

DAN ROSS

B5700  
SNOBOL  
MANUAL

this listing is a copy of  
tape CUBEA13  
file SNOBOL/L200011

---

when a copy of this file is on  
the disk, it should be named  
SNOBOL/MANUAL  
type = SEQ

print this manual using program  
(source) SNOBOL/LISTER  
(object) SNOBOL/LISTER  
(see the source in the  
B5700 UTILITIES binder)

1000 %PL 78 FOR 8 LINES/INCH.  
2000 B5500 SNOBOL3 MANUAL.  
3000  
4000  
5000  
6000  
7000  
8000  
9000  
10000  
11000  
12000  
13000 JOHN M. CHAMBERS  
14000 COMPUTER SCIENCES DEPT.  
15000 1210 W. DAYTON STREET  
16000 UNIVERSITY OF WISCONSIN  
17000 MADISON, WISCONSIN 53706  
18000  
19000  
20000 %SECTION "CONTENTS."  
21000 %PAGE 1  
22000 CONTENTS:  
23000  
24000 1. INTRODUCTION  
25000 2. CONTROL CARDS  
26000 2.1. MCP CONTROL CARDS.  
27000 2.2. SNOBOL CONTROL CARDS.  
28000 3. PROGRAM ORGANIZATION  
29000 3.1. INSTRUCTIONS  
30000 3.2. PROGRAM-DEFINED FUNCTIONS  
31000 3.2.1. FUNCTION CALLS.  
32000 3.2.2. FUNCTION RETURNS.  
33000 3.2.3. FUNCTION CODE.  
34000 3.2.4. CALLING DEFINE().  
35000 3.3. PROGRAM ENTRY POINTS.  
36000 4. RULES.  
37000 4.1. THE STRING REFERENCE.  
38000 4.2. RULES WITH STRING REFERENCE ONLY.  
39000 4.3. RULES WITH PATTERNS.  
40000 4.4. RULES WITH REPLACEMENTS.  
41000 4.5. RULES WITH BOTH PATTERN AND REPLACEMENT.  
42000 5. EXPRESSIONS.  
43000 5.1. NAMES AND VALUES.  
44000 5.1.1. LITERALS.  
45000 5.1.2. NAMES  
46000 5.1.3. INDIRECTION.  
47000 5.2. CONCATENATION.  
48000 5.3. FUNCTION CALLS.  
49000 5.4. ARITHMETIC.  
50000 5.4.1. DIVISION MODES.  
51000 5.4.2. FAILURE OF ARITHMETIC.

