTE EL)

X J])

(MKSETEL

[LAMBDA (L)

(\* edited (2-AUG-74 . 55))

(COND

((NULL (CDR L))

(CAR L))

((NLISTP (CAR L))

(MK1 (QUOTE SETEL)

L))

(T (SELECTQ (CAAR L)

[SETEL (MKSETEL (CONS (CADAR L)

(APPEND (CDR L)

(SETELMERGE (CDR L)

(CDDAR L]

(CONSTRUCT (SETELCONST L))

[CONSTSEQ\*(COND

((NUMBERP (CADR L))

(SETELCONST L))

(T (MK1 (QUOTE SETEL)

L]

(MK1 (QUOTE SETEL)

L])



```
(MAKELENGTH
 [LAMBDA (X)                                     (* edited (1-SEP-74 . 1641))
 (PROG ((M (MD X)))
 (RETURN
 (COND
 ((EQCAR M (QUOTE STRUCMODE))
 (FLENGTH (CDR M)))
 ((NLISTP X)
 (MK1*(QUOTE LENGTH)
 X))
 (T (SELECTQ (CAR X)
 (SETEL (MAKELENGTH (CADR X)))
 ((CONSTSEQ* CONSTRUCT)
 (FLENGTH (CDDR X)))
 (CONSTSEQ (CADDR X))
 [SUBSEQ (MKPLUS (CONS 1 (CONS (NEGOF (CADDR X))
 (CDDDR X])
 [CONCAT (MKPLUS (MAPCAR (CDR X)
 (FUNCTION MAKELENGTH]
 (MK1*(QUOTE LENGTH)
 X])])
```

```
(MAKESUBSEQ
 [LAMBDA (X M N)                                 (* edited (1-SEP-74 . 1242))
 (COND
 ((AND (EQ M 1)
 (EQCAR N (QUOTE LENGTH))
 (EQ (CADR N)
 X))
 X)
 (([LAMBDA (D)
 (COND
 ((NUMBERP D)
 (MK1 (QUOTE CONSTSEQ*)
 (CONS (MD X)
 (SUBENUM X (LIST 0 M)
 (ADD1 D])
 (MAKEDIFF N M)))
 ((NLISTP X)
 (MK1*(QUOTE SUBSEQ)
 X M N))
 (T (SELECTQ (CAR X)
 [CONSTSEQ (MK1 (QUOTE CONSTSEQ)
 (CONS (CADR X)
 (CONS (MKPLUS (LIST M N -1))
 (CDDDR X])
 [CONSTSEQ*(COND
 [(AND (NUMBERP M)
 (NUMBERP N))
 (PROG [(R (FNTH (CDDR X)
 M))
 (S (FNTH (CDDR X)
 (ADD1 N])
 (RETURN (COND
 [(AND R S (IGEQ (ADD1 N)
 M))
 (MK1 (QUOTE CONSTSEQ*)
 (CONS (CADR X)
 (LDIFF R S])
 (T (BADOBJECT (CADR X])
 (T (MK1*(QUOTE SUBSEQ)
 X M N])
 [SUBSEQ (MAKESUBSEQ (CADR X)
 (MKPLUS (LIST (CADDR X)
 M -1))
 (MKPLUS (LIST (CADDR X)
 N -1])
 (CONCAT (SUBCONCAT (CDR X)
 M N))
 (SETEL (SUBSETEL X M N))
 (MK1*(QUOTE SUBSEQ)
 X M N])
```

```
(SUBENUM
 [LAMBDA (X W K)
 (COND
 ((ILESSP (CAR W)
 K)
 (CONS (MAKEL X (MKPLUS W))
 (SUBENUM X (FRPLACA W (ADD1 (CAR W)))
 K]))
 (* edited (2-AUG-74 . 58))
```

```
(SUBCONCAT
 [LAMBDA (L M N)
 (PROG ((P NIL)
 (K M)
 (K1 N))
 (RETURN
 (MK
 (QUOTE LCOND)
 (MAPCON
 L
 (FUNCTION (LAMBDA (R)
 (PROG ((C (MAKELENGTH (CAR R)))
 Y V)
 (SETQ Y (MAKEDIFF K C))
 (SETQ V
 (LIST [MKAND (LIST (NOTOF P)
 (SETQ P
 (OR (NULL (CDR R))
 (MAKELEQ Y 0])
 (SUBCAT1 R K K1)))
 [COND
 ((CDR R)
 (SETQ K Y)
 (SETQ K1 (MAKEDIFF K1 C])
 (RETURN V]))
```

```
(SUBCAT1
 [LAMBDA (L M N)
 (COND
 ((NULL (CDR L))
 (MAKESUBSEQ (CAR L)
 M N))
 (T
 (PROG ((P NIL)
 (K M)
 (K1 N)
 CL)
 (RETURN
 (MK
 (QUOTE LCOND)
 (MAPCON
 L
 (FUNCTION (LAMBDA (R)
 (PROG ((C (MAKELENGTH (CAR R)))
 Y END V)
 (SETQ Y (MAKEDIFF K1 C))
 [SETQ V
 (LIST
 [MKAND (LIST (NOTOF P)
 (SETQ P
 (OR (NULL (CDR R))
 (MAKELEQ Y 0])
 (MKCONCAT
 (SETQ CL
 (NCONC CL (SETQ END
 (LIST (MAKESUBSEQ (CAR R)
 K K1])
 (COND
 ((CDR R)
 [FRPLACA END (COND
 ((EQ K 1)
 (CAR R))
 (T (MAKESUBSEQ (CAR R)
 K C])
 (SETQ K1 Y)
 (SETQ K 1)))
 (RETURN V]))
```



```
(SUBSETEL
[LAMBDA (X M N)                                (* edited (3-JUN-74 . 1726))
  (PROG ((J (CADDR X))
    P)
    (SETQ P (MK*(QUOTE AND)
      (MK*(QUOTE GEQ)
        J M)
      (MK*(QUOTE LEQ)
        J N)))
    (RETURN (MK*(QUOTE LCOND)
      P
      [MKSETEL (CONS (MAKESUBSEQ (CADR X)
        M N)
          (CONS (MKPLUS (LIST J (NEGOF M)
            1))
            (CDDDR X])
          (NOTOF P)
          (MAKESUBSEQ (MKSETEL (CONS (CADR X)
            (CDDDDR X)))
            M N]))
```

```
(XLSUBSEQ
[LAMBDA (E)                                    (* edited (1-SEP-74 . 1242))
  (PROG [(L (MAPCARN (CDR E)
    (FUNCTION XLX])
    (RETURN (RESTRICT (MKN E L)
      (INDEXCHECK (CAR L)
        (CADR L)
        (CADDR L])
```

```
(INDEXCHECK
[LAMBDA (S I J)                                (* edited (9-JUN-74 . 2303))
  (MK*(QUOTE AND)
    (MK*(QUOTE GEQ)
      I 1)
    (OR (EQ I J)
      (MK*(QUOTE LEQ)
        (MK*(QUOTE DIFFERENCE)
          I J)
        1))
    (MK*(QUOTE LEQ)
      J
      (MK*(QUOTE LENGTH)
        S]))
```

```
(MKCONCAT
[LAMBDA (L)
  (PROG (M)
    (MAPC L (FUNCTION CONCATITEM))
    (RETURN (COND
      ((CDR M)
        (MK1 (QUOTE CONCAT)
          (DREVERSE M)))
      (M (CAR M))
      (T (MK1*(QUOTE CONSTSEQ*)
        (MD (CAR L]))
```



```
(MAKEUSEQ
[LAMBDA (M N) (* edited (1-SEP-74 . 1245))
(MK1*(QUOTE CONSTSEQ)
M N (NULLOBJECT (CADR M]))

(OMCSEQ*
[LAMBDA (L)
(PROG ((M (CAR L))))
(RETURN (AND (EQCAR M (QUOTE SEQMODE))
(EVERY (CDR L)
(FUNCTION (LAMBDA (X)
(EQ (MD X)
(CADR M]))

(OMCSTRUCT
[LAMBDA (L)
(PROG ((M (CAR L))))
(RETURN (AND (EQCAR M (QUOTE STRUCMODE))
(EVERY2 (CDR L)
(CDR M)
(FUNCTION (LAMBDA (X Y)
(EQ (MD X)
(CADDR Y]))
)
)
(DECLARE: DONTCOPY
(FILEMAP (NIL (2172 12031 (MAKEL 2184 . 2918) (MKSETEL 2922 . 3428) (SETELMERGE
3432 . 3750) (SETELCONST 3754 . 4094) (ELENUM 4098 . 4243) (ELCAT 4247 . 4702)
(ELCASE 4706 . 4834) (ELCASE1 4838 . 5189) (MAKELLENGTH 5193 . 5775) (MAKESUBSEQ
5779 . 6966) (SUBENUM 6970 . 7190) (SUBCONCAT 7194 . 7727) (SUBCAT1 7731 .
8547) (SUBSETEL 8551 . 9052) (XLSUBSEQ 9056 . 9299) (INDEXCHECK 9303 . 9591)
(MKCONCAT 9595 . 9838) (CONCATITEM 9842 . 10406) (MODEL 10410 . 10659) (
MODECONCAT 10663 . 10707) (MODESETEL 10711 . 10754) (MODESUBSEQ 10758 . 10801)
(OMCONCAT 10805 . 11013) (OMSUBSEQ 11017 . 11166) (OMCSEQ 11170 . 11451) (
MAKEUSEQ 11455 . 11602) (OMCSEQ* 11606 . 11800) (OMCSTRUCT 11804 . 12028))))))
STOP
```

(FILECREATED "18-MAR-75 19:22:30" &lt;DEUTSCH&gt;VFAC.;6 6905

changes to: UNITDIVIDEP UNITDIVIDE UNITFACTORS

previous date: " 1-SEP-74 01:53:26" &lt;DEUTSCH&gt;VFAC.;5)

```

(LISPXPRINT (QUOTE VFACCOMS)
  T T)
(RPAQQ VFACCOMS ((PROP SFDEF *EXPT1P)
  (ENDUMP VFAC)))
(DEFLIST(QUOTE(
  [*EXPT1P (((VAR #1 (MODEX INT))
    (VAR #2 (MODEX INT)))
    (OR (EQUAL #1 1)
      (AND (NEQUAL #1 0)
        (EQUAL #2 0))
      (AND (EQUAL #1 -1)
        (EQUAL (REMAINDER #2 2)
          0))
    ))))
  ))(QUOTE SFDEF))
(RPAQQ VFACFNS (UNITFACTORS UFACEXPT UNITDIVIDE UNITDIVIDEP UDIVEXPT TERMGCD
  FACTORS POWERFACP MAPALLFACTORS MAPALLFAC1))
[RPAQQ VFACBLOCKS ((MAPALLFACTORS MAPALLFACTORS MAPALLFAC1 (LOCALFREEVARS
  MAPFN])
(DEFINEQ
(UNITFACTORS
[LAMBDA (E)
(COND
  ((NLISTP E)
    NIL)
  ((EQ (CAR E)
    (QUOTE TIMES))
    (CDDR E))
  ((AND (EQ (CAR E)
    (QUOTE PLUS))
    (ZEROP (CADR E)))
    (PROG ((L (CDDR E))
      U X XF N X1 X2 X3)
      L1 [SETQ X
        (COND
          [(NULL L)
            (RETURN (COND
              (XF (UFACEXPT U))
              (T U))
            (EQCAR (SETQ X (CAR L))
              (QUOTE TIMES))
            (SETQ X1 (SETQ X3 NIL))
            [MAPC
              (SETQ X (CDDR X))

```

(\* edited (18-MAR-75 . 1922))

```

(FUNCTION (LAMBDA (Y)
  (COND
    ((EQCAR Y (QUOTE IEXPT))
     (PROG ((N (CADDR Y)))
      (COND
        ((NUMBERP N)
         (SETQ X1 (CONS Y X1)))
        ((EQCAR N (QUOTE PLUS))
         (COND
           ((IGEQ (CADR N)
            0)
            [COND
              ((NEQ (CADR N)
               0)
               (SETQ X1
                (CONS (MK1*(QUOTE IEXPT)
                 (CADR N)
                 (CADR Y))
                 X1])
              [MAPC
                (CDDR N)
                (FUNCTION (LAMBDA (Z)
                  (SETQ X3
                    (CONS (MK1*(QUOTE IEXPT)
                     Z
                     (CADR Y))
                     X3])
                  (SETQ X2 (CONS Y X2]
                ]
            ]
          ]
        ]
      ]
    ]
    [(OR X1 X2)
     (NCONC [MAPCONC (SETQ XF X1)
                  (FUNCTION (LAMBDA (Y)
                    (REXPT (CADR Y)
                     (CADDR Y))
                  (NCONC X3 (NREML (NCONC X2 X1)
                    X])
                (T X)))
      ([AND (EQCAR X (QUOTE IEXPT))
        (NUMBERP (SETQ N (CADDR X])
        (SETQ XF (REXPT (CADR X)
          N)))
      (T (LIST X]
    (COND
      ((NULL U)
       (SETQ U X))
      ((SETQ U (INTERSECTION U X)))
      (T (RETURN NIL)))
      (SETQ L (CDR L))
      (GO L1])
(UFACEXPT
 [LAMBDA (L)
  (PROG ((M L)
        N)
    L1 (COND
      ((NULL (CDR M))
       (RETURN L))
      ((NEQ (CAR M)
       (CADR M))
       (SETQ M (CDR M))
       (GO L1)))
      (SETQ N 2)
    L2 (FRPLACD M (CDDR M))
      (COND
        ((EQ (CAR M)
         (CADR M))
         (ADDIVAR N)
         (GO L2)))
      [SETQ M (CDR (FRPLACA M (MK1* (QUOTE IEXPT)
        (CAR M)
        N])
      (GO L1])

```

```

(UNITDIVIDE
 [LAMBDA (E L) (* edited (18-MAR-75 . 1918))
 (OR (UNITDIVIDEP E L)
 (HELP]))

(UNITDIVIDEP
 [LAMBDA (E L) (* edited (18-MAR-75 . 1917))
 (* Calls HELP if division is not
 exact!))

(COND
 ((NULL L)
 E)
 ((NLISTP E)
 (HELP))
 (T (SELECTQ (CAR E)
 [PLUS (COND
 [(ZEROP (CADR E))
 (MK (QUOTE PLUS)
 (MAPCAR (CDDR E)
 (FUNCTION (LAMBDA (X)
 (UNITDIVIDE X L)
 (T (HELP]
 [TIMES (* The following relies on
 MKTIMES and UNITFACTORS
 combining like factors.)
 (MK (QUOTE TIMES)
 (CONS (CADR X)
 (MAPCAR (CDDR X)
 (FUNCTION (LAMBDA (Y)
 (COND
 ((FMEMB Y L)
 1)
 ((EQCAR Y (QUOTE IEXPT))
 (OR (UDIVEXPT Y L)
 Y))
 (T Y]
 (IEXPT (OR (UDIVEXPT X L)
 (HELP)))
 (COND
 ((AND (EQ X (CAR L))
 (NULL (CDR L)))
 1)
 (T (HELP]))

(UDIVEXPT
 [LAMBDA (X L)
 (COND
 ((FMEMB X L)
 1)
 ((NUMBERP (CADDR X))
 (COND
 [(FMEMB (CADR X)
 L)
 (MK* (QUOTE IEXPT)
 (CADR X)
 (SUB1 (CADDR X)
 (T ([LAMBDA (Y)
 (AND Y (MK* (QUOTE IEXPT)
 (CADR X)
 (IDIFFERENCE (CADDR X)
 (CADDRAR Y]
 (SOME L (FUNCTION (LAMBDA (Z)
 (AND (EQCAR Z (QUOTE IEXPT))
 (EQ (CADR Z)
 (CADR X))
 (NUMBERP (CADDR Z))

```

```
(TERMGCD
[LAMBDA (L)
  (PROG (D X)
    L1 (COND
      [(EQCAR (SETQ X (CAR L))
        (QUOTE TIMES))
        (SETQ X (ABS (CADR X)))
        (SETQ D (COND
          (D (GCD X D))
          (T X]
        (T (RETURN 1)))
      (COND
        ((SETQ L (CDR L))
          (GO L1)))
      (RETURN D])
```

```
(FACTORS
[LAMBDA (Y)

  (* Y is a non-zero integer. Returns sorted list of positive
  prime factors, smallest first.)
```

```
(PROG ((X Y)
  (J 3)
  L)
LX [COND
  ((ZEROP (LOGAND X 1))
    (SETQ L (CONS 2 L))
    (SETQ X (RSH X 1))
    (GO LX))
  ((MINUSP X)
    (SETQ X (IMINUS X]
  (COND
    ((EQ X 1)
      (RETURN L)))
  (SETQ J 3)
LP (COND
  [(IGREATERP (ITIMES J J)
    X)
    (RETURN (DREV (CONS X L]
  ((ZEROP (IREMAINDER X J))
    (SETQ L (CONS J L))
    (SETQ X (IQUOTIENT X J))
    (GO LP))
  (T (SETQ J (IPLUS J 2))
    (GO LP])
```

```
(POWERFACP
[LAMBDA (L)

  (* L is a sorted list of prime factors.
  If the product of L is A*B with B#1, returns
  (A . B), otherwise NIL.)
```

```

(AND (EQ (CAR L)
        (CADR L))
      (PROG ((W L)
             A B C D)
            L1 (SETQ C 1)
            L2 (COND
                ((EQP (CADR W)
                     (CAR W))
                 (SETQ C (ADD1 C))
                 (SETQ W (CDR W))
                 (GO L2))
                ((NULL A)
                 (SETQ A (CAR W))
                 (SETQ B C)
                 (* Precheck guarantees C#1)
                 )
                [(EQP B C)
                 (SETQ A (ITIMES A (CAR W)
                                   (EQ (SETQ D (GCD B C))
                                       1)
                                   (RETURN NIL))
                 (T [SETQ A (ITIMES (IEXPT A (IQUOTIENT B D))
                                   (IEXPT (CAR W)
                                         (IQUOTIENT C D))
                                   (SETQ B D))])
                (COND
                 ((SETQ W (CDR W))
                  (GO L1)))
                (RETURN (CONS A B]))

```

```

(MAPALLFACTORS
 [LAMBDA (MAPL MAPFN)

```

```

(* MAPL is a sorted list of prime factors.
Applies MAPFN to all unique subproducts of L.)

```

```

(MAPALLFAC1 MAPL 1])

```

```

(MAPALLFAC1
 [LAMBDA (MAPL MAPK)
  (PROG (MAPR)
   LP [COND
       ((NULL MAPL)
        (RETURN (CONS (APPLY* MAPFN MAPK)
                      MAPR])
        (SETQ MAPR (NCONC (MAPALLFAC1 (CDR MAPL)
                                       MAPK)
                          MAPR))
       LP1 (SETQ MAPK (ITIMES MAPK (CAR MAPL)))
       (COND
        [(EQP (CAR MAPL)
              (CAR (SETQ MAPL (CDR MAPL)
                              (GO LP1)))]
         (GO LP])
        ]
  )
 [DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
 (BLOCK: NIL UNITFACTORS UFACEXPT UNITDIVIDE UNITDIVIDEP UDIVEXPT TERMGCD
 FACTORS POWERFACP)
 (BLOCK: MAPALLFACTORS MAPALLFACTORS MAPALLFAC1 (LOCALFREEVARS MAPFN))
 ]
 [DECLARE: DONTCOPY
 (FILEMAP (NIL (750 6656 (UNITFACTORS 762 . 2386) (UFACEXPT 2390 . 2817) (
 UNITDIVIDE 2821 . 2948) (UNITDIVIDEP 2952 . 3863) (UDIVEXPT 3867 . 4314) (
 TERMGCD 4318 . 4616) (FACTORS 4620 . 5285) (POWERFACP 5289 . 6081) (
 MAPALLFACTORS 6085 . 6267) (MAPALLFAC1 6271 . 6653))))))
 STOP

```



(FILECREATED "11-MAY-74 21:25:45" VCA 10026

changes to: XLPROC, SUBIODEC, PROCINDECS, PROCOUTDECS, SFSUBP, SFSUB,  
SFSUB1

previous date: "10-MAY-74 11:21:54")

*TV* *COM*

(DEFINEQ

(XLPLY

[LAMBDA (XCL XFN XA)  
(PROG (CTX THLIST NEWCL (OLDCL XCL))  
LP [COND  
((NULL OLDCL)  
(RETURN (DREVERSE NEWCL))  
LP2 [SETCTX (SETQ CTX (CAR (SETQ THLIST OLDCL)  
(SETQ OLDCL (CDR OLDCL))  
(FRPLACD THLIST NIL)  
(COND  
((XLPLY2 XFN XA)  
(SETQ OLDCL (NCONC THLIST OLDCL))  
(GO LP2)))  
(SETQ NEWCL (NCONC (DREV THLIST)  
NEWCL))  
(GO LP])

(XLPLY2

[LAMBDA (FN A)

(APPLY\* FN A)  
NIL])

(\* Just provides a RETFROM point  
for XLCASES)

(ENDJCG

[LAMBDA (CTX VAL)  
(PROG [(POS (STKPOS (QUOTE XLPLY2)  
(COND  
((NEQ (CAR VAL)  
T)  
(HELP))  
((NULL POS)  
(HELP))  
(T (KILLCTX CTX)  
(RETFROM POS])

(CVASSERT

[LAMBDA (X C H1 H2)  
(PUTASSERT X C H1 H2])

(CVLEMMA

[LAMBDA (X J)  
(COND  
((NEQ X T)  
(PROG (((C1 (NEWCTX NIL (QUOTE JSIDE)  
(LIST J X))  
(C2 (NEWCTX NIL (QUOTE JBSIDE)  
NIL)))  
(PUTGOAL X C1)  
(SETQ GOALIST (NCONC1 GOALIST C1))  
(RPLACA THLIST C2)  
(SETCTX C2)  
(CVASSERT X])



(GVSUBST  
[LAMBDA (LIS E)  
(COND  
(LIS (HSUBLIS LIS E))  
(T E))

(XLQ  
[LAMBDA (XLE BV R E C)  
(PROG (E1 LCF XLCV CVF (OLDB BVMODES)  
(BVMODES (ADDBV BV BVMODES)))

(SETCTX C)  
(CVASSERT R)  
LP (SETQ E1 ~~(XLX E)~~) (XLQR E BV)  
(RETURN (COND

~~[LCF~~ (SETCTX (CDR C))  
(KILLTREE C)

(\* XLCASES intervened, E1 is  
~~irrelevant~~)

(MK (SELECTQ (CAR XLE)  
((FA FU)

conditional expression

(QUOTE AND))  
(EX (QUOTE OR))  
(HELP))

(CDR E1)

(MAPCAR ~~LCF~~ [FUNCTION (LAMBDA (X)  
(XLQ XLE BV (MK (QUOTE AND)  
(LIST R X))

E  
(NEWCTX (CDR C)  
(QUOTE JXLCQ)  
(LIST XLE X]

(FUNCTION CDDR]

((AND (NOT CVF)  
[SETQ CVF (SOME XLCV (FUNCTION (LAMBDA (Y)  
(FMEMB (CAR Y)  
(CDR BV]

(\* If variable is from higher  
quantifier, punt.)  
(\* A change of variable was  
recommended (see CHANGEVAR).)