JAVAMAN

BRJOD

DAN K022

Moore Business Forms, Inc. sv 1412

52000 5.4.3. ARITHMETIC FUNCTIONS.  
53000 5.4.4. ADDITIONAL COMMENTS ON ARITHMETIC.  
54000 6. PATTERNS.  
55000 6.1. CONSTANT PATTERN ELEMENTS.  
56000 6.2. STRING VARIABLES.  
57000 6.2.1. ARBITRARY STRING VARIABLES.  
58000 6.2.2. BALANCED STRING VARIABLES.  
59000 6.2.3. FIXED-LENGTH STRING VARIABLES.  
60000 6.3. BACK REFERENCES.  
61000 6.4. PATTERN MATCH MODES.  
62000 6.5. ASSIGNMENT OF STRING VARIABLES.  
63000 6.6. PATTERN MATCH ALGORITHM.  
64000 7. GO-TO PARTS, LABELS.  
65000 7.1 LABELS.  
66000 7.2. SIMPLE GO-TO PARTS.  
67000 7.3. RESERVED LABELS.  
68000 7.4. COMPUTED GO-TOS.  
69000 7.5. ERRORS IN GO-TO PARTS.  
70000 8. FAILURE.  
71000 8.1. CAUSES OF FAILURE.  
72000 8.2. ORDER OF EXECUTION.  
73000 9. INTRINSIC FUNCTIONS.  
74000 10. INPUT/OUTPUT.  
75000 10.1. I/O STRINGS.  
76000 10.2. OPEN AND CLOSED FILES.  
77000 10.3. STANDARD I/O STRINGS.  
78000 10.4. TELETYPES.  
79000 10.4.1. TELETYPE INPUT AND OUTPUT.  
80000 10.4.2. TELETYPE WAITING TIMES.  
81000 10.4.3. TELETYPE ADDRESSES.  
82000 10.4.4. SPECIAL CHARACTERS FOR THE TELETYPE.  
83000 10.5. CAUSES OF I/O.  
84000 11. DEBUGGING AIDS.  
85000 11.1. TRACING.  
86000 11.2. INTERACTIVE DEBUGGING.  
87000 12. MISCELLANEOUS.  
88000 APPENDIX A. BNF DEFINITIONS FOR SNOBOL3.  
89000 APPENDIX B. BNF NOTATION.  
90000 APPENDIX C. B5500 CHARACTER SET.  
91000 APPENDIX D. RUNNING JOBS FROM A TELETYPE.  
92000 INDEX.  
93000  
94000 %SECTION 1  
95000 %PAGE 1  
96000 1. INTRODUCTION  
97000  
98000 THIS MANUAL DESCRIBES THE PROGRAMMING LANGUAGE SNOBOL3, AS  
99000 IMPLEMENTED ON THE BURROUGHS B5500 AT THE UNIVERSITY OF WISCONSIN  
100000 (MADISON) BY JOHN M. CHAMBERS, WITH MUCH HELP IN IN THE DESIGN AND  
101000 INITIAL CODING STAGES BY WILLIAM KRUEGER, AND SOME HELP IN WRITING  
102000 THE COMPILER BY DAVID WILSON.  
103000 THE VERSION OF SNOBOL3 DESCRIBED HERE IS BASICALLY IDENTICAL  
104000 WITH THE LANGUAGE DEVELOPED AT BELL LABS, WITH THE OBVIOUS DIFFERENCES  
105000 OF CHARACTER SET AND I/O FACILITIES. (THERE IS ALSO A SLIGHT PROBLEM  
106000 WITH DIVISION, CAUSED BY THE FACT THAT THE B5500 HARDWARE ROUNDS  
107000 QUOTIENTS, WHILE MOST OTHER MACHINES TRUNCATE. SEE THE SECTION  
108000 ON ARITHMETIC.) THERE ARE, AS USUAL, A NUMBER OF "EXTENSIONS",