(PROG ((V (CAAR CVF))  
V1 S)

(SETQ V1 (CHVNAME V))  
[SETQ S

(LIST (CONS V (HSUBLIS  
(LIST (CONS (QUOTE \$\$)  
V1))

(CDAR CVF]

(SETQ BVMODES (ADDBV [SETQ BV  
(MKBV (HSUBLIS S  
(CDR BV]

OLDB))

(SETQ R (FSUBMAKE S R))  
(SETQ E (FSUBMAKE S E)))

(GO LP))

(T (SETCTX (CDR C))  
(KILLTREE C)  
(MK\*(CAR XLE)  
BV R E1))

(CHVNAME  
[LAMBDA (V)  
(PACK (FRPLACA (QUOTE (X 0))  
V]))

(XLQR  
[X (E BV) (\* Just provides a RETFROM point  
for XLCASES)  
(XLX E)]

(ENDJXLQ  
[LAMBDA (CTX VAL)  
(PROG [(POS (STKPOS (QUOTE XLQ)  
(COND  
((NEQ (CAR VAL)  
T)  
(HELP))  
((NULL POS)  
(HELP))  
((NEQ (EVALV (QUOTE C)  
POS)  
CTX)  
(HELP))  
(T (RETFROM POS (EQ (CAR (EVALV (QUOTE XLE)  
POS))  
QUOTE EX])

(CHANGEVAR  
[LAMBDA (E)  
(COND  
((AND (LISTP E)  
(NOT (CONSTP E))  
XLBV  
(OCCURSFREE XLBV E))  
(PROG (V S)  
(COND  
([SETQ V (SOME XLBV (FUNCTION (LAMBDA (X)  
(SETQ S (SOLVEFOR E X (QUOTE \$\$))  
(SETQ XLCV (CONS (CONS V S)  
XLCV]))

(SOLVEFOR  
[LAMBDA (E V V1)  
(COND  
((EQ E V)  
V1)  
((NLISTP E)  
NIL)  
(T  
(SELECTQ  
(CAR E)  
[TIMES (COND  
((AND (EQ (CADR E)  
-1)  
(EQ (CADDR E)  
V)  
(NULL (CDDDR E)))  
(MAKENEG V1]

(\* If E=f{V}, returns g{V1}  
where f{g{x}}=x, or NIL)

```

[PLUS
  (PROG [(R (FMEMB V (CDDR E)
            (RETURN (COND
                    [R (AND [NOT (OCCURSFREE V (SETQ R
                        (REMV1 V (CDR E)
                        (MAKEDIFF V1 (MKPLUS R)
                        ([SETQ R (SOME (CDDR E)
                            (FUNCTION (LAMBDA (X)
                                (AND (EQCAR X (QUOTE TIMES))
                                    (EQ (CADR X)
                                        -1)
                                    (EQ (CADDR X)
                                        V)
                                    (NULL (CDDDR X)
                                        (AND [NOT (OCCURSFREE V (SETQ R
                                            (REMV1 (CAR R)
                                                (CDR E)
                                                (MKPLUS (CONS (LTIMES -1 (LIST V1))
                                                    R]
                                                    NIL))

```

```

(XLVAR
 [LAMBDA (XLE)
 (OR (CVAL XLE)
     XLE])

```

```

(XLFNI (XLIS E])

```

```

(XL FNX
 [LAMBDA (E)

```

```

(XLFNI
 [X (E)]

```

```

(PROG ((E1 (XLIS E)))
  (X J V P)


```

```

(RETURN (COND ((AND XLBV (OCCURSFREE XLBV E1))
  (MAPC (CDR E1)
    (F' CHANGEVAR)
    E1)
  (T (FNVALUE E1
    (NOT NQELFLAG]))


```

```

(SETQ X (CADR E1))
(SETQ J (CADDR E1))
(AND [OR NQELFLAG (NOT (OR (VARP (SETQ V X)
  (AND (EQCAR X (QUOTE SETEL))
      (VARP (SETQ V (CADR X)
        (RETURN E1))
      (SETQ P (ELTEST V J)))
      (RETURN (COND
        [(CDR P)
          (CTXEV (MKCOND (CAR (LCONC P (LIST T E1)
            (P (CAR P))
            (T (CHANGEVAR J)
              E1])


```

```

(ELTEST
 [LAMBDA (X J)
 (PROG (P (EVALARGS (FUNCTION CTXEV)))
 [MAPC (ELLIST X)
 (FUNCTION (LAMBDA (Y)
 (ELTEST2 Y J)
 (RETURN P])


```

```

(ELTEST2
 [LAMBDA (Q J)
 (PROG ((V (CAR Q))
 (R (CADR Q))
 (Y (CADDR Q))
 Z)
 (AND (SETQ Z (FSSUB J V R))
 (ELTESTC Z (FSSUB J V Y])


```

```

(XLEL
 [X (E)

```

```

(PROG ((E1 (XLIS E)))
  (RETURN (MK* 'EL X J))
  (RESTRICT (XLFNI E)
    (MK* 'AND
      (MK* 'GEQ
        J (CADDR E1) 1)
      (MK* 'LEQ
        (CADDR E1) J)
      (MK* 'LENGTH
        X (CADR E1])


```

```

(ELTESTC
[LAMBDA (Z V)
(COND
[(EQ (SETQ Z (CTXEV Z))
T)
(RTRFM ELTEST (COND
((NULL P)
(LIST V))
(T (LCONC P (LIST Z V))
(T (SETQ P (LCONC P (LIST Z V))


```

```

(XLCHOICE
[λ (E)
(PROG ((BVMODES (ADDEV (CADR E) BVMODES))
(NCASEFLAG T))
(RETURN (XLIS E]))

```

```

(XLIS
[LAMBDA (E)
(MKN E (MAPCAR (CDR E)
(FUNCTION XLX))]

```

```

(XLCASES
[LAMBDA (E LCE)
(PROG (POS (XLBV XLBV)
(C (CURCTX))
FL)
TOP (AND XLBV (MAPC (CDR LCE)
(FUNCTION (LAMBDA (X)
(SETQ FL (FREEV X FL))
(FUNCTION CDDR))))

```

```

([λ (E)
(COND
((NEQCAR E1 'LCOND)
(RETURN E1))
((NEQ LCE (SETQ LCE E1))
(GO TOP))
(T *) (CTXEV LCE))

```

```

[COND
((NOT (INTERSECTP FL XLBV))
(XLCASERET (MAPCON (CDR LCE)
(FUNCTION [LAMBDA (X)
(XLCASEGEN (CAR X)
(CADR X)
E C])
(FUNCTION CDDR))
(* Make cases at CG level)

```

```

(SETQ POS (STKPOS))
L2 (OR
^ (SETQ POS (STKPOS (QUOTE XLQ)
1 (STKNTH -1 POS))) (HELP))
(COND
((NOT (INTERSECTP (BVL (EVALV (QUOTE BV)
POS))
FL))
(GO L2)
(* Make cases of lowest
quantifier intersecting FL)

```

```

))
((SETQ POS (STKPOS (QUOTE XLX)
-1 POS))
(RPLACA (STKSCAN (QUOTE LCL)
POS)
(CDR LCE))
(REFROM POS NIL)
(STKEVAL (QUOTE (SETQ LCF T)) POS)
(REFROM POS LCE)
(* Must terminate since XLBV is
just union of all BV lists)
(* See XLQ for what happens
next)


```

)]

```

(XLCASEGEN
[LAMBDA (X Y E C)
(COND
[(EQCAR X (QUOTE OR))
(MAPCONC (CDR X)
(FUNCTION (LAMBDA (Z)
(XLCASEGEN Z Y E C))
(T (XLPLY (LIST (NEWCTX C (QUOTE JCASE)
(LIST X E Y)))
(FUNCTION CVASSERT)
X))]

```

```
(XLCASERET
 [LAMBDA (CL)
  (PROG [(POS (STKPOS (QUOTE XLPLY2)
    (RETEVAL POS (CONS (QUOTE XLADDQ)
      CL))
```

```
(XLADDQ
 [NLAMBDA CL
  (SETQ THLIST (NCONC CL (CDR THLIST)))
  T])
```

XLRESVAL

```
(LISPXPRT (QUOTE VCAFNS)
 T)
```

```
(RPAQQ VCAFNS
 (XLPLY XLPLY2 ENDJCG CVASSERT CVLEMMA XLVAL XLX XLQUANT GVLIS GVSUBST
  XLQ CHVNAME ENDJXLQ CHANGEVAR SOLVEFOR XLVAR XLEL ELTEST
  ELTEST2 ELTESTC XLIS XLCASES XLCASEGEN XLCASERET XLADDQ))
```

```
(LISPXPRT (QUOTE VCAVARS)
 T)
```

```
(RPAQQ VCAVARS ((PROP (JFORM JENDFN)
  JCASE JXLQ JXLCQ JSIDE JBSIDE)
 (PROP XLFN EL FA FU EX QUOTE BV)
 (ENDUMP VCA)))
```

```
(DEFLIST(QUOTE(
 (JCASE ("for case " PRINSTAT ", so that " PRINEXP " => " PRINEXP))
 (JXLQ ("to evaluate " PRINSTAT))
 (JXLCQ ("to evaluate " PRINSTAT " restricted to " PRINSTAT))
 (JSIDE ("for " PRINI " lemma: " PRINSTAT))
 (JBSIDE ("to bypass lemma"))
 ))(QUOTE JFORM))
```

PRINSTAT,  
PRINEXP →  
NPRIN2

```
(JRES1 ("to verify restriction " NPRIN2))
(JRES2 ("assuming restriction " NPRIN2))
```

```
(DEFLIST(QUOTE(
 (JCASE ENDJCG)
 (JXLQ ENDJXLQ)
 (JXLCQ ENDJXLQ)
 (JSIDE ENDJFAG)
 (JBSIDE ENDJFAG)
 ))(QUOTE JENDFN))
```

```
(JRES1 ENDJCG)
(JRES2 ENDJCG)
```

```
(DEFLIST(QUOTE(
 (EL XLEL)
 (FA XLQUANT)
 (FU XLQUANT)
 (EX XLQUANT)
 (QUOTE I)
 (BV I)
 ))(QUOTE XLFN))
```

(EL XLFN) ← XLEL  
(LENGTH XLFNX)  
(CHOICE XLCHOICE)

```
[RPAQQ VCAPROPS ((XLFN (NOSET MACRO)
 (APPLYMAPC VCAPROPS (FUNCTION NEWPROP))
 (PROGN (QUOTE JUSTEVALUATE)
 (FILEMAP (NIL (183 8948 (XLPLY 195 . 641) (XLPLY2 645 . 826) (ENDJCG 830 .
 1042) (CVASSERT 1046 . 1106) (CVLEMMA 1110 . 1412) (XLVAL 1416 . 1552) (XLX
 1556 . 2358) (XLQUANT 2362 . 2744) (GVLIS 2748 . 2920) (GVSUBST 2924 . 3005)
 (XLQ 3009 . 4569) (CHVNAME 4573 . 4640) (ENDJXLQ 4644 . 4955) (CHANGEVAR 4959
 . 5241) (SOLVEFOR 5245 . 6166) (XLVAR 6170 . 6221) (XLEL 6225 . 6676) (ELTEST
 6680 . 6844) (ELTEST2 6848 . 7016) (ELTESTC 7020 . 7213) (XLIS 7217 . 7289)
 (XLCASES 7293 . 8488) (XLCASEGEN 8492 . 8738) (XLCASERET 8742 . 8867) (XLADDQ
 8871 . 8945))))))
STOP
```

(FILECREATED "18-JUL-74 10:15:12" VCG.;154 12986

changes to: VCGCOMS

previous date: "19-JUN-74 01:01:04" VCG.;153)

(LISPXPRINT (QUOTE VCGCOMS)  
T)

(RPAQQ VCGCOMS ((PROP ECATFN VAR EL LENGTH)  
(IFPROP CGFN \* CGOPS)  
(PROP CGFN GOAL STEP1)  
(PROP (JFORM JENDFN)  
JPATH JTEST JGOAL JFASIN JFASOUT JLEMMA)  
(PROP SFDEF \*EQMOD)  
(ENDUMP VCG)))

(DEFLIST(QUOTE(  
(VAR ECATVAR)  
(EL ECATFNX)  
(LENGTH ECATFNX)  
)))(QUOTE ECATFN))  
(RPAQQ CGOPS

(SETQ END IF WHILE ELSE BLKIF BLKWHILE BLKELSE EXIT NEXT STOP START  
ASSERT DECLARE INCLUDE DECVAR DECMODE PROGN LOOP LEMMA ELSEIF  
BLKELSEIF BEGIN EMPTY CANCEL ASSUME BLKREPEAT UNTIL BLKUNTIL FOR  
STEP BLKSTEP RETURN))

(DEFLIST(QUOTE(  
(SETQ CGSETQ)  
(IF CGTEST)  
(WHILE CGTEST)  
(BLKIF CGTEST)  
(BLKWHILE CGTEST)  
(STOP CGSTOP)  
(ASSERT CGASSERT)  
(DECLARE CGASSERT)  
(DECVAR CGDECVAR)  
(PROGN HELP)  
(LEMMA CGLEMMA)  
(ELSEIF CGTEST)  
(BLKELSEIF CGTEST)  
(EMPTY CGEMPTY)  
(ASSUME CGASSERT)  
(UNTIL CGTEST)  
(BLKUNTIL CGTEST)  
(FOR CGSETQ)  
(STEP CGSTEP)  
(BLKSTEP CGSTEP)  
(RETURN CGRETURN)

))(QUOTE CGFN))  
(DEFLIST(QUOTE(  
(GOAL CGGOAL)  
(STEP1 CGSTEP1)  
)))(QUOTE CGFN))

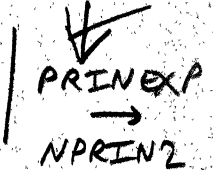
→ (STEP0 CGSTEP)



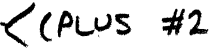
```

(DEFLIST(QUOTE(
  (JPATH ("for paths beginning " PRINTPATH))
  (JTEST ("assuming " PRIN1 " in " PLCN))
  (JGOAL ("for goal " PLCN))
  (JFASIN ("for .FA goal " PLCN ", case " PRINEXP " [in old range]"))
  (JFASOUT ("for .FA goal " PLCN ", case " PRINEXP " [not in old range]"))
  (JLEMMA ("to bypass lemma " PLCN))
)))(QUOTE JFORM))
(DEFLIST(QUOTE(
  (JPATH ENDJCG)
  (JTEST ENDJCG)
  (JGOAL ENDJCG)
  (JFASIN ENDJCG)
  (JFASOUT ENDJCG)
  (JLEMMA ENDJCG)
)))(QUOTE JENDFN))
(DEFLIST(QUOTE(
  [*EQMOD (((VAR #1 (MODEVAR INT))
            (VAR #2 (MODEVAR INT))
            (VAR #3 (MODEVAR INT)))
            (EX (BV (VAR @K (MODEVAR INT)))
                (GEQ @K 0)
                (EQUAL #1 (TIMES @K #3)
                        (PLUS #2

```


  
 PRINEXP
   
 →
   
 NPRIN2





```

)))(QUOTE SFDEF))
[RPAQQ VCGPROPS ((CVAL NIL CTX)
  (EQVAL NIL CTX)
  (CGFN (NOSET)
  (RPAQQ VCGFNS
    (SEGMRG SEGMRG1 CGEN CGSEG CGEXP CGSETQ CVALSET ECATVAR ECATSET
      ECATFNX CGTEST CGASSERT CGGOAL CGOAL CVGOAL CGLEMMA CGDECVAR
      CGSTEP MCONDQ CGSTEP1 STEPVAR STEPINIT CGRETURN CGEMPTY CGSTOP
      RPCVAL FASPLIT FAS1 FAS2 NOCHANGE))
  (RPAQQ VCGVARS (CGOPS))
  (APPLYMAPC VCGPROPS (FUNCTION NEWPROP))
(DEFINEQ
(SEGMRG
  [LAMBDA (OLD NEW)
    (SEGMRG1 (MAPCAR OLD (FUNCTION CADR))
      (MAPCAR NEW (FUNCTION CADR))
      1)])

```

```

(SEGMRG1
 [LAMBDA (OLD NEW K)
  (AND
   OLD (PROG [[OD (DISTRIB OLD (FUNCTION (LAMBDA (X)
                                         (CDAR (FNTH X K]
                                         (ND (DISTRIB NEW (FUNCTION (LAMBDA (X)
                                         (CDAR (FNTH X K]
   (MAPC OD
    (FUNCTION (LAMBDA (X)
      (PROG [(Z (SASSOC (CAR X)
                       ND))
            (X1 (CAR (FNTH (CADR X)
                       K]
            (SEGMRG1 (CDR X)
                    (CDR Z)
                    (ADD1 K))
      (COND
        ((NULL Z)
         (MAPC (CADR X1)
              (FUNCTION KILLTREE))
         (MAPC (CAAR X1)
              (FUNCTION KILLTREE))
         (RPLACA X1 (QUOTE NOBIND)))
        (T (RPLACA (CAR (FNTH (CADR Z)
                              K))
                   (CAR X1]))

```

```

(CGEN
 [LAMBDA (PATH TOP)
  (PROG ((L PATH)
        (FIRST T)
        CL)
  LP (COND
     ((NULL L)
      (RETURN CL))
     ((NEQ (CAAR L)
            (QUOTE NOBIND)))
     (SETQ CL (CAAR L)))
    (T [SETQ CL
         (CGSEG (CDAR L)
                (COND
                 [FIRST (LIST (NEWCTX TOP (QUOTE JPATH)
                                   (LIST (LIST (CAR L)
                                               (T (CAR CL]
                                               (RPLACA (CAR L)
                                                       CL)))
                 (SETQ L (CDR L))
                 (SETQ FIRST NIL)
                 (GO LP])

```

(CGSEG  
[LAMBDA (CGP CGC)

Several CG functions  
 ↗ (\* ~~CGTEST~~ and ~~CGOAL~~ use CGA free)  
 ↗ (\* CGOAL and ~~CVLEFMA~~ use GOALIST free)

↗ (PROG ((CGL CGP) **CGA**  
 (THLIST (APPEND CGC))  
 GOALIST E FN)  
 L1 [COND  
 ((NULL CGL)  
 (RETURN (LIST THLIST GOALIST)))  
 ([AND (SETQ E (CGEXP (CAR CGL)))  
 ↗ (SETQ FN (CGFN (CAR E) (SETQ CGA (AST (CAR CGL))))  
 (SETQ THLIST (XLPLY THLIST FN (CDR E))  
 (SETQ CGL (CDR CGL))  
 (GO L1])

(CGEXP  
[LAMBDA (A)  
(SELECTQ A  
 ((T F D)  
 NIL)  
 (EEXP A])

(CGSETQ  
[LAMBDA (L)  
(CVALSET (CAR L)  
 (XLVAL (CADR L]))

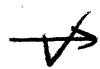
(CVALSET  
[LAMBDA (V X CTX)  
(COND  
 ((VARP V)  
 (SETCVAL V X CTX))  
 ((EQCAR V (QUOTE EL))  
 (CVALSET (CADR V)  
 (MK\* (QUOTE SETEL)  
 (XLVAL (CADR V))  
 (XLVAL (CADDR V))  
 X)  
 CTX))  
 [(EQCAR V (QUOTE VAL))  
 (PROG [(Y (XLVAL (CADR V))  
 (COND  
 ((EQCAR Y (QUOTE REF))  
 (SETCONT (CADR Y)  
 X))  
 (T (CVASSERT (MAKEQUALP Y (GENCONT X))  
 (T (HELP])

(ECATVAR  
[LAMBDA (W V E Q FLAG CTX)  
(COND  
 ((AND FLAG (NOT Q)  
 (LINEXP E T))  
 (ECATSET V E CTX])

```
(ECATSET
 [LAMBDA (V E CTX)
 (COND
 ((NULL (CVAL V CTX))
 (RPCVAL V E CTX)
 (SETCVAL V E CTX)
 (SETEQUAL V T CTX]))
```

```
(ECATFNX
 [LAMBDA (W X Y Q FLAG CTX)
 (AND FLAG (OR (NEQCAR Y (CAR X))
 (EXPORDER X Y))
 (COND
 (Q (AND (OCCURSFREE (BVL (CADR Q))
 X)
 (PROGN (ADDFNPATTERN X (BVL (CADR Q))
 (CADDR Q)
 Y
 (LIST Q)
 CTX))
```

*(OR (NOT (VARP Y))  
(CVAL Y CTX))*



```
(ADDFNVALUE X Y (LIST W)
 CTX)
 T])
```

```
(CGTEST
 [LAMBDA (L)
 (PROG ((E (XLVAL (CAR L)))
 (M (QUOTE FALSE))
 C E1)
 (SETQ E1 (SELECTQ (CADR CGL)
 (T (SETQQ M TRUE)
 E)
 (F (NOTOF E))
 (HELP)))
 [SETQ C (NEWCTX NIL (QUOTE JTEST)
 (LIST M (CAR L))
 (RPLACA THLIST C)
 (CVASSERT E1 C])
```

(\* CGL is bound by CGSEG)

*CGA*

```
(CGASSERT
 [LAMBDA (L)
 (CVASSERT (XLVAL (CAR L)
 T])
```

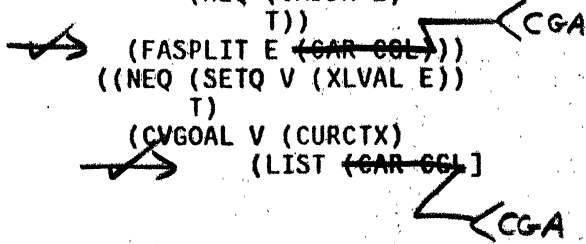
```
(CGGOAL
 [LAMBDA (L)
 (CGOAL (CAR L])
```

(CGOAL  
[LAMBDA (E)

(\* CG- FREE VARIABLES ARE BOUND  
BY CGSEG)

(PROG (C V)

[SETQ GOALIST (NCONC GOALIST (SETQ V (COND  
((AND (EQCAR E (QUOTE FA))  
(NEQ (CADDR E)  
T))  
(FASPLIT E (~~CAR CGL~~)))  
((NEQ (SETQ V (XLVAL E))  
T)  
(CVGOAL V (CURCTX)  
(LIST (~~CAR CGL~~))



(RETURN V])

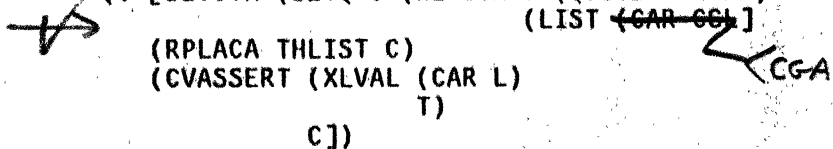
(CVGOAL  
[LAMBDA (X C A)  
(COND

[(EQCAR X (QUOTE AND))  
(MAPCONC (CDR X)  
(FUNCTION (LAMBDA (Y)  
(CVGOAL Y C A])  
(T (PROG ((C1 (NEWCTX C (QUOTE JGOAL)  
A)))  
(RETURN (XLPLY (LIST C1)  
(FUNCTION PUTGOAL)  
X]))

(CGLEMMMA  
[LAMBDA (L)

(PROG [(C (CURCTX))  
(V (CGOAL (CAR L])  
(COND

(V [SETCTX (SETQ C (NEWCTX C (QUOTE JLEMMMA)  
(LIST (~~CAR CGL~~))



(RPLACA THLIST C)  
(CVASSERT (XLVAL (CAR L)  
T)

C])

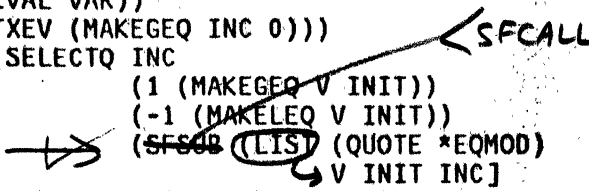
(CGDECVAR  
[LAMBDA (L)

(MAPC L (FUNCTION (LAMBDA (X)  
(COND  
((EQ (CAR X)  
(QUOTE IVAR))  
(SETCVAL (CADR X)  
(XLVAL (CADDR X)  
NIL T]))

(CGSTEP  
[LAMBDA (L)

(\* Doesn't precompute increment  
and limit)

(PROG ((VAR (STEPVAR L))  
(INIT (STEPINIT L))  
(LIM (XLVAL (CADR L)))  
(INC (XLVAL (CAR L)))  
V TEST S C)  
(SETQ V (XLVAL VAR))  
(SETQ S (CTXEV (MAKEGEQ INC 0)))  
[CVASSERT (SELECTQ INC



(1 (MAKEGEQ V INIT))  
(-1 (MAKELEQ V INIT))  
(SFCALL (LIST (QUOTE \*EQMOD)  
V INIT INC])  
(SETQ TEST (MCONDQ S (MAKELEQ V LIM)  
(MAKEGEQ V LIM)))

[SETQ C  
(NEWCTX  
NIL  
(QUOTE JTEST)  
(LIST  
(SELECTQ  
(CADR CGL)  
(T (QUOTE TRUE))  
(F [SETQ TEST  
(MCONDQ S (MCONDQ (MAKELEQ INIT LIM)  
[MKAND (LIST (NOTOF TEST)  
(MAKELEQ V (MAKEPLUS  
LIM INC])  
(MAKEQUAL V INIT))

(MCONDQ (MAKEGEQ INIT LIM)  
[MKAND (LIST (NOTOF TEST)  
(MAKEGEQ V (MAKEPLUS LIM  
INC])  
(MAKEQUAL V INIT)]

(QUOTE FALSE))  
(HELP))  
(CAR CGL) CGA  
(RPLACA THLIST C)  
(CVASSERT TEST C])

(D (\* This is a STEP, i.e.  
for an enclosing loop.)  
(RETURN  
(CVASSERT TEST)))

(MCONDQ  
[NLAMBDA (MCP MCQ MCR)  
([LAMBDA (MCX)  
(SELECTQ MCX  
(T (EVAL MCQ))  
(NIL (EVAL MCR))  
(MK (QUOTE LCOND)  
(LIST MCX (EVAL MCQ)  
(NOTOF MCX)  
(EVAL MCR])  
(EVAL MCP])

MCONDQ-MACRO: ((P X Y) ((λ (PRED)  
(MCONDQ1 PRED (OR (NULL PRED) X)  
(OR (EQ PRED T) Y))) P))

(MCONDQ1  
[λ (P X Y) (SELECTQ P (T X) (NIL Y)  
(MK 'LCOND <P X (NOTOF P) Y>)])

(CGSTEP1  
[LAMBDA (L)  
(PROG ((VAR (STEPVAR L)))  
(RETURN (CVALSET VAR (XLVAL (MAKEPLUS VAR (CADR (EEXP (AST L))))

(STEPVAR  
[LAMBDA (L)  
(CADR (SEXP (TUP (AST L))

(STEPINIT  
[LAMBDA (L)  
(CADDR (SEXP (TUP (AST L))

(\* Doesn't precompute)

(CGRETURN  
[LAMBDA (L)  
(COND  
((LISTP L)  
(CGSETQ (CONS (CURPROC)  
L])) (RESULTNAME \*)

(CGEMPTY  
[LAMBDA (L)  
NIL])

(CGSTOP  
[LAMBDA (L)  
NIL])

(RPCVAL  
[LAMBDA (V X CTX)  
(MAPPROP (QUOTE CVAL)  
CTX  
(FUNCTION (LAMBDA (E Z)  
(COND  
((OCCURSFREE V Z)  
(SETCVAL E (FSSUB X V Z)  
CTX]))

(FASPLIT

[LAMBDA (X A)  
(PROG ((GL (GVLIS (CDADR X)))  
(R (CADDR X))  
(E (CADDR X))  
(C (CURCTX))  
Y C1 C2 R1 P1 R2 Z)  
(SETQ R1 (GVSUBST GL R))  
(SETQ P1 (GVSUBST GL E))  
(SETQ R2 (XLVAL R1 NIL T))  
(SETQ Z (LIST P1 X GL))  
(RETURN (COND

(\* X IS (FA VL R E), R # T)

[(EQ R1 R2)  
(COND  
((AND (THCLAUSEP X C)  
(NOCHANGE P1 NIL (CDADR X)  
C))  
NIL)  
(T [SETCTX (SETQ C1 (NEWCTX C (QUOTE JGOAL)  
(LIST A])  
(XLPLY (LIST C1)  
(FUNCTION FAS2)  
(CONS R1 Z])

(T [COND  
((SETQ Y (MK (QUOTE AND)  
(LIST R1 R2))))  
[SETCTX (SETQ C1 (NEWCTX C (QUOTE JFASIN)  
(LIST A Y])  
(SETQ C1 (XLPLY (LIST C1)  
(FUNCTION FAS1)  
(CONS Y Z])

{(COND  
((THCLAUSEP X C) #)  
(T (F FAS2)))

[COND  
((SETQ Y (MK (QUOTE AND)  
(LIST (MAKENOT R1)  
R2))))  
[SETCTX (SETQ C2 (NEWCTX C (QUOTE JFASOUT)  
(LIST A Y])  
(SETQ C2 (XLPLY (LIST C2)  
(FUNCTION FAS2)  
(CONS Y Z])

(NCONC C1 C2])

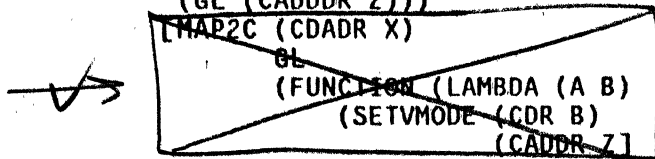
(FAS1

[LAMBDA (Z)  
(FAS2 Z T])



(FAS2

[LAMBDA (Z F1)  
(PROG ((Y (CAR Z))  
(P (CADR Z))  
(X (CADDR Z))  
(GL (CADDR Z)))



(CVASSERT Y NIL X)  
(AND F1 (PUTASSERT P NIL X))  
(PUTGOAL (XLVAL P)  
NIL X])

[MAPC (CADR X)  
(F/V (SETVMODE  
OR (CDR (FASOC (CADR V) GL)  
(CADR V))  
(CADDR V])

(NOCHANGE

[LAMBDA (E FL BL CTX)  
(NOTANY (FREEV E FL BL)  
(FUNCTION (LAMBDA (X)  
(AND (CVAL X CTX)  
(NOT (EQUAL X CTX))

)  
(DEFLIST(QUOTE(  
(CVALSET (9-JUN-74 . 2318))  
(ECATVAR (27-MAY-74 . 114))  
(ECATFNX (6-JUN-74 . 1018))  
(CGSTEP (13-JUN-74 . 42))  
(MCONDQ (13-JUN-74 . 1509))  
(FASPLIT (27-MAY-74 . 852))  
(FAS2 (27-MAY-74 . 852))  
(NOCHANGE (27-MAY-74 . 852))

))(QUOTE EDITDATE))  
(RPAQQ CGOPS  
(SETQ END IF WHILE ELSE BLKIF BLKWHILE BLKELSE EXIT NEXT STOP START  
ASSERT DECLARE INCLUDE DECVAR DECMODE PROGN LOOP LEMMA ELSEIF  
BLKELSEIF BEGIN EMPTY CANCEL ASSUME BLKREPEAT UNTIL BLKUNTIL FOR  
STEP BLKSTEP RETURN))

(LISPPRINT (QUOTE VCGCOMS)

T)  
[RPAQQ VCGCOMS ((PROP ECATFN VAR EL LENGTH)  
(IFPROP CGFN \* CGOPS)  
(PROP CGFN GOAL STEP1)  
(PROP (JFORM JENDFN)  
JPATH JTEST JGOAL JFASIN JFASOUT JLEMMA)  
(PROP SFDEF \*EQMOD)  
(ENDUMP VCG)  
(DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY (ADDVARS  
(NLAMA)  
(NLAML MCONDQ)

[DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY  
(ADDOVAR NLAMA)  
(ADDOVAR NLAML MCONDQ)

]

(DECLARE: DONTCOPY

(FILEMAP (NIL (2360 11960 (SEGMRG 2372 . 2504) (SEGMRG1 2508 . 3797) (CGEN  
3801 . 4258) (CGSEG 4262 . 4846) (CGEXP 4850 . 4955) (CGSETQ 4959 . 5027)  
(CVALSET 5031 . 5447) (ECATVAR 5451 . 5567) (ECATSET 5571 . 5700) (ECATFNX  
5704 . 6025) (CGTEST 6029 . 6541) (CGASSERT 6545 . 6625) (CGGOAL 6629 . 6672)  
(CGOAL 6676 . 7374) (CVGOAL 7378 . 7753) (CGLEMMMA 7757 . 8106) (CGDECVAR 8110  
. 8283) (CGSTEP 8287 . 9441) (MCONDQ 9445 . 9660) (CGSTEP1 9664 . 9792) (STEPVAR  
9796 . 9850) (STEPINIT 9854 . 9967) (CGRETURN 9971 . 10063) (CGEMPTY 10067  
. 10101) (CGSTOP 10105 . 10138) (RPCVAL 10142 . 10322) (FASPLIT 10326 . 11438)  
(FAS1 11442 . 11479) (FAS2 11483 . 11813) (NOCHANGE 11817 . 11957))))))  
STOP

(FILECREATED " 1-SEP-74 01:45:15" VCM.;50 23955

changes to: CATPUT VCMPROPS CATNOT REMCATNOT CEVTV CEVTN CTXEV CEVCOND  
CEVARGS VCMSTATS PUTLBI LBCONS VCMCOMS VCMBLOCKS

previous date: "30-AUG-74 06:34:06" VCM.;49)

*Edith*

```

(LISPXPRINT (QUOTE VCMCOMS)
T)
(RPAQQ VCMCOMS ((PROP (JFORM JENDFN)
JROOT)
(IFPROP (CATFN RCATFN CEVFN)
EQUAL NEQUAL GEQ LEQ FA EX FU AND OR LCOND NOT EQUALP NEQUALP
RESTRICT EQV NEQV)
(ENDUMP VCM)))
(DEFLIST(QUOTE(
(JROOT ("root"))
)))(QUOTE JFORM))
(DEFLIST(QUOTE(
(JROOT NIL)
)))(QUOTE JENDFN))
(DEFLIST(QUOTE(
(EQUAL CATEQUAL)
(NEQUAL CATEQX)
(GEQ CATGEQ)
(LEQ CATLEQ)
(FA CATFA)
(EX CATEU)
(FU CATEU)
(AND CATAND)
(OR CATOR)
(NOT CATNOT)
)))(QUOTE CATFN))
(DEFLIST(QUOTE(
(EQUAL REMCATEQUAL)
(NEQUAL REMCATREL)
(GEQ REMCATREQ)
(LEQ REMCATREQ)
(FA REMCATFA)
(OR REMCATOR)
(NOT REMCATNOT)
(EQUALP REMCATEQUALP)
)))(QUOTE RCATFN))
(DEFLIST(QUOTE(
(EQUAL (CEVREL))
(NEQUAL (CEVREL))
(GEQ (CEVREL))
(LEQ (CEVREL))
(FA (CEVTN))
(EX (CEVTN))
(FU (CEVTN))
(AND (CEVTN CEVARGS))
(OR (CEVTN CEVARGS))
(LCOND (CEVTN CEVARGS))
(NOT (CEVTN))
(EQUALP (CEVTN))
(NEQUALP (CEVTN))
(RESTRICT (CEVRES))
(EQV (CEVTN CEVARGS))
(NEQV (CEVTN CEVARGS))
)))(QUOTE CEVFN))
(RPAQQ VCMBLKFN (CAT1 DIFFORM))

```

*→ (EQV CATREL)*  
*→ (EQV CATREL)*

*→ (EQV REMCATEQUALP)*

*→ (EQV (CEVTN))*

```
[RPAQQ VCMPPROPS (((BLTERM BLSEP BLCON . BLFROM)
  (NOSET MACRO))
  (CEFFORT NIL HASH (SETHASHQ CEA 100 2.0)
    CHASH)
  (HIST NIL CTX)
  (CLAUSENO NIL CTX)
  (CCC NIL HASH (SETHASHQ CCCA 100 2.0)
    CHASH)
  (CJUST NIL HASH (SETHASHQ CJA 100 2.0)
    CHASH)
  (CPH NIL HASH (SETHASHQ CPHA 100 2.0)
    CHASH)
  (ULBL (MACRO)
    CTX)
  (RC$2 (MACRO)
    CTX)
  (C$1 (MACRO)
    CTX)
  (QC$1 (MACRO)
    CTX)
  (ORC$1 (MACRO)
    CTX)
  (THCLAUSEP (MACRO REM MAP)
    CTX)
  (NOTCLAUSE (MACRO REM)
    CTX)
  (CATFN (NOSET MACRO))
  (RCATFN (NOSET MACRO))
  (CEVFN (NOSET MACRO))
  (ECATFN (NOSET MACRO))
```

```
(RPAQQ VCMFNS
  (GCCPPROP CLRCPROP CHPROPS NEWCTX CATPUT CATREM CAT1 EQCAT CATRELTEST
    CATGEQ CATLEQ CATEQUAL CATEQX CATREL REMCATREL REMCATEQUAL
    REMCATREQ CATEQUALP REMCATEQUALP EQPCAT EQPCAT1 CATAND CATEU
    CATFA REMCATFA FACAT CATOR REMCATOR ORCAT ORCAT1 ORCHECK
    CATNOT REMCATNOT CTXEV CEVTV CEVTVN CEVREL CEVRR CEVARGS
    CEVRES TESTREL TESTRR SIGNUM PUTASSERT PUTGOAL PUTCLAUSE
    HISTFIX ADDCAT REMCLAUSE CLAUSEPUT CLAUSEREM PUTLB1 LBPROD
    LBSUM LBREL LBCONS REMLB2 DIFFORM NUMCLAUSE REPCLAUSE))
```

```
(RPAQQ VCMVARS ((EVALARGS NIL)))
[RPAQQ VCMADVICE ([CLRCTX AFTER NIL (PROGN (CLRCPROP)
  (SETCJUST (ROOTCTX)
    (QUOTE (JROOT))
```

```
(GCCTX AFTER NIL (GCCPROP)
```

```
[RPAQQ VCMBLOCKS
```

```
((PUTLB1 PUTLB1 LBPROD LBSUM LBREL LBCONS)
  (CATBLOCK PUTCLAUSE HISTFIX ADDCAT CATPUT CATRELTEST CATGEQ CATLEQ
    CATEQUAL CATEQX CATREL CATEQUALP CATAND CATEU CATFA CATOR
    ORCHECK CATNOT CLAUSEPUT (ENTRIES PUTCLAUSE CATPUT
    CLAUSEPUT)
```

```
(LOCALFREEVARS EHIST EFLAG ELIST)
(SPECVARS EHIST EFLAG ELIST)
(BLKAPPLYFNS CATEQUAL CATGEQ CATLEQ CATEQX CATEQUALP CATEU
  CATAND CATFA CATOR CATNOT))
```

```
(RCATBLOCK REMCLAUSE CATREM REMCATREL REMCATEQUAL REMCATREQ
  REMCATEQUALP REMCATFA REMCATOR REMCATNOT CLAUSEREM
  (ENTRIES REMCLAUSE CATREM CLAUSEREM)
  (BLKAPPLYFNS REMCATREL REMCATEQUAL REMCATREQ REMCATEQUALP
  REMCATFA REMCATOR REMCATNOT))
```

```
(CEVBLOCK CTXEV CEVREL CEVARGS CEVRES TESTREL (ENTRIES CTXEV)
  (NOLINKFNS CEVTV CEVTVN CEVRR TESTRR)
  (BLKAPPLYFNS CEVREL CEVARGS CEVRES)
```

```
(RPAQQ VCMSTATS (CTXEV# CEVNL# CEVN# CEVT# CEVNIL# CEVX# CEVREL# UTEST#
  CEVRR# LBCONS# TESTRR#))
```

```
(ADDTOVAR BLKLIBRARY CAT1 DIFFORM)
[DECLARE: DOEVAL@COMPILE
```

← CEVCTX#

← PUTCTX# ELIST# REMCTX#  
 PUTLB# LBCONS# REMLB#

```

(DEFLIST(QUOTE(
  [CAT1 (LAMBDA (C P E V CTX)
    (PROG ((D (GETX C P CTX)))
      (COND ((FMEMB E D)
        (OR V (PUTX C P (REMV1 E D)
          CTX)))
        (V (PUTX C P (CONS E D)
          CTX)]
    [DIFFORM (LAMBDA (R FN CTX)
      (PROG ((X (CADR R))
        (Y 0)
        (C (CADDR R)))
      [COND ((EQCAR X (QUOTE PLUS))
        (SETQ X (CADDR X))
        (MAPC (CDDDR (CADR R))
          (FUNCTION
            (LAMBDA
              (Z)
                (COND [(AND (EQCAR Z (QUOTE TIMES))
                  (MINUSP (CADR Z)))
                  (SETQ
                    Y
                    (SELECTQ
                      Y
                      (0 (MAKENEG Z))
                      (MAKEDIFF Y Z]
                (T (SETQ X (MAKEPLUS X Z]
          (RETURN (APPLY* FN X Y C R CTX]
    ))(QUOTE BLKLIBRARYDEF))
  ]
  (APPLYMAPC VCMPPROPS (FUNCTION NEWPROP))
[DECLARE: EVAL@COMPILE
  (ADDTOVAR GLOBALVARS CEA CCCA CJA CPHA)
]
(DEFINEQ
  (GCCPROP
    [LAMBDA NIL
      (MAPC (CHPROPS)
        (FUNCTION (LAMBDA (Z)
          (SETQ Z (CAAR Z))
          (MAPHASH Z (FUNCTION (LAMBDA (Y X)
            (COND
              ((NEQ X (CLIST (CI X)))
                (* CONTEXT WAS COLLECTED BY
                  GCCTX)
                (PUTHASH X NIL Z])
          (CLRCPROP
            [LAMBDA NIL
              (MAPC (CHPROPS)
                (FUNCTION (LAMBDA (Z)
                  (CLRHASH (CAAR Z])
            (CHPROPS
              [LAMBDA NIL
                (MAPCAR (GETP (QUOTE CHASH)
                  (QUOTE NEWPROPS))
                  (FUNCTION CADDR])
            (NEWCTX
              [LAMBDA (CTX J A)
                (PROG ((C (GENCTX CTX)))
                  (SETCJUST C (CONS J A))
                  (SETCCC C 0)
                  (RETURN C])