109000 MOSTLY IN THE INTRINSIC FUNCTIONS. THE BASIC SNOBOL3 FUNCTIONS  
 110000 ACT AS THEY DO ON OTHER SYSTEMS, BUT THOSE WHOSE ACTIONS ARE MACHINE  
 111000 DEPENDENT ARE OFTEN DIFFERENT. THERE ARE A NUMBER OF NEW FUNCTIONS  
 112000 THAT REFLECT THE PECULIARITIES OF THE B5500 OPERATING SYSTEM.  
 113000 MOST SNOBOL3 PROGRAMS WRITTEN ON OTHER MACHINES CAN BE RUN ON  
 114000 THE B5500 WITH ONLY MINOR CHANGES IN THE I/O, MOST OF WHICH CAN BE  
 115000 MADE BY THE COMPILER ITSELF.  
 116000 AN EXTREMELY USEFUL FACILITY OF THE B5500 IS THE AVAILABILITY OF  
 117000 TELETYPES AS I/O DEVICES. NATURALLY, THIS IS AVAILABLE TO SNOBOL  
 118000 IN AN EASILY-USED MANNER, AND A SIMPLE, VERY USEFUL INTERACTIVE  
 119000 DEBUGGING AID HAS BEEN SUPPLIED FOR TELETYPE USERS.  
 120000 %SECTION 2  
 121000 %PAGE 1  
 122000 2. CONTROL CARDS  
 123000  
 124000 THERE ARE TWO TYPES OF CONTROL CARDS RELEVANT FOR SNOBOL JOBS--  
 125000 THOSE THAT GIVE INFORMATION TO THE OPERATING SYSTEM (THE "MASTER  
 126000 CONTROL PROGRAM", OR "MCP"), WHICH START WITH A QUESTION MARK; AND  
 127000 THOSE THAT GIVE INFORMATION TO THE SNOBOL COMPILER, WHICH START WITH  
 128000 A MINUS SIGN ("-").  
 129000  
 130000 2.1. MCP CONTROL CARDS.  
 131000 AT THE START OF A DECK OF CARDS FOR BATCH JOBS, OR TYPED TO  
 132000 INITIATE A TELETYPE JOB, THE USER NEEDS A SET OF MCP CONTROL CARDS  
 133000 FOR BATCH JOBS. THESE ARE SEPARATE CARDS, EACH STARTING WITH A  
 134000 QUESTION MARK ("ILLEGAL CHARACTER"). FOR TELETYPE JOBS, THE ENTIRE  
 135000 SET OF CONTROL CARDS IS PRECEDED BY A PAIR OF QUESTION MARKS, AND  
 136000 THE CARDS ARE SEPARATED BY SEMICOLONS (";"), WITH AN ARROW AFTER THE  
 137000 LAST CONTROL CARD.  
 138000 THE FIRST TWO CARDS IN A BATCH JOB SHOULD BE:  
 139000  
 140000 ? USER <PROJ#> / <USER#>  
 141000 ? EXECUTE SNOBOL/SNOBOL  
 142000  
 143000 THE FIRST CARD IDENTIFIES THE USER TO THE MCP; THE SECOND TELLS IT  
 144000 TO INITIATE SNOBOL. THE PROGRAM AND DATA DECK MUST BE PRECEDED BY  
 145000  
 146000 ? DATA <NAME>  
 147000  
 148000 WHICH NAMES THE INPUT DECK. BETWEEN THE EXECUTE CARD AND THE DATA  
 149000 CARD CAN COME ANY NUMBER OF OTHER CONTROL CARDS; SOME OF THE CARDS  
 150000 THAT ARE USEFUL TO SNOBOL USERS ARE DESCRIBED BELOW. ONE OF THE  
 151000 OTHER CONTROL CARDS MUST BE THE FOLLOWING:  
 152000  
 153000 ? FILE PROGRAM = <NAME>  
 154000  
 155000 WHERE <NAME> IS THE SAME AS THE <NAME> ON THE DATA CARD. THIS  
 156000 ASSIGNS THE CARD DECK AS THE INPUT FILE TO THE COMPILER. SOME OTHER  
 157000 MCP CONTROL CARDS THAT CAN BE USED ARE:  
 158000  
 159000 ? PROCESS = <INTEGER>  
 160000 THIS PUTS A CPU TIME LIMIT TO THE ENTIRE RUN OF <INTEGER>  
 161000 MINUTES. IF THIS LIMIT IS EXCEEDED, THE JOB WILL BE TERMINATED  
 162000 BY THE MCP. THE DEFAULT CPU TIME LIMIT IS 2 MINUTES.  
 163000 ? IO = <INTEGER>  
 164000 THIS PUTS A TIME LIMIT OF <INTEGER> MINUTES TO THE I/O TIME  
 165000 THAT CAN BE USED BY THE RUN. THE DEFAULT I/O LIMIT IS

Moore Business Forms, Inc. sv 14121

166000 5 MINUTES.

167000 ? STACK = <INTEGER>

168000 THIS ASSIGNS A STACK OF <INTEGER> WORDS TO THE RUN. THIS IS ONLY NECESSARY IF THE PROGRAM REACHES GREAT DEPTHS OF FUNCTION CALLS, CAUSING STACK OVERFLOWS. THE DEFAULT STACK SIZE IS 512.