```

```

(CATPUT
[LAMBDA (E CTX)
(COND
((NLISTP (SETQ E (CTXEV E CTX)))
(SELECTQ E CTX) ← (CURCTX)
(NIL (SUCCESS E CTX))
(T)
(CLAUSEPUT E CTX)))
(T (PROG [(FN (CATFN (CAR E)
(COND
(FN (BLKAPPLY* FN E CTX))
(T (CLAUSEPUT E CTX))

```

→ Abundance of CTX's

```

(CATREM
[LAMBDA (E CTX)
(PROG [(FN (RCATFN (CAR E)
(COND
(FN (BLKAPPLY* FN E CTX))
(T (CLAUSEREM E CTX))

```

```

(CAT1
[LAMBDA (C P E V CTX)
(PROG ((D (GETX C P CTX)))
(COND
((FMEMB E D)
(OR V (PUTX C P (REMV1 E D)
CTX)))
(V (PUTX C P (CONS E D)
CTX))

```

```

(EQCAT
[LAMBDA (E Q FLAG CTX)
(SOME [RELTERMS E (FUNCTION (LAMBDA (X)
(ECATFN (COND
((LISTP X)
(CAR X))
(T (QUOTE VAR]
(FUNCTION (LAMBDA (Y)
(BLKAPPLY* [ECATFN (COND
((LISTP (CAR Y))
(CAAR Y))
(T (QUOTE VAR]
E
(CAR Y)
(CDR Y)
Q FLAG CTX])

```

```

(CATRELTEST
[LAMBDA (E CTX)
(PROG ((V (RC$2 (CADR E)
CTX))
R)
(RETURN (COND
((AND V (CJREF (SETQ V (CONS E V))
(SETQ R (CJREL V T))
T))
[MAPC (CDR V)
(FUNCTION (LAMBDA (X)
(REMCLAUSE X CTX])
(CATPUT R CTX)
T])

```

```

(CATGEQ
[LAMBDA (E CTX)
(OR (CATRELTEST E CTX)
(PROGN (DIFFORM E (FUNCTION [LAMBDA (X Y C E CTX)
(PUTLBI X Y C 1 E CTX])
CTX)
(CATREL E CTX))

```

```
(CATLEQ
 [LAMBDA (E CTX)
  (OR (CATRELTEST E CTX)
    (PROGN (DIFFORM E (FUNCTION [LAMBDA (X Y C E CTX)
      (PUTLB1 Y X (MINUS C)
        1 E CTX]))
      CTX)
    (CATREL E CTX]))
```

```
(CATEQUAL
 [LAMBDA (E CTX)
  (OR (CATRELTEST E CTX)
    (PROGN (DIFFORM E (FUNCTION [LAMBDA (X Y C E CTX)
      (PUTLB1 X Y C 0 E CTX))
      CTX)
    (CATREL E CTX)
    (EQCAT E NIL T CTX]))
```

```
(CATEQX
 [LAMBDA (E CTX)
  (OR (CATRELTEST E CTX)
    (CATREL E CTX))
```

```
(CATREL
 [LAMBDA (E CTX)
  (CLAUSEPUT E CTX)
  (CAT1 (CADR E)
    (QUOTE RC$2)
    E T CTX)
  (ORCHECK (QUOTE (LEQ GEQ))
    CTX))
```

```
(REMCATREL
 [LAMBDA (E CTX)
  (CLAUSEREM E CTX)
  (CAT1 (CADR E)
    (QUOTE RC$2)
    E NIL CTX))
```

```
(REMCATEQUAL
 [LAMBDA (E CTX)
  (REMCATREQ E CTX)
  (EQCAT E NIL NIL CTX))
```

```
(REMCATREQ
 [LAMBDA (E CTX)
  (DIFFORM E (FUNCTION [LAMBDA (X Y C E CTX)
    (REMLB2 X E CTX))
    CTX)
  (REMCATREL E CTX))
```

```
(CATEQUALP
 [LAMBDA (E CTX)
  (CLAUSEPUT E CTX)
  (EQPCAT E NIL T CTX))
```

```
(REMCATEQUALP
 [LAMBDA (E CTX)
  (CLAUSEREM E CTX)
  (EQPCAT E NIL NIL CTX))
```

```
(EQPCAT
 [LAMBDA (E Q FLAG CTX)
  (OR (EQPCAT1 E (CADR E)
    (CADDR E)
    Q FLAG CTX)
    (EQPCAT1 E (CADDR E)
    (CADR E)
    Q FLAG CTX))
```

```
(EQPCAT1
 [LAMBDA (E X Y Q FLAG CTX)
 (PROG [(FN (ECATFN (COND
 ((LISTP X)
 (CAR X))
 (T (QUOTE VAR]
 (RETURN (AND FN (BLKAPPLY* FN E X Y Q FLAG CTX))])
```

```
(CATAND
 [LAMBDA (E CTX)
 (MAPC (CDR E)
 (FUNCTION (LAMBDA (X)
 (CATPUT X CTX]))
```

```
(CATEU
 [LAMBDA (E CTX)
 (PROG [(LIS (MAPCAR (CDADR E)
 (FUNCTION (LAMBDA (V)
 ([LAMBDA (W)
 (SETVMODE W (CADDR V)
 CTX)
 (CONS (CADR V)
 W]
 (GENVAR (CADR V]
 (CATPUT (FSUBMAKE LIS (CADDR E))
 CTX)
 (CATPUT (FSUBMAKE LIS (CADDR E))
 CTX))])
```

```
(CATFA
 [LAMBDA (E CTX)
 (CLAUSEPUT E CTX)
 (FACAT E T CTX)]
```

```
(REMCATFA
 [LAMBDA (E CTX)
 (CLAUSEREM E CTX)
 (FACAT E NIL CTX)]
```

```
(FACAT
 [LAMBDA (E FLAG CTX)
 (PROG ((X (CADDR E)))
 (CAT1 (CAR X)
 (QUOTE QC$1)
 FLAG CTX)
 (COND
 ((EQCAR X (QUOTE EQUAL))
 (EQCAT X E FLAG CTX))
 ((EQCAR X (QUOTE EQUALP))
 (EQPCAT X E FLAG CTX))
```

```
(CATOR
 [LAMBDA (E CTX)
 (CLAUSEPUT E CTX)
 (ORCAT E T CTX)]
```

```
(REMCATOR
 [LAMBDA (E CTX)
 (CLAUSEREM E CTX)
 (ORCAT E NIL CTX)]
```

```
(ORCAT
 [LAMBDA (E FLAG CTX)
 (ORCAT1 E)]
```



```
(ORCAT1
 [LAMBDA (Y)
  (MAPC (CDR Y)
    (FUNCTION (LAMBDA (X)
      (COND
        ((EQCAR X (QUOTE AND))
         (ORCAT1 X))
        ((LISTP X)
         (CAT1 (CAR X)
              (QUOTE ORC$1)
              E FLAG CTX))
      )
    )
```

```
(ORCHECK
 [LAMBDA (L CTX)
  (PROG (DONE X (L1 L))
    LP [MAPC (SETQ X (ORC$1 (CAR L1)
      CTX))
      (FUNCTION (LAMBDA (Y)
        (COND
          ((NOT (FMEMB Y DONE))
           ([LAMBDA (Z)
             (COND
               ((NEQ Z (CDR Y))
                (REMCLAUSE Y CTX)
                (CATPUT (MK (QUOTE OR)
                        Z)
                       CTX)
                (MAPCARN (CDR Y)
                        (FUNCTION (LAMBDA (W)
                          (CTXEV W CTX))
                        )
          )
        )
      )
    )
    ((SETQ L1 (CDR L1))
     (SETQ DONE (APP2 X DONE))
     (GO LP])
```

```
(CATNOT
 [LAMBDA (E CTX)
  (CLAUSEPUT E CTX)
  (SETNOTCLAUSE (CADR E)
   T CTX)
```

```
(REMCATNOT
 [LAMBDA (E CTX)
  (CLAUSEREM E CTX)
  (REMNOTCLAUSE (CADR E)
   CTX)
```


```
(CTXEV
 [LAMBDA (E CTX)
  (INC CTXEV#)
  (COND
    ((NLISTP E)
     (INC CEVNL#)
     E)
    (T (PROG [(FN (CEVFN (CAR E)
      RETURN(COND
        ((NULL FN)
         (INC CEVN#)
         (RETURN E))
        (T (CTXEVI E FN]
          (AND CTX (NEQ CTX (CURCTX)))
          (INC CEVCTX#)
          (RESETFORM (RSETCTX CTX) (CTXEVI E FN]
          (CTXEVI
           [λ (E L) (PROG NIL
             (RETURN E]))
           (COND
             ((NEQ E (SETQ E (BLKAPPLY* (CAR FN)
              E CTX)))
              (INC CEVX#))
             ((SETQ FN (CDR FN))
              (GO LP)))
             (RETURN E])
```



```

(TESTREL
[LAMBDA (R CTX)
  (PROG ((OP (CAR R))
        (E (CADR R))
        (CO (CADDR R))
        (C 0)
        C1
        (AS 0)
        BL TL DL D L X A [ZO (QUOTE (((EL (EL RL (PLUS 1 (TIMES -1 J)
                                     (LENGTH RL)))
                                     2)
                                     0 0) . 1]
        Y
        (S 0)
        P V)
(COND
  ((NEQCAR E (QUOTE PLUS))
   [COND
    ((SETQ D (FASSOC 0 (ULBL E CTX)))
     (SETQ C1 (BLCON D))
     [SETQ P (COND
      ((ZEROP (SETQ X (BLSEP D)))
       (QUOTE EQUAL))
      ((MINUSP X)
       (QUOTE LEQ))
      (T (QUOTE GEQ]
      (COND
        ((RIMPLY P C1 OP CO)
         (RETURN (TESTRR R T D NIL CTX)))
        ((RCLASHAND P C1 OP CO)
         (RETURN (TESTRR R NIL D NIL CTX))
        (RETURN R)))
     (SETQ L (CDDR E))
    L1 (COND
      [(EQCAR (SETQ X (CAR L))
              (QUOTE TIMES))
       (SETQ A (CADR X))
       (SETQ X (LTIMES 1 (CDDR X))
              (T (SETQ A 1)))
      (COND
        ((NULL BL)
         (OR (SETQ BL (ULBL X CTX))
             (RETURN R))
         (FRPLACA (CAR ZO)
                  X)
         (FRPLACD ZO A))
        ((SETQ Y (FASSOC X BL))
         (COND
           ([SOME TL (FUNCTION (LAMBDA (Z)
                                (NOT (FASSOC X (ULBL (BLTERM (CAR Z))
                                CTX))
                                (RETURN R)))
            [SETQ C (PLUS C (TIMES A (BLCON Y)
                                (SETQ TL (CONS (CONS Y A)
                                TL)))
            (T (RETURN R)))
          (SETQ AS (PLUS AS A))
        (COND
          ((SETQ L (CDR L))
           (GO L1)))
        (COND
          ((ZEROP AS))
          [(SETQ D (FASSOC 0 BL))
           (SETQ C (DIFFERENCE C (TIMES AS (BLCON D)
           (T (RETURN R)))
          (SETQ P (IMINUS (SIGNUM (SETQ C (PLUS C CO)
          [SETQ DL (SETQ TL (SORT (CONS ZO TL)
                                (FUNCTION (LAMBDA (X Y)
                                (ILESSP (BLSEP (CAR X))
                                (BLSEP (CAR Y)

```


  
*Remove*
  
 CTX'o

```

L2 (SETQ S (PLUS S (CDR TL)))
  [COND
    ((NEQ S 0)
      (COND
        ([MINUSP (ITIMES P (SETQ P (SIGNUM S)
          (RETURN R)
        (COND
          ((SETQ TL (CDR TL))
            (GO L2)))
        [SETQ V (COND
          ((ZEROP C)
            (SELECTQ OP
              (GEQ (SELECTQ P
                (-1 (MAKEQUAL E CO))
                T))
              (LEQ (SELECTQ P
                (1 (MAKEQUAL E CO))
                T))
            R))
          (T (APPLY* OP 0 C]
        (RETURN (COND
          ((EQ V R)
            R)
          (T (TESTRR R V D DL CTX]))

```

↓  
 /  
 1  
 2  
 3

```

(TESTRR
 [LAMBDA (R V D L CTX)
 (INC TESTRR#)
 V])
 (* For tracing)

```

```

(SIGNUM
 [LAMBDA (X)
 (COND
 ((MINUSP X)
 -1)
 ((ZEROP X)
 0)
 (T 1])

```

```

(PUTASSERT
 [LAMBDA (E CTX H1 H2)
 (PUTCLAUSE E CTX (QUOTE ASSERT)
 H1 H2)])

```

```

(PUTGOAL
 [LAMBDA (E CTX H1 H2)
 (PUTCLAUSE (NOTOF E)
 CTX
 (QUOTE GOAL)
 H1 H2)])

```

```

(COND
 ((AND CTX (NEQ CTX (CURCTX)))
 (INC PUTCTX#)
 (RESETPFORM (RESETCTX CTX)
 (PUTCLAUSE1 E H1 H2)))
 (T (PUTCLAUSE1 E H1 H2]))

```

```

(PUTCLAUSE
 [LAMBDA (E CTX EFLAG H1 H2)
 (CR CTX (SETQ CTX (CURCTX)))
 (OR EFLAG (SETQ EFLAG T))
 (PROG ([EHIST (COND
 (H2 (APP2 (HISTFIX H1 CTX)
 (HISTFIX H2 CTX)))
 (H1 (HISTFIX H1 CTX))
 (ELIST NIL))
 (CATPUT E CTX)
 LP (COND
 (ELIST
 (INC ELIST#)(CATPUT (PROG1 (CAR ELIST)
 (SETQ ELIST (CDR ELIST)))
 (GO LP]))

```

```

(PUTCLAUSE1
 [X (E H1 H2)

```

```

(* Derived clauses.)
(ELIST
 (INC ELIST#)(CATPUT (PROG1 (CAR ELIST)
 (SETQ ELIST (CDR ELIST)))
 (GO LP]))

```

```
(HISTFIX
 [LAMBDA (E CTX)
  (PROG ((H (HIST E CTX)))
    [COND
      ((NEQ (CAR H)
        E)
        (SETHIST E (SETQ H (CONS E H))
          (GETXCTX E (QUOTE THCLAUSEP)
            CTX))
      (RETURN H])
```

```
(ADDCAT
 [LAMBDA (E)
  (SETQ ELIST (ADDELT E ELIST])
```

```
(REMCLAUSE
 [LAMBDA (E CTX)
  (CATREM E CTX)
```

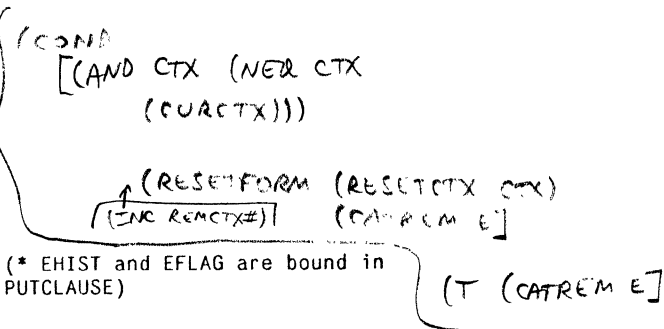
```
(CLAUSEPUT
 [LAMBDA (E CTX)
  (PROG (V) (CTX (CURECTX))
    (AND EHIST (SETHIST E EHIST CTX))
    (SETCCC CTX (SETQ V (ADD1 (CCC CTX)))
      (SETCLAUSENO E (CONS CTX V)
        CTX)
    (CAT1 (CAR E)
      (QUOTE C$1)
      E T CTX)
    (SETHCLAUSEP E EFLAG CTX])
```

```
(CLAUSEREM
 [LAMBDA (E CTX)
  (CAT1 (CAR E)
    (QUOTE C$1)
    E NIL CTX)
  (SETHCLAUSEP E NIL CTX])
```

```
(PUTLB1
 [LAMBDA (X Y C S E CTX)
```

(\* Record X-Y>=C resulting from clause E.  
S=1 for GEQ, S=0 for EQUAL. Assumes no inconsistency! Also  
assumes X and Y result from DIFFORM, e.g. neither is  
(TIMES -n --).)

```
(INC PUTLB#)
 (PROG ((XL (CONS (FRPLACA (QUOTE (A 0 0))
  X)
  (ULBL X CTX)))
  (YL (CONS (FRPLACA (QUOTE (0 0 0))
  Y)
  (ULBL Y CTX)))
  XD)
 (OR (NOT (FASSOC Y (CDR XL)))
  (NEQ S 0)
  (RETURN)
  )
```



(\* Can't represent the result, so punt.)

```

XLP (SETQ XD (CAR XL))
  (AND (OR (ZEROP S)
    (ILEQ (BLSEP XD)
      0))
    (PROG (L1 (C1 (MINUS (BLCON XD)))
      (S1 (IDIFFERENCE S (BLSEP XD)))
      (A (BLTERM XD))
      (AL (ULBL (BLTERM XD)
        CTX))
      (Z YL)
      XP XU YD YS L2 C2 S2 B)
    (COND
      ((LISTP A)
        (SELECTQ (CAR A)
          (TIMES (SETQ XP (NEQ (CADR A)
            1)))
          (PLUS (SETQ XU T)
            NIL)))
    LP [COND
      ((AND (NEQ [SETQ B (BLTERM (SETQ YD (CAR Z)
        A)
        (IGEQ (ITIMES S1 (SETQ YS (BLSEP YD)))
          0))
        (SETQ S2 (IPLUS S1 YS))
        (SETQ L2 (APP2 [OR L1 (SETQ L1 (CONS E (BLFROM XD)
          (BLFROM YD)))
        (SETQ C2 (BLCON YD))
        [SETQ C2 (COND
          ((MINUSP S2)
            (MINUS (PLUS C1 C C2)))
          (T (PLUS C1 C C2))
        (SETQ AL (LBCONS B S2 C2 L2 AL))
        (SETULBL B (LBCONS A (IMINUS S2)
          (MINUS C2)
          L2
          (ULBL B CTX))
        CTX)
        (COND
          ((OR (AND XP (LBPROD A B))
            (AND (LISTP B)
              (EQ (CAR B)
                (QUOTE TIMES))
              (NEQ (CADR B)
                1)
              (LBPROD B A))
            (AND XU (LBSUM A B))
            (AND (LISTP B)
              (EQ (CAR B)
                (QUOTE PLUS))
              (LBSUM B A)))
          (ADDCAT (LBREL A B C2 S2]
        (COND
          ((SETQ Z (CDR Z))
            (GO LP)))
        (SETULBL A AL CTX))
    (COND
      ((SETQ XL (CDR XL))
        (GO XLP]))
(LBPROD
  [LAMBDA (XP Y)
    (OR (NUMBERP Y)
      (AND (EQ Y (CADDR XP))
        (NULL (CDDDR XP)))
      (AND (LISTP Y)
        (EQ (CAR Y)
          (QUOTE TIMES))
        (OR (EQ (CDDR Y)
          (CDDR XP))
          (ZEROP (REMAINDER (CADR Y)
            (CADR XP]))))

```

```

(LBSUM
  [LAMBDA (XU Y)
    (SELECTQ (CAR (LISTP Y))
      [PLUS (SOME (CDR Y)
        (FUNCTION (LAMBDA (W)
          (LBSUM XU W)
        ]
      ]
    [TIMES (COND
      [(CDDDR Y)
        (SOME (CDDR XU)
          (FUNCTION (LAMBDA (W)
            (AND (LISTP W)
              (EQ (CDDR W)
                (CDDR Y))
            (EQ (CAR W)
              (QUOTE TIMES))
          ]
        ]
      ]
      (T (LBSUM XU (CADDR Y)
        (SOME (CDDR XU)
          (FUNCTION (LAMBDA (W)
            (OR (EQ W Y)
              (AND (LISTP W)
                (EQ (CADDR W)
                  Y)
              (EQ (CAR W)
                (QUOTE TIMES))
              (NULL (CDDDR W))
            ]
          ]
        ]
      ]
    ]
  ]

```

```

(LBREL
  [LAMBDA (X Y C S)
    (PROG ((D (MAKEDIFF X Y)))
      (RETURN (COND
        ((MINUSP S)
          (MK*(QUOTE LEQ)
            D
            (MINUS C)))
        ((ZEROP S)
          (MK*(QUOTE EQUAL)
            D C))
        (T (MK*(QUOTE GEQ)
          D C]))
      ]
    ]

```

```

↗ (LBCONS
  [LAMBDA (Y S C L XL) (INC LBCONS#)
    (PROG ((M (MINUSP S))
      CO
      (DL XL)
      (D1 (FASSOC Y XL)))
      (COND
        ((NULL D1)
          (RETURN (CONS (CONS Y (CONS S (CONS C L)))
            XL)))
        ((ILEQ (ITIMES S (BLSEP D1))
          0)
          (* Weak contradiction, can't
            represent it safely.)
          (RETURN XL)))
        (INC LBCONS#)
      ]
    ]
  LOC (AND [COND
    [M (OR (LESSP (SETQ CO (BLCON D1))
      C)
      (AND (EQP CO C)
        (IGREATERP (BLSEP D1)
          S]
      (T (OR (GREATERP (SETQ CO (BLCON D1))
        C)
        (AND (EQP CO C)
          (ILESSP (BLSEP D1)
            S]
        [SETQ D1 (FASSOC Y (SETQ DL (CDR (FMEMB D1 DL)
          (GO LOC))
        (RETURN (LDCONC XL DL (CONS (CONS Y (CONS S (CONS C L)))
          DL]))
    ]
  ]

```





```

(CEVTVN (31-AUG-74 . 1924))
(CEVARGS (31-AUG-74 . 1928))
(CEVRES (26-AUG-74 . 622))
(PUTLB1 (31-AUG-74 . 1933))
(LBSUM (29-AUG-74 . 1930))
(LBCONS (31-AUG-74 . 1932))
(DIFFORM (29-AUG-74 . 2143))
)) (QUOTE EDITDATE))
(RPAQ EVALARGS NIL)
(APPLYMAPC VCMADVICE (FUNCTION ADVISE))
[DECLARE: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY
(BLOCK: NIL GCCPROP CLRCPROP CHPROPS NEWCTX CAT1 EQCAT EQPCAT EQPCAT1 FACAT
ORCAT ORCAT1 CEVTV CEVTVN CEVRR TESTRR SIGNUM PUTASSERT PUTGOAL
REMLB2 DIFFORM NUMCLAUSE REPCLAUSE)
(BLOCK: PUTLB1 PUTLB1 LBPROD LBSUM LBREL LBCONS)
(BLOCK: CATBLOCK PUTCLAUSE HISTFIX ADDCAT CATPUT CATRELTEST CATGEQ CATLEQ
CATEQUAL CATEQX CATREL CATEQUALP CATAND CATEU CATFA CATOR ORCHECK
CATNOT CLAUSEPUT (ENTRIES PUTCLAUSE CATPUT CLAUSEPUT)
(LOCALFREEVARS EHIST EFLAG ELIST)
(SPECVARS EHIST EFLAG ELIST)
(BLKAPPLYFNS CATEQUAL CATGEQ CATLEQ CATEQX CATEQUALP CATEU CATAND
CATFA CATOR CATNOT))
(BLOCK: RCATBLOCK REMCLAUSE CATREM REMCATREL REMCATEQUAL REMCATREQ
REMCATEQUALP REMCATFA REMCATOR REMCATNOT CLAUSEREM
(ENTRIES REMCLAUSE CATREM CLAUSEREM)
(BLKAPPLYFNS REMCATREL REMCATEQUAL REMCATREQ REMCATEQUALP REMCATFA
REMCATOR REMCATNOT))
(BLOCK: CEVBLOCK CTXEV CEVREL CEVARGS CEVRES TESTREL (ENTRIES CTXEV)
(NOLINKFNS CEVTV CEVTVN CEVRR TESTRR)
(BLKAPPLYFNS CEVREL CEVARGS CEVRES))
]
(CS VCMSTATS)
(LISPXPRINT (QUOTE VCMCOMS)
T)
(RPAQ VCMCOMS ((PROP (JFORM JENDFN)
JROOT)
(IFPROP (CATFN RCATFN CEVFN)
EQUAL NEQUAL GEQ LEQ FA EX FU AND OR LCOND NOT EQUALP NEQUALP
RESTRICT EQV NEQV)
(ENDUMP VCM))))
(DECLARE: DONTCOPY
(FILEMAP (NIL (4663 21699 (GCCPROP 4675 . 5108) (CLRCPROP 5112 . 5224) (CHPROPS
5228 . 5352) (NEWCTX 5356 . 5496) (CATPUT 5500 . 5755) (CATREM 5759 . 5908)
(CAT1 5912 . 6159) (EQCAT 6163 . 6494) (CATRELTEST 6498 . 6985) (CATGEQ 6989
. 7161) (CATLEQ 7165 . 7354) (CATEQUAL 7358 . 7561) (CATEQX 7565 . 7644) (CATREL
7648 . 7792) (REMCATREL 7796 . 7915) (REMCATEQUAL 7919 . 8001) (REMCATREQ
8005 . 8140) (CATEQUALP 8144 . 8223) (REMCATEQUALP 8227 . 8311) (EQPCAT 8315
. 8460) (EQPCAT1 8464 . 8644) (CATAND 8648 . 8757) (CATEU 8761 . 9079) (CATFA
9083 . 9153) (REMCATFA 9157 . 9232) (FACAT 9236 . 9489) (CATOR 9493 . 9563)
(REMCATOR 9567 . 9642) (ORCAT 9646 . 9693) (ORCAT1 9697 . 9889) (ORCHECK 9893
. 10380) (CATNOT 10384 . 10474) (REMCATNOT 10478 . 10569) (CTXEV 10573 . 10923)
(CEVTV 10927 . 11084) (CEVTVN 11088 . 11233) (CEVREL 11237 . 11671) (CEVRR
11675 . 11773) (CEVARGS 11777 . 12004) (CEVRES 12008 . 12180) (TESTREL 12184
. 14442) (TESTRR 14446 . 14546) (SIGNUM 14550 . 14659) (PUTASSERT 14663 .
14759) (PUTGOAL 14763 . 14895) (PUTCLAUSE 14899 . 15478) (HISTFIX 15482 .
15765) (ADDCAT 15769 . 15826) (REMCLAUSE 15830 . 15880) (CLAUSEPUT 15884 .
16288) (CLAUSEREM 16292 . 16416) (PUTLB1 16420 . 18456) (LBPROD 18460 . 18810)
(LBSUM 18814 . 19371) (LBREL 19375 . 19617) (LBCONS 19621 . 20449) (REMLB2
20453 . 20838) (DIFFORM 20842 . 21288) (NUMCLAUSE 21292 . 21518) (REPCLAUSE
21522 . 21696))))))
STOP

```

(FILECREATED "23-MAY-74 13:00:45" VPF 21822

changes to: FOLSTEP, FOLSTEP, FOLEND, VPFVARS, VPFBLOCKS, AST, AST,  
FOLRETURN, FOLSTEP1

previous date: " 8-MAY-74 17:52:58"

(DEFINEQ

~~✓~~ COMS

(EEXP

```
[LAMBDA (X)
  (COND
    ((LISTP X)
     (CAR X))
    (T (PROG ((E (SEXP X)))
              (RETURN (COND
                       ((EQ (CAR E)
                            (QUOTE PROGN))
                        (CADR E))
                       (T E]))))
```

(ESUCC

```
[LAMBDA (X)
  (COND
    ((LISTP X)
     (OR (LISTP (SETQ X (CDR X)))
          (TSUCC X)))
    (T (PROG ((E (SEXP X)))
              (RETURN (COND
                       ((EQ (CAR E)
                            (QUOTE PROGN))
                        (ESUCC (CDR E)))
                       (T (TSUCC X))))))
```

(IFSUCC

```
[LAMBDA (A)
  (SETQ A (OR (TLS A)
              (TSUCC A)))
  (SELECTQ (CAR (SEXP A))
           (BLKELSE (TLS A))
           (ELSE (TSUCC A))
           ((ELSEIF BLKELSEIF)
            (IFSUCC A))
           A))
```

(EXITLABEL

```
[LAMBDA (A B)
  (PROG ((C A))
    LP [SETQ C (COND
              ((LISTP C)
               (CDR (LAST C)))
              ((NULL C)
               (SCANERROR1 (QUOTE illegal% EXIT)
                            A))
              ((EQ B (TLABEL C))
               (RETURN C))
              (T (TUP C]
          (GO LP]))
```

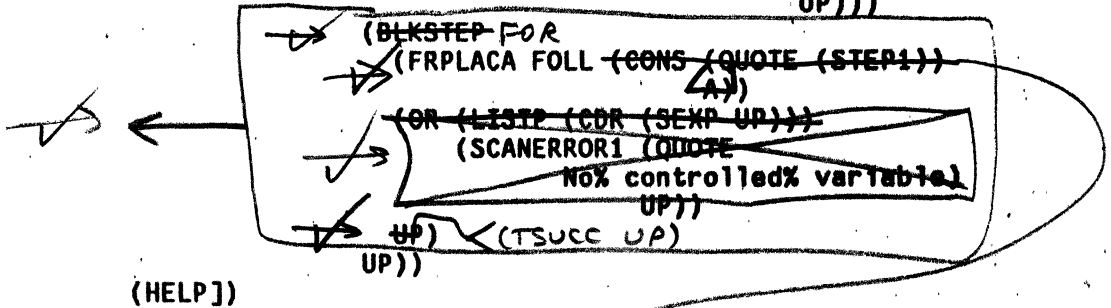




```

(FOLEND
 [LAMBDA (L A)
  (PROG ((UP (TUP A))
        S)
  (RETURN
   (FOLPATH
    (SELECTQ (CAR (SEXP UP))
              ((BLKIF BLKELSEIF)
               (IFSUCC UP))
              ((BLKELSE BEGIN)
               (TSUCC A))
              (LOOP (SELECTQ [CAR (EEXP (SETQ S (NBSUCC UP)
                                     (BLKUNTIL (SETQ FOLL (CONS S FOLL))
                                     (RETURN (SPLITFOL (TLS UP)
                                                         UP))))

```



```

(FOLIF
 [LAMBDA (L A)
  (SPLITFOL (APPEND (CDR L)
                    (GOMARK A (IFSUCC A)))
            (TSUCC A])

```

{ (GOMARK A (NBSUCC UP) (QUOTE STEP1))

```

(FOLWHILE
 [LAMBDA (L A)
  (SPLITFOL (APPEND (CDR L)
                    (GOMARK A (TUP A)))
            (TLS (TUP A)])

```

```

(GOMARK
 [LAMBDA (FROM TO)
  (CONS (CONS (QUOTE GO) TO) FROM])

```

```

(FOLGO
 [LAMBDA (L A)
  (SETQ FOLL (CDR FOLL))
  (FOLPATH L])
 (* Delete GO)

```

```

(FOLELSE
 [LAMBDA (L A)
  (FOLPATH L])

```

```

(FOLBLKIF
 [LAMBDA (L A)
  (SPLITFOL (TSUCC A)
            (TLS A])

```

```

(FOLBLKWHILE
 [LAMBDA (L A)
  (SPLITFOL (TSUCC A)
            (TLS (TUP A)])

```

```
(FOLUNTIL
 [LAMBDA (L A)
  (SETQ FOLL (CDR FOLL))
  (FOLPATH (APPEND (CDDR L)
                  (CONS (QUOTE (UNTIL1))
                        A]))]
```

```
(FOLUNTIL1
 [LAMBDA (L A)
  (FRPLACA FOLL (CDR A))
  (SPLITFOL (TLS (TUP A))
            (TUP A))]
```

```
(FOLBLKUNTIL
 [LAMBDA (L A)
  (SETQ FOLL (CDR FOLL))
  (FOLPATH (ESUCC A))]
```

```
(FOLEXIT
 [LAMBDA (L A)
  (PROG [(B (EXITLABEL A (CAR L)
                (RETURN (FOLPATH (SELECTQ (CAR (SEXP B))
                                           ((LOOP BEGIN)
                                            (TLS B))
                                           (BLKIF (IFSUCC B))
                                           (SCANERROR1 (QUOTE illegal% EXIT)
                                                       A)))]]
```

```
(FOLNEXT
 [LAMBDA (L A)
  (PROG [(B (EXITLABEL A (CAR L)
                (RETURN (FOLPATH (SELECTQ (CAR (SEXP B))
                                           (LOOP B)
                                           (SCANERROR1 (QUOTE illegal% NEXT)
                                                       A)))]]
```

```
(FOLSTOP
 [LAMBDA (L A)
  (LIST (REVERSE (CONS (CONS (QUOTE NOBIND)
                             (DREVERSE FOLL))
                       FOLL)))]]
```

```
(FOLASSERT
 [LAMBDA (L A)
  (PROG ((R (ASFOL A T)))
        (RETURN
         (LIST (REVERSE (CONS [CONS (QUOTE NOBIND)
                                   (DREVERSE (NCONC (CDR R)
                                                       (CDR FOLL))
             FOLL])))]]
```

```
(FOLLOOP
 [λ (L A)
  (LOOPFOL A
   '(WHILE BLKWHILE UNTIL
      BLKUNTIL BLKREPEAT)
   "'not legal between LOOP and
   WHILE"'])
```

```
(FOLFOR [λ (L A) (LOOPFOL A
 '(STEP BLKSTEP)
 "'not legal between FOR and BY/TO"'])
```

```
(FOLLOOP LOOPFOL
 [LAMBDA (L A) (CA OPS ERR)
  (PROG ((B (NBSUCC A)))
        (RETURN
```

```
(SELECTQ (CAR (EEXP B)) (COND [(FMEMB (CAR (EEXP B)) OPS)
                               [(WHILE BLKWHILE UNTIL BLKUNTIL BLKREPEAT)
                                (RETURN (FOLPATH (ESUCC A))
                                (T (SCANERROR
                                   (CAR (EEXP B))
                                   (QUOTE "not legal between LOOP and WHILE"
                                       A)))]])
  A))]
```

ERR

```
(NBSUCC
 [LAMBDA (A)
  (PROG ((B A))
    LP (COND
      ([PBEGP (CAR (EEXP (SETQ B (ESUCC B)
        (GO LP))))
      (RETURN B])
```

```
(FOLRETURN
 [LAMBDA (L A)
  (PROG ([B (TLAST (TPGN (AST A)
    B1)
    LP (COND
      ((EQ (CAR (SEXP B))
        (QUOTE ASSERT))
      (SETQ B (TPRED (SETQ B1 B)))
      (GO LP)))
    (RETURN (COND
      (B1 (FOLPATH B1))
      (T (FOLSTOP]))
```

```
(FOLSTEP
 [LAMBDA (L A)
  (FOLPATH (APPEND (CDDR L)
    (CONS (QUOTE (STEP1))
      (TSUCC A]))
```

*SPLITFOL* (handwritten arrow pointing to (APPEND (CDDR L) (CONS (QUOTE (STEP1)) (TSUCC A))))

*(GOMARK A A (QUOTE STEP1))* (handwritten arrow pointing to (QUOTE (STEP1)))

```
(FOLSTEP1
 [LAMBDA (L A)
  (FOLPATH (TSUCC (TUP (AST A))
```

```
(AST
 [LAMBDA (A)
  (COND
    ((NLISTP A)
      A)
    (T (CDR (LAST A]))
```

```
(PRINTPATH
 [LAMBDA (L)
  (PROG (LAST LM LD LP LPM LPD (LEFT (IPLUS (POSITION)
    2))
    (RIGHT (IPLUS (LINELENGTH)
      -8)))
    [MAPC L (FUNCTION (LAMBDA (M)
      (PPATHO (CDR M)
        (AND LAST (PPATH2 LAST LM))
```

```
(PPATHO
 [LAMBDA (P)
  (PROG ((L P))
    L1 (OR L (RETURN))
    (SELECTQ (CADR L)
      [(T F D)
        (PPATH1 (CAR L)
          (CAR (SETQ L (CDR L)
            (PPATH1 (CAR L)))
          (SETQ L (CDR L))
          (GO L1)])
```

(PPATH1  
[LAMBDA (A M)

(\* LAST is bound in  
PRINTPATH)

(PROG [(D (EQ M (QUOTE D))  
L1 (COND  
((LISTP A)  
(SETQ A (CDR A))  
(GO L1))  
((NULL LAST)  
(PPATH2 A M))  
((AND (EQ D LD)  
(EQ A LAST)))  
((OR (NEQ D LD)  
(AND M (NOT D))  
(NEQ A (TSUCC LAST)))  
(PPATH2 LAST LM)  
(PPATH2 A M)))  
(SETQ LAST A)  
(SETQ LM M)  
(SETQ LD D])

(PPATH2  
[LAMBDA (A M)

(\* LP, LEFT, RIGHT are  
bound in PRINTPATH)

(COND  
((COND  
((AND (EQ A LP)  
(EQ (EQ M (QUOTE D))  
LPD))  
NIL)  
((NULL LP))  
((OR (EQ LP LAST)  
(EQ A (TSUCC LP)))  
(PRINQ -))  
(T (PRINQ ...)))  
(COND  
((IGREATERP (POSITION)  
RIGHT)  
(TAB LEFT)))  
(PROG ((N (OR (TLNO A)  
A)))  
(SELECTQ M  
(D (PRINQ %[]  
(PRIN1 N)  
(PRINQ %]))  
(T (PRIN1 N)  
(PRINQ %()  
(PRINSTAT (CADR (EEXP A)))  
(PRINQ %)))  
(F (PRIN1 N)  
(PRINQ %()  
[PRINSTAT (NOTOF (CADR (EEXP A])  
(PRINQ %)))  
(PRIN1 N))  
(SETQ LP A)  
(SETQ LPD (EQ (SETQ LPM M)  
(QUOTE D]))

↓  
PRINSTAT →  
NPRIN2



```

(VERTPATHS
  [LAMBDA (P)
    (CLRHASH PMMA)
    [PROG ((INDEX 1))
      (MAPC
        (PATHL P)
        (FUNCTION (LAMBDA (Y)
          (PROG ((FIRST 1))
            [MAPC (CADR Y)
              (FUNCTION (LAMBDA (L)
                (PROG NIL
                  (SETQ L (CDR L))
                  LP (SELECTQ (CADR L)
                    ((T F D)
                     (VERTP1 (CAR L)
                               (CADR L))
                     (SETQ L (CDR L)))
                     (VERTP1 (CAR L)))
                    (COND
                     ((SETQ L (CDR L))
                      (GO LP]
                  (ADD1VAR INDEX]
            [MAPPM (FUNCTION (LAMBDA (X Y)
              (SETPMM X (DREVERSE Y)
                (MAPSTL (TSUCC P)
                  NIL
                  (FUNCTION PPM]))

```

```

(VERTP1
  [LAMBDA (X M)
    [COND
      ((LISTP X)
       (SETQ X (CDR (LAST X)
        (PROG ((L (PMM X)))
          [OR (EQ (CAAR L)
                INDEX)
            (SETPMM X (SETQ L (CONS (CONS INDEX 0)
                                     L]
          (FRPLACD (CAR L)
            (LOGOR (CDAR L)
              (COND
                ((PBEGP (CAR (SEXP X))) FIRST)
                (T (SETQ FIRST 2)
                   0))
              (SELECTQ M
                (D 4)
                (T 8)
                (F 16)
                0))

```



```

(PPST
[LAMBDA (A)
  (PROG ((B A)
        (L A)
        E
        (POS 1)
        (D -1)
        TX)
  [COND
    ((LISTP A)
     (SETQ B (CDR (LAST A))
     (SETQ E (SEXP B))
     (SETQ TX (TEXT B))
  [COND
    ((EQ L B)
     (SETQ L (CDR E)
     (SETQ POS (BALPOS (SELECTQ (CAR E)
                               ((IF
                                ELSEIF)
                                (QUOTE
                                 THEN))
                               (ELSE (QUOTE
                                     ELSE)))
                               (WHILE STEP)
                               (WHILE (QUOTE :))
                               (PROGN (SETQ D 0)
                                      (GO L1))
                               (RETURN (LISTST B)))
                               TX 1 T))
    (OR (EQ A B)
        (SETQ D 0))
    (* To account for 60
    element)
  L1 (LISTLCN B)
     (TABLEV)
     (PSTSN TX (IPLUS D (IDIFFERENCE (LENGTH E)
                                     (LENGTH L)))
              POS)
     (TERPRI])

```

```

(PSTSN
[LAMBDA (TX N POS)
  (PROG ((K N)
        (P1 (SUB1 POS))
        (P 1))
  LP (RETURN (COND
            ((ZEROP K)
             (PSTSEG TX P P1))
            ([SETQ P1 (BALPOS (QUOTE ;)
                              TX
                              (SETQ P (ADD1 P1))
                              (SUB1VAR K)
                              (GO LP))
             ((NEQ K 1)
              (SPACES P)
              (PRINQ ---))
             (T (PSTSEG TX P]))
    (* SOMETHING IS WRONG)

```

```

(BALPOS
[LAMBDA (X Y POS TAIL)
  (PROG ((L (FRPLACA (QUOTE (; %" X( %[ %] %)))
                X))
        (P POS)
        (LEV 0)
        P1)
  LP [COND
    ((NOT (SETQ P (STRPOS L Y P)))
     (RETURN NIL))
    (T (SELECTQ (NTHCHAR Y P)
                ((%[ %]
                 (ADD1VAR LEV))
                 ((%] %])
                 (COND
                  ((ZEROP LEV)
                   (RETURN NIL)))
                 (SUB1VAR LEV))
        [%" (SETQ P (STRPOS (QUOTE %")
                            Y
                            (ADD1 P]
                (COND
                 ((AND (ZEROP LEV)
                      (SETQ P1
                          (STRPOS X Y P NIL T TAIL)))
                  (RETURN P1]
    (ADD1VAR P)
    (GO LP])

```

```

(PSTSEG
[LAMBDA (TX POS POS1)
  (COND
   ((NEQ POS 1)
    (SETQ POS (STRPOS (QUOTE (% ) )
                      TX POS T))
     (SPACES (IPLUS POS -5))
     (PRINQ ...% )))
  (PRIN1 (SUBSTRING TX POS (OR POS1 -1)))
  (COND
   (POS1 (PRINQ % ...])
  )
(LISPXPRINT (QUOTE VPFNS)
            T)
(RPAQQ VPFNS
 (EEXP ESUCC IFSUCC EXITLABEL SPLITFOL FOLLOWPATHS FOLFROM ASFOL ASFIX
 FOLPATH FOLNIL FOLEND FOLIF FOLWHILE GOMARK FOLGO FOLELSE
 FOLBLKIF FOLBLKWHILE FOLUNTIL FOLUNTIL1 FOLBLKUNTIL FOLEXIT
 FOLNEXT FOLSTOP FOLASSERT FOLLOOP NBSUCC FOLRETURN FOLSTEP
 FOLSTEP1 AST PRINTPATH PPATH0 PPATH1 PPATH2 VERTPATHS VERTP1 PPM
 TEXTPATH PPST PSTSN BALPOS PSTSEG))
(LISPXPRINT (QUOTE VPFVARS)
            T)
(RPAQQ VPFVARS (PPMSG (IFPROP (FOLPATHFN PBEGP)
                              * CGOPS)
                 (PROP FOLPATHFN GO UNTIL1 STEP1)
                 (ENDUMP VPF)))

```

```
(RPAQQ PPMSGS ((0 % )
  (1 %[
  (2 % . %])
  (3 %[ . %])
  (5 %(
  (6 % . %))
  (7 %( . %))
  (8 % . T)
  (16 % . F)))
```

```
(RPAQQ CGOPS
  (SETQ END IF WHILE ELSE BLKIF BLKWHILE BLKELSE EXIT NEXT STOP START
  ASSERT DECLARE INCLUDE DECVAR DECMODE PROGN LOOP LEMMA ELSEIF
  BLKELSEIF BEGIN EMPTY CANCEL ASSUME BLKREPEAT UNTIL BLKUNTIL
  STEP BLKSTEP RETURN))
```

```
→ FOR
(DEFLIST(QUOTE(
  (END FOLEND)
  (IF FOLIF)
  (WHILE FOLWHILE)
  (ELSE FOLELSE)
  (BLKIF FOLBLKIF)
  (BLKWHILE FOLBLKWHILE)
  (EXIT FOLEXIT)
  (NEXT FOLNEXT)
  (STOP FOLSTOP)
  (ASSERT FOLASSERT)
  (DECLARE FOLASSERT)
  (PROGN HELP)
  (LOOP FOLLOOP)
  (ELSEIF FOLIF)
  (BLKELSEIF FOLBLKIF)
  (CANCEL FOLASSERT)
  (UNTIL FOLUNTIL)
  (BLKUNTIL FOLBLKUNTIL)
  (STEP FOLSTEP)
  (RETURN FOLRETURN)
)))(QUOTE FOLPATHFN))
```

→ (FOR FOLFOR)

→ (BLKSTEP FOLBLKWHILE)

```
(DEFLIST(QUOTE(
  (STOP T)
  (START T)
  (ASSERT T)
  (DECLARE T)
  (DECVAR T)
  (DECMODE T)
  (CANCEL T)
)))(QUOTE PBEGP))
```

```
(DEFLIST(QUOTE(
  (GO FOLGO)
  (UNTIL1 FOLUNTIL1)
  (STEP1 FOLSTEP1)
)))(QUOTE FOLPATHFN))
```

→ (STEPφ HELP)

[RPAQQ VPFBLOCKS

((FOLBLOCK FOLLOWPATHS FOLFROM IFSUCC EXITLABEL SPLITFOL ASFOL FOLPATH  
 FOLNIL FOLEND FOLIF FOLWHILE GOMARK FOLGO FOLELSE FOLBLKIF  
 FOLBLKWHILE FOLUNTIL FOLUNTIL1 FOLBLKUNTIL FOLEXIT FOLNEXT  
 FOLSTOP FOLASSERT FOLLOOP FOLRETURN FOLSTEP FOLSTEP1  
 (ENTRIES FOLLOWPATHS)  
 (BLKAPPLYFNS FOLNIL FOLEND FOLIF FOLWHILE FOLGO FOLELSE  
 FOLBLKIF FOLBLKWHILE FOLUNTIL FOLUNTIL1  
 FOLBLKUNTIL FOLEXIT FOLNEXT FOLSTOP FOLASSERT  
 FOLLOOP FOLRETURN FOLSTEP FOLSTEP1)  
 (LOCALFREEVARS FOLL FOLL1))  
 (PRINTPATH PRINTPATH PPATH0 PPATH1 PPATH2  
 (LOCALFREEVARS LAST LM LD LP LPM LPD LEFT RIGHT))  
 (VERTPATHS VERTPATHS VERTP1 PPM (LOCALFREEVARS INDEX FIRST])

FOLFOR LOOPFOL NBSUCC

FOLFOR

[RPAQQ VPFPROPS ((PMM (MACRO MAP)

HASH  
 (SETHASHQ PMMA 50 1.5))  
 (PPMSG (MACRO NOSET)  
 ASSOC PPMMSG)  
 (PATHL)  
 (FOLPATHFN (NOSET MACRO))  
 (PBEGP (NOSET MACRO])

(DECLARE

(BLOCK: NIL EEXP ESUCC ASFIX NBSUCC AST TEXTPATH PPST PSTSN BALPOS PSTSEG)  
 (BLOCK: FOLBLOCK FOLLOWPATHS FOLFROM IFSUCC EXITLABEL SPLITFOL ASFOL FOLPATH  
 FOLNIL FOLEND FOLIF FOLWHILE GOMARK FOLGO FOLELSE FOLBLKIF  
 FOLBLKWHILE FOLUNTIL FOLUNTIL1 FOLBLKUNTIL FOLEXIT FOLNEXT FOLSTOP  
 FOLASSERT FOLLOOP FOLRETURN FOLSTEP FOLSTEP1 (ENTRIES FOLLOWPATHS)  
 (BLKAPPLYFNS FOLNIL FOLEND FOLIF FOLWHILE FOLGO FOLELSE FOLBLKIF  
 FOLBLKWHILE FOLUNTIL FOLUNTIL1 FOLBLKUNTIL FOLEXIT  
 FOLNEXT FOLSTOP FOLASSERT FOLLOOP FOLRETURN FOLSTEP  
 FOLSTEP1)

(LOCALFREEVARS FOLL FOLL1))  
 (BLOCK: PRINTPATH PRINTPATH PPATH0 PPATH1 PPATH2  
 (LOCALFREEVARS LAST LM LD LP LPM LPD LEFT RIGHT))  
 (BLOCK: VERTPATHS VERTPATHS VERTP1 PPM (LOCALFREEVARS INDEX FIRST))

) (APPLYMAPC VPFPROPS (FUNCTION NEWPROP))

(ADDTovar GLOBALVARS PMMA)

(PROGN (QUOTE JUSTEVALUATE)

(FILEMAP (NIL (194 18306 (EEXP 206 . 490) (ESUCC 494 . 838) (IFSUCC 842 .  
 1090) (EXITLABEL 1094 . 1442) (SPLITFOL 1446 . 2118) (FOLLOWPATHS 2122 . 2668)  
 (FOLFROM 2672 . 2884) (ASFOL 2888 . 4300) (ASFIX 4304 . 4600) (FOLPATH 4604  
 . 5215) (FOLNIL 5219 . 5268) (FOLEND 5272 . 5875) (FOLIF 5879 . 6005) (FOLWHILE  
 6009 . 6138) (GOMARK 6142 . 6209) (FOLGO 6213 . 6317) (FOLELSE 6321 . 6364)  
 (FOLBLKIF 6368 . 6443) (FOLBLKWHILE 6447 . 6530) (FOLUNTIL 6534 . 6696) (  
 FOLUNTIL1 6700 . 6808) (FOLBLKUNTIL 6812 . 6894) (FOLEXIT 6898 . 7290) (FOLNEXT  
 7294 . 7577) (FOLSTOP 7581 . 7736) (FOLASSERT 7740 . 8045) (FOLLOOP 8049 .  
 8441) (NBSUCC 8445 . 8601) (FOLRETURN 8605 . 8866) (FOLSTEP 8870 . 8967) (  
 FOLSTEP1 8971 . 9032) (AST 9036 . 9115) (PRINTPATH 9119 . 9451) (PPATH0 9455  
 . 9756) (PPATH1 9760 . 10379) (PPATH2 10383 . 11596) (VERTPATHS 11600 . 12688)  
 (VERTP1 12692 . 13396) (PPM 13400 . 13972) (TEXTPATH 13976 . 15136) (PPST  
 15140 . 16426) (PSTSN 16430 . 17017) (BALPOS 17021 . 18014) (PSTSEG 18018  
 . 18303))))))

STOP

(FILECREATED "30-AUG-74 08:42:56" VQM.;22 10498

changes to: QMPLUS VQMCOMS

previous date: "28-AUG-74 08:21:03" VQM.;21)

```

(LISPXPRINT (QUOTE VQMCOMS)
  T)
(RPAQQ VQMCOMS
  ((PROP QMATCHFN PLUS TIMES QUOTIENT FA EX FU LEQ GEQ QUOTE EQUALP
    NEQUALP AND OR EQV NEQV)
  (PROP FAMOPS LEQ GEQ)
  (ADDVARS (GLOBALVARS RANGEOFFSET SUMSIZE))
  (ENDUMP VQM)))
(DEFLIST(QUOTE(
  (PLUS (QMPLUS T))
  (TIMES (QMTIMES))
  (QUOTIENT (QMQUOTIENT T))
  (FA (QMQ))
  (EX (QMQ))
  (FU (QMQ))
  (LEQ (QMORD T))
  (GEQ (QMORD T))
  (QUOTE (QMCON))
  (EQUALP (QMEQ))
  (NEQUALP (QMEQ))
  (AND (QMANY T))
  (OR (QMANY T))
  (EQV (QMEQ))
  (NEQV (QMEQ))
)) (QUOTE QMATCHFN))
(DEFLIST(QUOTE(
  (LEQ (LEQ GEQ))
  (GEQ (GEQ LEQ))
)) (QUOTE FAMOPS))
(ADDTOVAR GLOBALVARS RANGEOFFSET SUMSIZE)
[RPAQQ VQMPPROPS ((FAMOPS (NOSET MACRO))
  (QTL (MACRO SETMACRO)
    HASH
    (SETHASHQ QTLA 100 2.0))
  (QMATCHFN (NOSET MACRO))
  (RPAQQ VQMFNS
    (QMATCH QMAT QMBIND QMSIZE QMCON QML QMLIST QMATL QMQ QMQUOTIENT
      QMQUOT1 QMPLUS QMTIMES QMPERM QMORD QMEQ QMEQ1 QMANY MATCHLIS
      MATCHLIS1 BVPAIR))
  (RPAQQ VQMVARS NIL)
  [RPAQQ VQMBLOCKS
    ((QMBLOCK QMATCH QMAT QMATL QMBIND QMCON QML QMLIST QMQ QMQUOTIENT
      QMQUOT1 QMPLUS QMTIMES QMPERM QMORD QMEQ QMEQ1 QMANY
      (ENTRIES QMATCH)
      (SPECVARS EVALARGS)
      (BLKAPPLYFNS QMCON QMQ QMQUOTIENT QMPLUS QMTIMES QMORD QMEQ
        QMANY)
      (LOCALFREEVARS EFFORT))
    (QMSIZE)
    (MATCHLIS MATCHLIS MATCHLIS1 (LOCALFREEVARS OPS)
      (APPLYMAPC VQMPPROPS (FUNCTION NEWPROP))
    ]
  [DECLARE: EVAL@COMPILE
    (ADDTOVAR GLOBALVARS QTLA)
  ]
  (DEFINEQ
    (QMATCH
      [LAMBDA (FORM EXP ALIST EFFORT)
        ([LAMBDA (EVALARGS)
          (QMAT FORM EXP ALIST)
          NIL])

```

```

(QMAT
  [LAMBDA (FORM EXP ALIST)
    (COND
      [(LISTP FORM)
        (PROG [(FN (QMATCHFN (CAR FORM)
          (RETURN (COND
            ((NULL FN)
              (AND (LISTP EXP)
                (EQ (CAR FORM)
                  (CAR EXP)))
              (QMLIST (CDR FORM)
                (CDR EXP)
                  ALIST)))
            ([OR (CADR FN)
              (AND (LISTP EXP)
                (EQ (CAR FORM)
                  (CAR EXP]
                (BLKAPPLY* (CAR FN)
                  FORM EXP ALIST])
          ]
        ]
      [(LITATOM FORM)
        (PROG [(D (FASSOC FORM (CAR ALIST)
          (RETURN (COND
            ((NULL D)
              (QMCN FORM EXP ALIST))
            [(EQ (CDR D)
              (QUOTE NOBIND))
              (LIST (QMBIND FORM EXP ALIST (CDR ALIST)
                ((EQ (CDR D)
                  EXP)
                  (LIST ALIST]
                (T (QMCN FORM EXP ALIST])
          ]
        ]
      ]
    ]
  ]
(QMBIND
  [LAMBDA (VAR EXP ALIST SCORE)
    (CONS (CONS (CONS VAR EXP)
      (CAR ALIST))
      (COND
        ((NLISTP EXP)
          SCORE)
        (T (IDIFFERENCE SCORE (QMSIZE EXP)]
      ]
    ]
  ]
(QMSIZE
  [LAMBDA (EXP)
    (COND
      ((NLISTP EXP)
        0)
      (T (PROG ((S 1)
        [MAPC (CDR EXP)
          (FUNCTION (LAMBDA (X)
            (SETQ S (IPLUS S (QMSIZE X)
              1]
          ]
        (RETURN (COND
          ((AND (EQ (CAR EXP)
            (QUOTE PLUS))
            (CDDDR EXP))
            (IPLUS S SUMSIZE))
          (T S])
        ]
      ]
    ]
  ]
(QMCN
  [LAMBDA (FORM EXP ALIST)
    (AND (EQ FORM EXP)
      (LIST (CONS (CAR ALIST)
        (IPLUS 10 (CDR ALIST])
      ]
    ]
  ]
(QML
  [LAMBDA (FORM EXP ALIST)
    (QMLIST (CDR FORM)
      (CDR EXP)
      ALIST])
  ]

```



```

(QMLIST
[LAMBDA (FL EL LL)
(COND
[FL (AND EL (SETQ LL (QMAT (CAR FL)
(CAR EL)
LL))
(PROG NIL
L1 (RETURN (COND
((SETQ FL (CDR FL))
(AND (SETQ EL (CDR EL))
(SETQ LL (QMATL (CAR FL)
(CAR EL)
LL))
(GO L1)))
((CDR EL)
NIL)
(T LL]
(EL NIL)
(T (LIST LL])
(QMATL
[LAMBDA (FORM EXP LL)
(AND LL (MAPCONC LL (FUNCTION (LAMBDA (X)
(QMAT FORM EXP X])
(QMQ
[LAMBDA (FORM EXP ALIST)
(PROG ((AL (CAR ALIST))
(FL (CADR FORM))
(EL (CADR EXP)))
LP (RETURN (COND
[(SETQ FL (CDR FL))
(AND (SETQ EL (CDR EL))
(EQ (CDDAR FL)
(CDDAR EL))
(PROGN (SETQ AL (CONS (CONS (CADAR FL)
(CADAR EL))
AL))
(GO LP]
((CDR EL)
NIL)
(T (QMLIST (CDDR FORM)
(CDDR EXP)
(CONS AL (CDR ALIST]))
(QMQUOTIENT
[LAMBDA (FORM EXP ALIST)
(NCONC (AND (EQCAR EXP (QUOTE QUOTIENT))
(QML FORM EXP ALIST))
(AND (LITATOM (CADR FORM))
(NUMBERP (CADDR FORM))
(PROG ((D (FASSOC (CADR FORM)
(CAR ALIST)))
(K (CADDR FORM)))
(RETURN (COND
((NULL D)
NIL)
((EQ (CDR D)
(QUOTE NOBIND))
(AND (GREATERP K 0)
(QMQUOT1 (CAR D)
(MAKETIMES K EXP)
ALIST
(IPLUS -2 (CDR ALIST))
K)))
((EQ (MAKEQUOTIENT (CDR D)
K)
EXP)
(LIST (CONS (CAR ALIST)
(IPLUS 15 (CDR ALIST]))

```



```

      ([AND D1
        (EQ EXP (SETQ X
          (COND
            [NEG1 (MAKENEG (MKPLUS (CONS C R]
              (T (MKPLUS (CONS C R]
                (LIST (CONS (CAR ALIST)
                  (IPLUS 5 (QMSIZE X)
                    (CDR ALIST]
          (AND (EQ (CAR EXP)
            (QUOTE PLUS))
            (EQP C (CADR EXP))
            (QMPERM L (CDDR EXP)
              ALIST]))
(QMTIMES
  [LAMBDA (FORM EXP ALIST)
    (AND (EQP (CADR FORM)
      (CADR EXP))
      (QMPERM (CDDR FORM)
        (CDDR EXP)
        ALIST]))
(QMPERM
  [LAMBDA (FL EL ALIST)
    (AND (EQ (FLENGTH FL)
      (FLENGTH EL))
      (COND
        ((NULL (CDR EL))
          (QMAT (CAR FL)
            (CAR EL)
            ALIST))
        [(NULL (CDDR EL))
          (NCONC (QMLIST FL EL ALIST)
            (QMATL (CAR FL)
              (CADR EL)
              (QMAT (CADR FL)
                (CAR EL)
                ALIST]
          (T
            (QMLIST FL EL ALIST])
    (* Too complicated, punt)
(QMORD
  [LAMBDA (FORM EXP ALIST)
    (COND
      ((EQ (CAR FORM)
        (CAR EXP))
        (QMAT (CADR FORM)
          (CADR EXP)
          ALIST))
      ((EQ (CAR EXP)
        (SELECTQ (CAR FORM)
          (GEQ (QUOTE LEQ))
          (LEQ (QUOTE GEQ))
          (HELP)))
        (QMAT (NEGOF (CADR FORM))
          (CADR EXP)
          ALIST]))
(QMEQ
  [LAMBDA (FORM EXP ALIST)
    (NCONC (QMEQ1 (CADR FORM)
      (CADDR FORM)
      (CADR EXP)
      (CADDR EXP)
      ALIST)
      (QMEQ1 (CADR FORM)
        (CADDR FORM)
        (CADR EXP)
        (CADR EXP)
        ALIST]))

```

```

(QMEQ1
[LAMBDA (A1 B1 A2 B2 ALIST)
  (PROG ((LA (QMAT A1 A2 ALIST))
        (LB (QMAT B1 B2 ALIST)))
    (RETURN (COND
              ((NULL LA)
               LB)
              ((NULL LB)
               LA)
              (T (SETQ LA (NCONC (QMATL B1 B2 LA)
                                LA))
                  (NCONC (DREML LA (NCONC (QMATL A1 A2 LB)
                                          LB))
                          LA]))))

```

```

(QMANY
[LAMBDA (FORM EXP ALIST)
  (COND
    ((AND (LISTP EXP)
          (EQ (CAR EXP)
              (CAR FORM))))
    (QMPERM (CDR FORM)
            (CDR EXP)
            ALIST))
    (T (MAPCONC (CDR FORM)
                (FUNCTION (LAMBDA (X)
                           (QMAT X EXP ALIST)))))

```

```

(MATCHLIS
[LAMBDA (FORM)
  (OR (QTL FORM)
      (PROG (OPS L)
            (SETQ L (MATCHLIS1 (NOTOF FORM)
                              (QUOTE (NIL . 0))
                              T))
            (SETQTL FORM (SETQ L (CONS OPS L)))
            (RETURN L]))
  (* FORM is (FA --))

```

```

(MATCHLIS1
[LAMBDA (FORM AL BODY)
  (SELECTQ (CAR FORM)
    [(AND OR)
     (MAPCONC (CDR FORM)
              (FUNCTION (LAMBDA (X)
                         (MATCHLIS1 X AL BODY)))]
    [(EX FU)
     (PROG ((AL1 (NCONC (BVPAIR (CDADR FORM)
                              (CAR AL)))
                (D (CDR AL))))
           (RETURN (NCONC (MATCHLIS1 (CADDR FORM)
                                     (CONS AL1 (IPLUS D RANGEOFFSET))
                                     NIL)
                          (MATCHLIS1 (CADDR FORM)
                                     (CONS AL1 D)
                                     BODY)))]
    (PROG ((Y (FAMOPS (CAR FORM)))
          (OP (CAR FORM)))
          [OR (EQ OPS T)
             (SETQ OPS (SELECTQ Y
                               (T T)
                               (NIL (ADDELT OP OPS))
                               (UNION Y OPS))
             (RETURN (LIST (CONS FORM (CONS AL BODY))

```

```

(BVPAIR
 [LAMBDA (L)
  (MAPCAR L (FUNCTION (LAMBDA (V)
    (CONS (CADR V)
          (QUOTE NOBIND]))
    )
  )
)
(DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
 (BLOCK: NIL BVPAIR)
 (BLOCK: QMBLOCK QMATCH QMAT QMATL QMBIND QMCON QML QMLIST QMQ QMQUOTIENT
  QMQUOT1 QMPLUS QMTIMES QMPERM QMORD QMEQ QMEQ1 QMANY (ENTRIES QMATCH)
  (SPECVARS EVALARGS)
  (BLKAPPLYFNS QMCON QMQ QMQUOTIENT QMPLUS QMTIMES QMORD QMEQ QMANY)
  (LOCALFREEVARS EFFORT))
 (BLOCK: QMSIZE)
 (BLOCK: MATCHLIS MATCHLIS MATCHLIS1 (LOCALFREEVARS OPS))
]
(DECLARE: DONTCOPY
 (FILEMAP (NIL (1649 9525 (QMATCH 1661 . 1765) (QMAT 1769 . 2485) (QMBIND 2489
 . 2664) (QMSIZE 2668 . 2982) (QMCON 2986 . 3123) (QML 3127 . 3227) (QMLIST
 3231 . 3626) (QMATL 3630 . 3732) (QMQ 3736 . 4187) (QMQUOTIENT 4191 . 4797)
 (QMQUOT1 4801 . 4980) (QMPLUS 4984 . 6690) (QMTIMES 6694 . 6827) (QMPERM 6831
 . 7240) (QMORD 7244 . 7657) (QMEQ 7661 . 7876) (QMEQ1 7880 . 8214) (QMANY
 8218 . 8455) (MATCHLIS 8459 . 8711) (MATCHLIS1 8715 . 9416) (BVPAIR 9420 .
 9522))))))
STOP

```

(FILECREATED "14-SEP-73 19:18:52" VTL)

(DEFINEQ

~~(VTLSET  
[LAMBDA NIL  
(SETQ BREAK NIL)  
(SETQ BREAKCTX NIL)  
(SETQ INTERRUPTFLAG NIL)  
(TLSET])~~

~~COMS~~

~~(TLSET  
[LAMBDA NIL (\* Called by VXLSET)  
(CAR (SETQ TCOMDESC (CONS (QUOTE \$)  
(KWSET (QUOTE (LISTQ TCOMQ))~~

(INTERTEST  
[LAMBDA NIL  
(COND  
((AND (EQ INTYPE 1)  
(EQ (CAR (QUOTE INTERRUPTFLAG))  
T))  
(FRPLACA (QUOTE INTERRUPTFLAG)  
(QUOTE INTERRUPT))  
(FRPLACA (QUOTE BREAK)  
T)  
(SETQ INTYPE -1)  
(DISSMISS 1000)  
(PRIN1 (CHARACTER 7)  
T))

(\* Causes INTERRUPT to resume)

(INTERSETQ  
[NLAMBDA (INTERSX)  
(SETQ BREAKCTX NIL)  
(RESETVAR INTERRUPTFLAG T (EVAL INTERSX))

(DOPROOF  
[LAMBDA (CTX MODE)  
(PROG ((VAL (PROOFDONE CTX)))  
(TERPRI)  
(PRINQ Proof% for% )  
(PCTX CTX)  
(SPACES 1)  
(PJUST CTX)  
[COND  
((CAR VAL)  
(TERPRI)  
(PCEFF CTX))  
(T (KILLSONS CTX)  
(PRINTQ :)  
(PROG ((C (NEWCTX CTX (QUOTE JPROOF)  
NIL)))  
(OR BREAK (SETQ BREAK (CAR MODE)))  
(TOPPROOF C (CDR MODE))  
(PCEFF C]  
(RETURN VAL])

(TOPPROOF

[LAMBDA (CTX CXFLAG)  
(MAP (OR CTX (SETQ CTX (CURCTX)))  
(FUNCTION PROOFSTART))

(KILLSONS CTX)  
(INTERSETQ (PROOFDO (LIST CTX)  
CXFLAG])

(FIXPRI TPLIST)  
(FIXPRI CXLIST)

(ENDJPROOF

[LAMBDA (CTX VAL)  
(PPRET CTX VAL)  
(SETTHCLAUSEP T VAL (CDR CTX))  
(RTFRM TOPPROOF VAL)]

(PROOFDO

[LAMBDA (THLIST CXFLAG)  
(PROG (C1 CTX)  
L1 (SETCTX (SETQ CTX (CAR THLIST)))  
(COND

(BREAK (TLEXEC CTX))  
((PROOFSTEP TPLIST CTX))  
[C1 (COND  
((PROOFSTEP CXLIST CTX))  
((PROOFSTEP (QUOTE ((-)))  
CTX)  
(FAILURE CTX))  
(T (COUNTERSUCCESS CTX)  
[CXFLAG (OR [SETQ C1 (SOME CTX  
(FUNCTION (LAMBDA (X Y)  
(EQ (CAR (CJUST Y))  
(QUOTE JCX])  
(RPLACA THLIST (NEWCTX CTX (QUOTE JCX])  
(T (FAILURE CTX)))  
(GO L1)]

(FIXPRI  
[X (L)  
(SORT L (FXY  
(LEQ (PSPRI X)  
(PSPRI Y)])

(PROOFSTEP

[LAMBDA (L CTX) ~~P V~~  
(PROG [M ~~M2~~ MATCHLIST MATCHVAR1 MATCHVAR2  
(HEAD (LIST (LIST (QUOTE CLAUSE]  
L2 (SETQ M (CAR L)) ~~(SETQ P (PSPROP \*))~~  
(SETQ MATCHLIST (SETQ MATCHVAR1 (SETQ MATCHVAR2 NIL)))  
(COND

([OR [COND  
[~~(CADR (SETQ M2 (CDDR M)))~~ ~~(SETQ V (PSVAL M))~~  
(SOME (GETX ~~(CADR M2)~~ ~~V~~  
~~(CAR M2)~~ ~~P~~  
CTX)  
(FUNCTION (LAMBDA (X)  
(SOMEPROOF X M HEAD]  
(T (NOT (TRY (MAPPROP (OR ~~(CAR M2)~~ ~~P~~  
(QUOTE THCLAUSEP))  
CTX  
(FUNCTION (LAMBDA (X)  
(AND (SOMEPROOF X M HEAD)  
(FAIL])  
(AND MATCHLIST ~~(CAR (SETQ M2 (CDDR M2)))~~ ~~(SETQ P (PSTEST M))~~  
(TRY (APPLY\* ~~(CAR M2)~~ ~~P~~  
MATCHLIST]

(RETURN L))  
((SETQ L (CDR L))  
(GO L2))  
(T (RETURN NIL]))

```

(SOMEPROOF
→ [LAMBDA (X M H)
  (FRPLACD (CAR H)
    X)
→ (SOMEPROOF1 (PATINT (CAR L) (PSPAT M)
  X)
  → (CADAR L) (PSEXP M)
    H])

```

```

(SOMEPROOF1
  [LAMBDA (M E H)
    (SOME M (FUNCTION (LAMBDA (Z)
      (TRY (EVALA E (FRPLACD H Z))

```

```

(PROOFEX
  [LAMBDA (FN ARGS CTX)
    [COND
      (CTX (SETCTX CTX))
      (T (SETQ CTX (CURCTX)
        (SETCPH CTX (CONS (CONS FN ARGS)
          (CPH CTX))))
      (APPLY FN ARGS])

```

```

(PROOFDONE
  [LAMBDA (CTX)
    (THCLAUSEP T CTX)]

```

```

(PROOFSTART
  [LAMBDA (CTX)
    (REMTHTHCLAUSEP T CTX)]

```

```

(SUCCESS
  [LAMBDA (E CTX)
    (PROOFEND CTX (CONS T CTX))]

```

```

(COUNTERSUCCESS
  [LAMBDA (CTX)
    (PROOFEND CTX (CONS (QUOTE CX)
      CTX))]

```

```

(FAILURE
  [LAMBDA (CTX)
    (PROOFEND CTX (CONS NIL CTX))]

```

```

(PROOFEND
  [LAMBDA (CTX VAL)
    (PROG [(FN (JENDFN (CAR (CJUST CTX)
      (SETHTHCLAUSEP T VAL CTX)
→ [SETQ THLIST (DFILTER THLIST (F/X (NOT (FTAILP CTX X)
      (RETURN VAL))]

```

```

(ENDNULL
  [LAMBDA (CTX VAL)
    (PROOFEND (CDR CTX)
      VAL)]
  [COND
    (FN (APPLY* FN CTX VAL))
    (T (HELP]

```



```

(TLEXEC
 [LAMBDA (CTX)
 (OR CTX (SETQ CTX (CURCTX)))
 (COND
 ((OR (NEQ BREAK T)
 (NEQ CTX BREAKCTX))
 (PRINQ %(in% )
 (PCTX (SETQ BREAKCTX CTX))
 (PRINTQ %)))
 (SETQ BREAK NIL)
 (SETQ INTERRUPTFLAG T)
 (OR (EXECLP TCOMDESC T)
 (ERROR!]))

```

```

(TLCFIX
 [λ NIL
 (COND
 ((NEQ CTX (CAR THLIST))
 (SETQ BREAK T)
 (RTERM TLEXEC T]))

```

```

(TLPXQ
 [NLAMBDA (X)
 (PROOFEX (CAR X)
 (MAPCAR (CDR X)
 (FUNCTION EVAL))
 CTX))

```

```

(TLCASES
 [LAMBDA (E)
 (COND
 ((NEQCAR E (QUOTE OR))
 (PRINTQ "Not a disjunction")
 (T (FACBOOL E))

```

```

(TLEXPAND
 [LAMBDA (E)
 (OR (TRY (SFSUBST E))
 (PRINTQ "Nothing to expand"]])

```

```

(TLNEXT
 [LAMBDA NIL
 (OR (PROOFSTEP TPLIST CTX)
 (PRINTQ "No progress"])

```

```

(TLRUN
 [λ NIL
 (PROG NIL
 LP (COND
 ((NOT (PROOFSTEP TPLIST CTX))
 (PRINTQ "No progress"))
 ((EQ (CAR THLIST) CTX)
 (GO LP))

```

```

(TLQUIT
 [LAMBDA NIL
 (FAILURE (EVALV (QUOTE CTX)
 (STKPOS (QUOTE TOPPROOF))

```

```

(TLTEST
 [LAMBDA NIL
 (PROG [(C [NEWCTX CTX (QUOTE JUSER)]
 (SETQ BREAK T)
 (SETQ THLIST (CONS THLIST))
 (RTERM TLEXEC T])

```

```

(ENDJUSER
 [LAMBDA (CTX VAL)
 (PPRET CTX VAL)
 (ERROR!))

```

```

(TLEVAL
 [LAMBDA (E)
 (PRINTEXP (CTXEV E CTX))

```

← NPRINT

(TLADD  
[LAMBDA (E)  
→ (PUTCLAUSE E CTX)]

← 'USER

(TLDELETE  
[LAMBDA (E)  
(REMCLAUSE E CTX)]

(TLINST  
[LAMBDA (E LIS)  
(PUTCLAUSE (FSUBMAKE LIS E T)  
→ CTX)]

← 'USER

(TLASGN  
[LAMBDA (LIS)  
(PROG [(V (MAPCAR LIS (FUNCTION CAR)  
(EQSUB1 NIL V LIS))

(CANSUBS  
[LAMBDA (LIS)  
(MAPCAR LIS (FUNCTION (LAMBDA (X)  
(CONS (CAR X)  
(TLC (CDR X))

(PTLSUBS  
[LAMBDA (LIS)  
(MAPRINT LIS NIL NIL NIL (QUOTE " " )  
(F/X (PRINEXP (CAR X)  
(PRINQ =)  
(PRINSTAT (CDR X])

4  
NPRIN2

(TMLATCH  
[LAMBDA (E Q)  
(COND  
((NEQCAR Q (QUOTE FA))  
(PRINTQ "Second clause is not quantified"))  
(T (PROG ((L (TMLAT E Q))

(J 1)  
K A)  
(COND  
((NULL L)  
(PRINTQ "No matches")  
(RETURN))  
→ ((NULL (CDR L))  
~~(TMPUT (CAR L))~~  
(RETURN))  
(PRIN2 (SETQ K (LENGTH L)))  
(PRINTQ " matches:")  
[MAPC L (FUNCTION (LAMBDA (Z)  
(PRIN2 J)  
(PRINQ " scope=")  
~~(PRINT (IPLUS (CDR Z)  
(CDR Z)))~~  
~~(DAL (ALCUT (CADR Z)))~~  
(ADDIVAR J)

(TMP (CAR L))  
(PRINQ "Add?")  
(SELECTQ (READ)  
((NIL N NO))  
(TMPUT (CAR L)))

← (TMP Z)

```

LP (PRINQ "Add which one(s)? ")
  (SETQ A (NCONS (READ)
                (READLINE)))
(COND
  ([OR (NULL A)
        (EQUAL A (QUOTE (ALL)
                    (MAPC L (FUNCTION TLMPUT))))
        [[EVERY A (FUNCTION (LAMBDA (X)
                            (AND (FIXP X)
                                  (GEQ X 1)
                                  (LEQ X K)
                                  (MAPC A (FUNCTION (LAMBDA (X)
                                                    (TLMPUT (CAR (FNTH L X)
                                                            (GO LP]))
                                                            (PRINTQ ?)
                                                            (GO LP]))
                                                    (PRINTQ ?)
                                                    (GO LP]))
                                  (GO LP]))
        ]])

```

```

(TLPM
 [λ (z)
 (PRINQ "score_n")
 (PRINT (IPLUS (CDAR z)
               (CDDR z)))
 (PAL (ALCUT (CADR z]))

```

```

(TLMAT
 [LAMBDA (E Q)

```

(\* FAM2 uses Q, MIN, MAX free)

```

  (PROG ((MIN -1000)
         (MAX 1000))
    (RETURN (FAM2 (LIST E)
                  [CDR (MATCHLIS Q)]))

```

```

(TLMPUT
 [LAMBDA (Z)
  (FAMPUT (CDAAR Z)
          (CAAAR Z)
          (ALCUT (CADR Z))
          NIL)]

```

```

(TLC
 [LAMBDA (E)
  (PROG ((ASSUMED NIL) (MDCHECK E)
          (SETQ E (CANON E)) (CANON E)
          (CKAS ASSUMED)
          (RETURN E)))

```

```

(LISPXPRT (QUOTE VTLFNS)
  T)

```

```

(RPAQQ VTLFNS
 (VTLSET TLSET INTERTEST INTERSETQ DOPROOF TOPPROOF ENDJPROOF
         PROOFDO PROOFSTEP SOMEPROOF SOMEPROOF1 PROOFEX
         PROOFDONE PROOFSTART SUCCESS COUNTERSUCCESS FAILURE
         PROOFEND ENNULL TLEXEC TLPXQ TLCASES TLEXPAND TLNEXT
         TLQUIT TLTEST ENDJUSER TLEVAL TLADD TLDELETE TLINST
         TLAGN CANSUBS PTLSUBS TLMATCH TLMAT TLMPUT TLC))

```

```

(LISPXPRT (QUOTE VTLVARS)
  T)

```

```

(RPAQQ VTLVARS ((PROP (JFORM JENDFN)
                      JUSER JPROOF)
                (PROP HFORM TLCASES TLEXPAND TLADD TLDELETE TLINST TLSET
                      TLMATCH)
                (PROP (PARSEDEF SENAME)
                      TCOMQ CLDES CANREL SUBS ONESUB)
                (ADDVARS (GLOBALVARS INTERRUPTFLAG BREAK) EVAL@COMPILE
                          (BREAKMACROS (DEFER (RPAQQ BREAK T)
                                              OK))))

```

```

(ENDDUMP VTL)))
(DEFLIST(QUOTE(
  (JUSER ("for testing"))
  (JPROOF ("for proof"))
  ))(QUOTE JFORM))

```

```

(BREAK NIL)
(BREAKCTX NIL)
(INTERRUPTFLAG NIL)

```

```
(DEFLIST(QUOTE(
  (JUSER ENDJUSER)
  (JPROOF ENDJPROOF)
))(QUOTE JENDFN))
```

```
(DEFLIST(QUOTE(
  (TLCASES ("CASES " PCN))
  (TLEXPAND ("EXPAND " PCN))
  (TLADD ("ADD " PCN PCN)) ← NPRIN2
  (TLDFLETE ("CANCEL " PCN))
  (TLINST ("INST " PCN " " PTLSUBS))
  (TLSET ("SET " PTLSUBS))
  (TLMATCH ("MATCH." PCN " " PCN))
))(QUOTE HFORM))
```

```
(DEFLIST(QUOTE(
  [TCOMQ ([ (RUN (: (TLRUN)))
    (NEXT (: (TLNEXT))) ←
    (QUIT (: (TLQUIT))) ←
    (TEST (: (TLTEST)))
    (EVAL (CANREL)
      (: TLEVAL 1))
    ((EITHER (CASES (CLDES)
      (: TLCASES 1))
      (EXPAND (CLDES)
        (: TLEXPAND 1))
      (ADD (CANREL)
        (: TLADD 1))
      (CANCEL (CLDES)
        (: TLDELETE 1))
      (INST (CLDES)
        (SUBS)
        (: TLINST 2))
      (SET (SUBS)
        (: TLASGN 1))
      (MATCH (CLDES)
        (CLDES)
        (: TLMATCH 2)))
      (: TLPXQ 1] (: (TLCEIX)) (: PROG2)
    (CLDES ($NUM % . $NUM (: GETNCLAUSE 2)))
    (CANREL ((QUREL)
      (: QUOTE 1)
      (: TLC 1)))
    (SUBS ((ONESUB)
      ($ 1)
      (REPEAT , (ONESUB))
      (; $)
      (: QUOTE 1)
      (: CANSUBS 1)))
    (ONESUB ($ID = (GEXPR)
      (# CONS 2)))
  ))(QUOTE PARSEDEF))
```

```
(DEFLIST(QUOTE(
  (TCOMQ <proof-command>)
  (CLDES <clause-number>)
  (CANREL NONE)
  (SUBS <values>)
  (ONESUB <value>)
))(QUOTE SENAME))
```

(ADDOVAR GLOBALVARS INTERRUPTFLAG BREAK)  
(ADDOVAR BREAKMACROS (DEFER (RPAQQ BREAK T)

OK))

✓ [RPAQQ VTLPROPS ([JENDFN (NOSET MACRO] ((PSPRI PSPAT PSEXP PSPROP PSVAL PSTEST)

[RPAQQ VTLADVICE ((INTERRUPT BEFORE NIL (INTERTEST]

✓ [NIL PSTEAD]

(APPLYMAPC VTLPROPS (FUNCTION NEWPROP))

(APPLYMAPC VTLADVICE (FUNCTION ADVISE))

(LISPXPRINT (QUOTE (VTLSET))

T)

(VTLSET)

STOP

(FILECREATED "27-MAY-74 23:34:16" VTP 20052

changes to: FNSUBST, TPLIST, FNSUBSTP, FACBOOL

previous date: "25-MAY-74 10:59:00")

(DEFINEQ

(GENVAR

[LAMBDA (V)

(GENNAME V (FUNCTION (LAMBDA (X)  
(OCCTHM X]))

(OCCTHM

[LAMBDA (V CTX)

(MAPTHCLAUSEP [FUNCTION (LAMBDA (X)  
(COND  
((OCCURSFREE V X)  
(RTFRM OCCTHM T])

CTX)

NIL]))

(FNSPLIT

[LAMBDA (E E1 X Y OP)

(PROG ((XL (CDR X))  
(YL (CDR Y))  
(L (FNSL E1 CTX))  
G C)

(COND

((FMEMB E L)  
(FAIL)))

[SETQ G

(COND

(NIL

(\* (ARGPERM (CAR X)))

[PERMMAP (APPEND YL)

(FUNCTION (LAMBDA (Z)

(SETQ G

(CONS (FSSUB (PAIREQ XL Z NIL OP)  
E E1)

G])

(MK (QUOTE AND)

G))

(T (FSSUB (PAIREQ XL YL NIL OP)

E E1])

(SETFNSL E1 (CONS E L)

CTX)

(SELECTQ (SETQ G (CTXEV G CTX))

(T (FAIL))

(NIL (SUCCESS E1 CTX))

(FNSPLIT1 E1 G]))



```

(EQSUBST
 [LAMBDA (L)
  (PROG ([Z (SORT L (FUNCTION (LAMBDA (X Y)
    (ILESSP (CADR X)
    (CADR Y)
    M V X S FL)
  LO [COND
    ([SETQ X (SOME (CADDAR Z)
      (FUNCTION (LAMBDA (D)
        (AND (NOT (FMEMB (CAR D)
          V))
          (NOT (FMEMB (CAR D)
            FL]
      (SETQ V (CONS (CAAR X)
        V))
      (SETQ S (CONS (CONS (CAR V)
        (FSUBMAKE S (CDAR X)))
        S))
      (SETQ M (CONS (CAAR Z)
        M))
      (SETQ FL (FREEV (CDAR S)
        FL])
    (COND
      ((SETQ Z (CDR Z))
      (GO LO)))
    (EQSUB1 M V S])

```

```

(EQSUB1
 [LAMBDA (L V S)
  (MAPC L (FUNCTION REMCLAUSE))
  [MAPTHCLAUSEP (FUNCTION (LAMBDA (X Y)
    (OR ((EQCAR X (QUOTE OR))) (FMEMB (CAR X) ESUBOPS1)
    ((EQCAR X (QUOTE FA))) (FMEMB (CAR X) ESUBOPS2)
    (EQSUB2 V X S Y])

```

```

(MAPC (CS1 (QUOTE OR))
  (FUNCTION (LAMBDA (X)
    (EQSUB2 V X S NIL]
  (MAPC (CS1 (QUOTE FA))
    (FUNCTION (LAMBDA (X)
      (EQSUB2 V X S NIL]
  (SETCSL CTX (APPEND S (CSL CTX]))

```

```

(EQSUB2
 [LAMBDA (V X S Y)
  (COND
    ((OCCURSFREE V X)
    (OR Y (SETQ Y (THCLAUSEP X)))
    (REMCLAUSE X)
    (PUTCLAUSE (FSUBMAKE S X)
      NIL Y X]))

```

```

(EQSUBL
  EA (L V S)
  (MAPC L (F/Y)
  (MAPC (CS1 Y) (F/X)
  (EQSUB2 V X S NIL]))

```

(EQSUBL ESUBOPS1 V S)

(EQSUBL ESUBOPS2 V S)



```
(EQUALTERMS
 [LAMBDA (E)
  (AND (EQUALPRED (CAR E))
        (NCONS (EQUALT1 (CADR E)
                        (CADDR E))
              (NCONS (EQUALT1 (CADDR E)
                              (CADR E))
                    (CADR E))
              NIL]))
```

```
(EQUALT1
 [LAMBDA (X Y)
  (AND (VARP X)
        (NOT (OCCURSFREE X Y))
        (CONS X Y]))
```

```
(EQSAVE
 [LAMBDA (R W)
  (SETQ MATCHLIST (CONS (LIST R (COUNT (CDAR W)
                                         W)
                        MATCHLIST))
        (FAIL]))
```

```
(SUBFIX
 [LAMBDA (L S)
  (COND ((NULL L)
         S)
        (T (SUBFIX (CDR L)
                    (CONS (CONSN (CAR L)
                                (CAAR L)
                                (FSUBMAKE S (CDAR L)))
                          S))
```

```
(CJREF
 [LAMBDA (L V CJ)
  (AND (CDR L)
        (OR (NEQ (CAR V)
                 (BC CJ))
            (NEQ (FLENGTH L)
                 (FLENGTH (CDR V))
```

```
(MATCHMIN
 [LAMBDA (E)
  (PROG ((S (EXPSIZE E))
         (COND ((OR (NULL MATCHVAR1)
                    (ILESSP S MATCHVAR1))
                (SETQ MATCHVAR1 S)
                (SETQ MATCHLIST E)))
        (FAIL]))
```

← FN

← [COND (FN (APPLY\* FN E)) (T (EXPSIZE E))

```
(ESUBSIZE
 [λ (E) (* E is (OR --))
  (MAPRED (CDR E)
          (F/XY (IPLUS (ESUBSIZE| X) Y))
          ]])
```

```
(ESUBSIZE|
 [λ (E) (IPLUS (EXPSIZE E)
               (COND ([COND ((EQCAR E 'EQUAL)
                            (REITERMS E))
                      (T (EQUALTERMS E))
                    -1 0)
                 (T 0))
```

(FACBOOL  
[LAMBDA (E)

(PROG [~~(C1 (NEWCTX CTX (QUOTE JFAC1)  
(LIST E)))~~ A  
~~(C2 (NEWCTX CTX (QUOTE JFAC2)  
(LIST E)))~~ R C1 C2

(FLAG (THCLAUSEP E CTX))  
(L (SORT [MAPCAR (CDR E)

(FUNCTION (LAMBDA (X) (~~(EXPSIZE X)~~  
(CONS (~~(PLUS (COND  
(EQUALPRED (CAR X))~~ (ESUBSIZE X)  
(T 0))  
(SELECTO (CAR X)  
(EQUAL 3)  
0))))

X]  
(FUNCTION (LAMBDA (X Y)  
(ILESSP (CAR X)  
(CAR Y]

(SETQ THLIST (CONS C1 (CONS C2 THLIST)))

(REMCLAUSE E C1)

(PUTCLAUSE ~~(CDR L)~~ A  
C1 FLAG E)

(REMCLAUSE E C2)

(PUTCLAUSE [MKAND (LIST (NOTOF ~~(CDR L)~~ A  
(MK (CAR E)  
(REMV1 ~~(CDR L)~~ A  
A (CDR E])

R  
C2 FLAG E])

(ENDJFAC  
[LAMBDA (CTX VAL)

(COND  
((OR (NEQ (CAR VAL)  
T)

(EVERY (SONCTX (CDR CTX))  
(FUNCTION PROOFDONE)))

(PROOFEND (CDR CTX)  
VAL])

(SETQ C1 (NEWCTX CTX (QUOTE JFAC)  
(LIST A E)))  
(SETQ C2 (NEWCTX CTX (QUOTE JFAC)  
(LIST R E)))

(EVBOOL  
[LAMBDA (E)

(PROG [(E1 (MAPCARN (CDR E)  
(FUNCTION (LAMBDA (X)  
(CTXEV X CTX]

(COND  
((EQ E1 (CDR E))

(\* Only safe to fail  
because CTXEV doesn't  
look at disjunctions.)

(FAIL))

(T (REPCLAUSE E (MK ~~(QUOTE OR)~~ (CAR E)  
E1)  
(QUOTE EVBOOL)  
CTX])

(EXPSIZE  
[LAMBDA (E)  
(COUNT E)])

(FAMTEST  
[LAMBDA (Q MIN MAX)

(\* FAM2 uses arguments  
free)

(PROG [(TL (MATCHLIS Q))  
(M (ITIMES CTXFACTOR (FLENGTH (CURGTR)) TDPACTX)  
(PROG ((MIN (DIFFERENCE MIN M)) IPLUS  
(MAX (DIFFERENCE MAX M)))  
(SETQ MATCHLIST (NCONC (FAM1 (CAR TL)  
(CDR TL))  
MATCHLIST))  
(FAIL)])

(FAMADD  
[LAMBDA (L) UNIT  
(PROG (Q Z E AL SCORE)  
[SETQ L (SORT L (FUNCTION (LAMBDA (X Y)  
(IGREATERP (IPLUS (CDAR X)  
(CDDR X))  
(IPLUS (CDAR Y)  
(CDDR Y))

L1 [SETQ E (CAAR (SETQ Z (CAR L))  
(SETQ Q (CDAAR Z))  
(SETQ SCORE (IPLUS (CDAR Z)  
(CDDR Z)))

(ADDRESS E Q)  
(SETQ AL (ALCUT (CADR Z)))  
[COND

((CDR AL)  
(SETQ AL (REVERSE AL])  
(OR (FAMPUT Q E AL SCORE) UNIT  
(COND  
((SETQ L (CDR L))  
(GO L1))  
(T (FAIL]))

CAAR → MLEXP  
CDAAR → MLQUANT  
CADR → MLAL  
CDAR → MLVALΦ  
CDDR → MSCORE

(ALCUT  
[LAMBDA (AL)  
(PROG (AL1)  
[COND  
([SETQ AL1 (SOME AL (FUNCTION (LAMBDA (X)  
(EQ (CDR X)  
(QUOTE NOBIND])  
(SETQ AL (LDIFF AL AL1])  
(RETURN AL])



```

[COND
  ((SETQ Z (QMATCH (CAAR M)
                   E
                   (CDAR M))))
  (CADAR M) (CDDAR M)
  (END (FRPLACD END Z)) (FRPLACD (FLAST Z) R)
  (T (SETQ R Z)) (SETQ R Z)
  (SETQ END (FLAST Z))
  (SETQ M (CDR M))
  (GO L1)
L3 [MAPC
  R
  (FUNCTION (LAMBDA (X)
    (PROG [(Y (IPLUS V (CDDA X)
                    (AND (GEQ Y MIN)
                        (LEQ Y MAX)
                        (SETQ R1
                          (CONS (COND
                                (E1 (CONS E1 (CDA X)))
                                (T X))
                                R1)]
    (MLSCORE
    (MLAS
  (SETQ M1 (CDR M1))
  (GO L0])

```

```

(FAMPUT
 [LAMBDA (Q E AL SCORE)
 (PROG ((EVALARGS NIL)
        N)
 (RETURN (COND
          ((NEQ (SETQ N (CTXEV (FSUBMAKE AL Q T)))
                T)
           (PUTCLAUSE N NIL (QUOTE GOAL)
                      Q E)
           (NEQCAR N (QUOTE OR))
           (ADDRESSL E Q) U
          (AND * (OR (NEQCAR N
                        'OR))
                 (NOT UNIT)))
          (FAMPUT1
           (Q E AL SCORE N)
           (* For tracing))
          (UNIT

```

```

(SFSUBST
 [LAMBDA (E)
 (PROG ((E1 (SFSUB E T)))
 (COND
  ((NEQ E1 E)
   (REPCLAUSE E E1 (QUOTE SFSUBST)
               CTX))
  (T (FAIL]))
(FNSUBSTP
 [LAMBDA (E PFLAG)
 (AND (NEQCAR E (QUOTE FA))
      (FNSUBP E PFLAG)
      ([LAMBDA (E1)
        (AND (NEQ E1 E)
              (LIST E1)
              (FNSUB E (LIST E)
                     PFLAG)]
        (NEQ (SETQ E1 (CONVNF E1 NIL))
              E)
        (CONVNF * NIL)

```

```
(FNSUBST
 [LAMBDA (E E1)
  (REPCLAUSE E (CAR E1)
   (QUOTE FNSUBST)
   CTX])
```

```
(CXSET
 [LAMBDA (VAR E C)
  (EQSUB1 (NLIST C)
   (LIST VAR)
   (LIST (CONS VAR E]))
```

```
(CXNEQ
 [LAMBDA (C)
  (PROG ([E1 (MK* (QUOTE GEQ)
   (CADR C)
   (ADD1 (CADDR C])
   [E2 (MK* (QUOTE LEQ)
   (CADR C)
   (SUB1 (CADDR C])
   C1 C2)
  (SETQ THLIST (CONS C1 (CONS C2 THLIST)))
  [SETCTX (SETQ C1 (NEWCTX CTX (QUOTE JNECASE)
   (LIST E1 C])
  (REMCLAUSE C)
  (PUTCLAUSE E1)
  [SETCTX (SETQ C2 (NEWCTX CTX (QUOTE JNECASE)
   (LIST E2 C])
  (REMCLAUSE C)
  (PUTCLAUSE E2)
  (FAIL])
```

```
(ENDJNECASE
 [LAMBDA (CTX VAL)
  (COND
   ((EQ (CAR VAL)
    (QUOTE CX))
   (PROOFEND (CDR CTX)
    VAL))
  ((EVERY (SONCTX (CDR CTX))
   (FUNCTION PROOFDONE))
  (FAILURE (CDR CTX]))
```

```
(CXANY
 [LAMBDA (C)
  (PROG ([L (RELTERMS C (FUNCTION (LAMBDA (X)
   (NOT (CONSTP X]
   (V 0))
  (OR L (FAIL))
  (OR (CDR L)
   (SETQ V (CDAR L)))
  (EQSUB1 (LIST C)
   (LIST (CAAR L))
   (LIST (CONS (CAAR L)
    V)))
  (PUTCLAUSE (FSSUB V (CAAR L)
   C)
   CTX])
```

```
(CXFA1
[LAMBDA (Q)
  (PROG ((R ([LAMBDA (EVALARGS)
              (DOMSATL (BVL (CADR Q))
              (CADDR Q]
              NIL))
        R1)
  (COND
```

(\* CXFA2 will deal with this)

```
  ((EQ R T)
   (FAIL))
  ((EQ (SETQ R1 (CTXEV R CTX))
   T)
   (FAIL))
  ((NULL R1)
   (REMCLAUSE Q CTX)
   (RETURN))
  (T (PUTCLAUSE (COND
                ((EQCAR R1 (QUOTE AND))
                 R1)
                (T (NOTOF R1)))
                CTX NIL Q)
     (RETURN))
```

```
(CXFA2
[LAMBDA (Q)
  (PROG [(V (FREEV (CADDR Q)
                  NIL
                  (BVL (CADR Q]
```

(\* Assign arbitrarily, since must be in both upper & lower bounds)

```
  (COND
   (V
    [EQSUB1 NIL V (MAPCAR V (FUNCTION (LAMBDA (X)
                                        (CONS X 1]
    (RETURN))
   (T (REMCLAUSE Q CTX)
      (ENUMFA (CADDR Q)
              NIL NIL]))
```





(LISPXPRINT (QUOTE VTPFNS)

T)  
(RPAQQ VTPFNS

(GENVAR OCCTHM FNSPLIT FSPAT1 FNSPLIT1 ENDJFNSPLIT PAIREQ EQSUBST  
EQSUB1 EQSUB2 EQUALTERMS EQUALT1 EQSAVE SUBFIX CJREFF MATCHMIN  
FACBOOL ENDJFAC EVBOOL EXPSIZE FAMTEST FAMADD ALCUT FAM1 FAM2  
FAMPUT SFSUBST FNSUBSTP FNSUBST CXSET CXNEQ ENDJNECASE CXANY  
CXFA1 CXFA2 ENUMFA))

(LISPXPRINT (QUOTE VTPVARS)

T)

(RPAQQ VTPVARS (CTXFACTOR FSPAT NFSPAT ~~JFAC1~~ CXLIST RANGEOFFSET SUMSIZE  
(PROP (JFORM JENDFN)  
JFAC1 JFAC2 JFNSPLIT JCX JNECASE)  
(ENDUMP VTP)))

(RPAQQ CTXFACTOR 5)

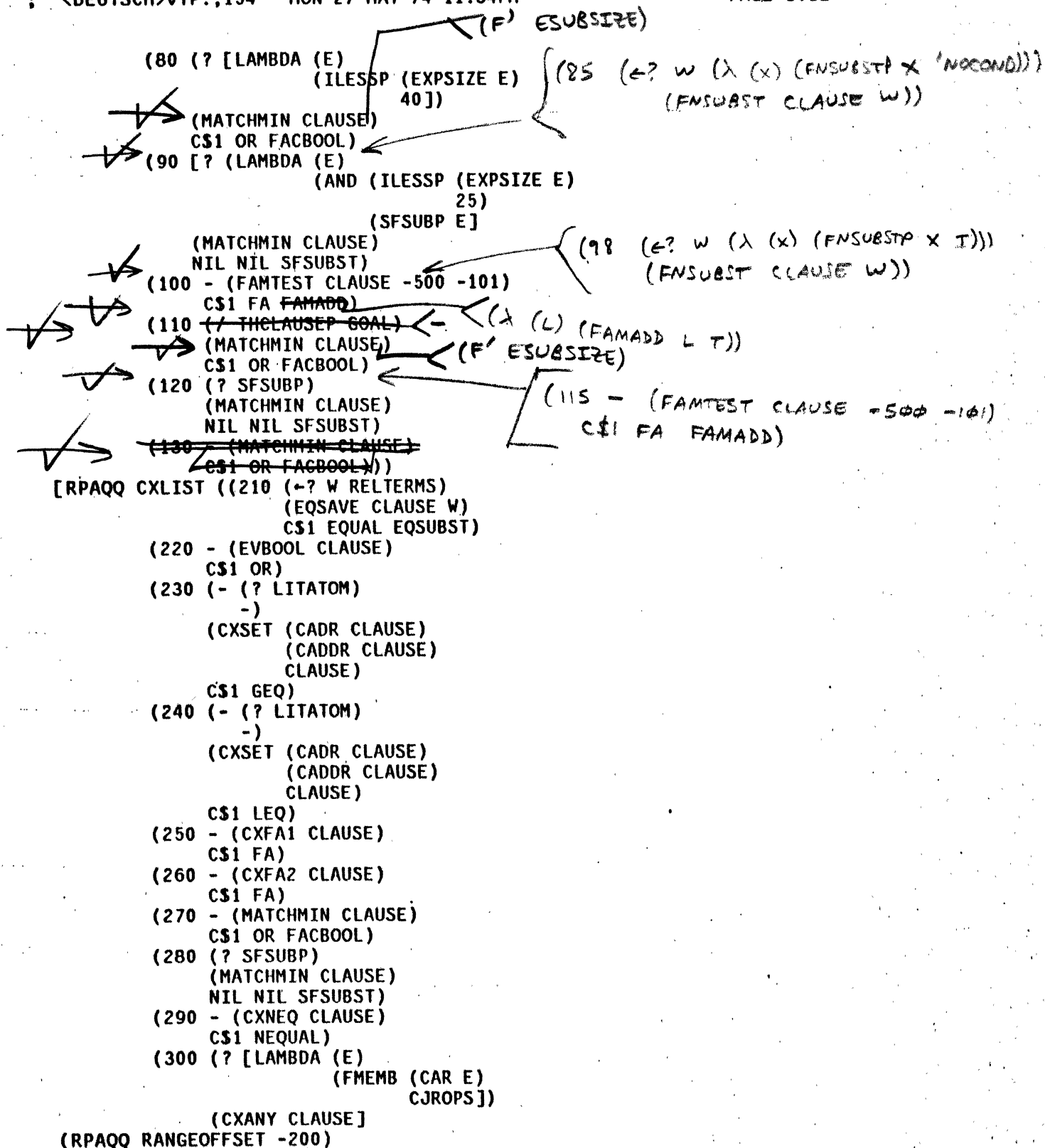
[RPAQQ FSPAT (! [PLUS 0 (+ X (- --))  
(TIMES -1 (+ Y (\$ (FSPAT1 X)  
(PLUS 0 (TIMES -1 (+ X (- --)))  
(+ Y (\$ (FSPAT1 X)

[RPAQQ NFSPAT (~~NOT ((? EQUALPRED~~ ((? (λ (X) (EQUALPRED (NOTOP X)))  
(+ X (- --))  
(+ Y (\$ (FSPAT1 X)

(RPAQQ TPLIST ((10 (+? W RELTERMS)  
(EQSAVE CLAUSE W)  
CS1 EQUAL EQSUBST)

Use ADDVARS

- (11 (+? W EQUALTERMS)  
(EQSAVE CLAUSE W)  
NIL NIL EQSUBST)
- (20 - (EVBOOL CLAUSE)  
CS1 OR)
- (30 - (FAMTEST CLAUSE -100 1000)  
CS1 FA FAMADD)
- (40 (NEQUAL (\$ FSPAT)  
0)  
(FNSPLIT CLAUSE CLAUSE X Y)  
CS1 NEQUAL)
- (50 (\$ NFSPAT)  
(FNSPLIT CLAUSE CLAUSE X Y (QUOTE EQUALP))  
CS1 NEQUALP)
- (60 [FA - - (! (+ E1 (NEQUAL (\$ FSPAT)  
0))  
(OR -- (+ E1 (NEQUAL (\$ FSPAT)  
0])  
(FNSPLIT E1 CLAUSE X Y)  
CS1 FA)
- (70 [FA - - (! (+ E1 (\$ NFSPAT))  
(OR -- (+ E1 (\$ NFSPAT]  
(FNSPLIT E1 CLAUSE X Y (QUOTE EQUALP))  
CS1 FA)
- (75 (+? W FNSUBSTP)  
(FNSUBST CLAUSE W))



SUMSIZE = 2φ

```

(DEFLIST(QUOTE(
(JFAC1 ("to prove one disjunct of " PCN"))
(JFAC2 ("to prove remaining disjuncts of " PCN"))
(JFNSPLIT ("to show " PCN " by showing " PRINSTAT))
(JCX ("for counterexample"))
(JNECASE ("for case " PRINSTAT " of " PCN))
))(QUOTE JFORM))

```

(JFAC ("assuming " PRINSTAT " (from " PCN ")))

PRINSTAT →  
NPRIN2

```

(DEFLIST(QUOTE(
(JFAC ENDJFAC)
(JFAC2 ENDJFAC)
(JFNSPLIT ENDJFNSPLIT)
(JCX ENNULL)
(JNECASE ENDJNECASE)
))(QUOTE JENDFN))

```

(((MEXP . MQUANT) . MVAL) MLAL . MSCORE)  
(MACRO SETMACRO)  
( (MLEB) . MLAS)(MACRO SETMACRO))

```

[RPAQQ VTPPROPS ((MATCHL NIL CTX)
  (RESL (ADD)
    CTX)
  (FNSL NIL CTX)
  (CSL NIL HASH (SETHASHQ CSA 100 2.0])
  (APPLYMAPC VTPPROPS (FUNCTION NEWPROP))
  (ADDOVAR GLOBALVARS CSA)
  (RPAQQ VTPFNS
    (GENVAR OCCTHM FNSPLIT FSPAT1 FNSPLIT1 ENDJFNSPLIT PAIREQ EQSUBST
      EQSUB1 EQSUB2 EQUALTERMS EQUALT1 EQSAVE SUBFIX CJREFF MATCHMIN
      FACBOOL ENDJFAC EVBOOL EXPSIZE FAMTEST FAMADD ALCUT FAM1 FAM2
      FAMPUT SFSUBST FNSUBSTP FNSUBST CXSET CXNEQ ENDJNECASE CXANY
      CXFA1 CXFA2 ENUMFA))

```

```

(DEFLIST(QUOTE(
  (FACBOOL (27-MAY-74 . 2302))
  (FNSUBSTP (27-MAY-74 . 2251))
  (FNSUBST (27-MAY-74 . 110))
))(QUOTE EDITDATE))

```

```

(PROGN (QUOTE JUSTEVALUATE)
  (FILEMAP (NIL (151 16102 (GENVAR 163 . 253) (OCCTHM 257 . 475) (FNSPLIT 479
    . 1118) (FSPAT1 1122 . 1296) (FNSPLIT1 1300 . 1482) (ENDJFNSPLIT 1486 . 1602)
  (PAIREQ 1606 . 1994) (EQSUBST 1998 . 2953) (EQSUB1 2957 . 3429) (EQSUB2 3433
    . 3624) (EQUALTERMS 3628 . 3788) (EQUALT1 3792 . 3877) (EQSAVE 3881 . 4045)
  (SUBFIX 4049 . 4300) (CJREFF 4304 . 4472) (MATCHMIN 4476 . 4710) (FACBOOL
    4714 . 5478) (ENDJFAC 5482 . 5703) (EVBOOL 5707 . 6263) (EXPSIZE 6267 . 6306)
  (FAMTEST 6310 . 6804) (FAMADD 6808 . 7575) (ALCUT 7579 . 7852) (FAM1 7856
    . 8284) (FAM2 8288 . 10448) (FAMPUT 10452 . 10781) (SFSUBST 10785 . 10983)
  (FNSUBSTP 10987 . 11160) (FNSUBST 11164 . 11257) (CXSET 11261 . 11368) (CXNEQ
    11372 . 11993) (ENDJNECASE 11997 . 12240) (CXANY 12244 . 12699) (CXFA1 12703
    . 13179) (CXFA2 13183 . 13662) (ENUMFA 13666 . 16099))))))
STOP

```

→ (ADDPROPS TRACEGROUP  
{ P from VPP }

→ ESUBOPS1 = (LEQ GEQ)

ESUBOPS2 = (EQUAL NEQUAL OR FA)