169000

170000

171000

172000 ? CORE = <INTEGER>

173000 THIS ASSIGNS A CORE ESTIMATE OF <INTEGER> WORDS TO THE RUN. THE JOB WILL BE STARTED WHEN THIS MANY WORDS BECOME AVAILABLE. THE DEFAULT CORE ESTIMATE IS 10000.

174000

175000

176000

177000 A BATCH JOB CAN EXECUTE A SNOBOL PROGRAM FROM A DISK FILE IF THE DATA CARD IS OMITTED AND THE PROGRAM CARD IS CHANGED TO:

178000

179000

180000 ? FILE PROGRAM = <NAME1>/<NAME2> SERIAL

181000

182000 WHERE <NAME1>/<NAME2> IS THE NAME OF THE DISK FILE. IN GENERAL, THE INPUT FILE NAMED "PROGRAM" CAN BE ASSIGNED TO ANY FILE WHICH CAN BE READ WITH 10,30 BLOCKING.

183000

184000 ANOTHER CONTROL CARD WHICH IS OCCASIONALLY USEFUL, THOUGH THE SAME RESULTS CAN BE ACHEIVED WITH THE SNOBOL FUNCTION FILL(), IS THE FILE CARD, WHICH HAS THE GENERAL FORM:

185000

186000

187000

188000

189000 ? FILE <NAME> = <NAME1>/<NAME2> <UNIT>

190000

191000 WHERE <NAME> IS THE "INTERNAL" NAME OF THE FILE, AND <NAME1>/<NAME2> IS THE "EXTERNAL" NAME OF THE FILE ACTUALLY BEING USED. FOR SNOBOL JOBS, <NAME> SHOULD BE THE I/O STRING NAME, WITH THE EXCEPTION OF THE STRING READ, WHICH THE MCP THINKS IS CALLED "PROGRAM".

192000

193000 <UNIT> INDICATES WHAT SORT OF I/O DEVICE IS TO BE USED. SOME POSSIBLE VALUES FOR <UNIT> ARE:

194000

195000 SERIAL MEANS SERIAL DISK FILE

196000

197000 PRINT MEANS LINE PRINTER FILE--NO BACK-UP

198000

199000 PRINT OR BACK UP MEANS LINE PRINTER OR PRINTER BACK-UP

200000

201000 TAPE MEANS MAGNETIC TAPE

202000 FOR INPUT FILES, IT IS ONLY NECESSARY TO DISTINGUISH DISK FILES FROM NON-DISK FILES. OF THE STANDARD SNOBOL FILES, NEWDISK IS ASSUMED TO BE A DISK FILE UNLESS DECLARED OTHERWISE; ALL OTHERS ARE NON-DISK UNLESS THE USER SPECIFICALLY ASKS FOR DISK.

203000

204000 SOME EXAMPLES OF FILE CARDS ARE:

205000

206000

207000 ? FILE CARD = RALPH/GRK SERIAL (DISK FILE)

208000 ? FILE PROGRAM = MLX/PLP (NON-DISK; ASSIGNS READ)

209000 ? FILE PRINT = P PRINT OR BACK UP (GOES TO PRINTER IF IT CAN)

210000 ? FILE PRINT = PRNT PRINT (LP, NO BACK-UP)

211000 ? FILE NEWDISK = ART/BEVL (GOES TO DISK)

212000

213000 NOTE THAT THE FILE CARD ONLY APPLIES TO THE FIRST FILE THAT AN I/O STRING IS ASSIGNED TO. TO RE-ASSIGN AN I/O STRING TO A SECOND FILE, IT IS NECESSARY TO USE THE SNOBOL FUNCTIONS CLOSE() AND FILL().

214000

215000 NOTE ESPECIALLY THAT TO ASSIGN THE INPUT FILE READ (WHICH CAN ONLY BE DONE WITH TELETYPE JOBS, SINCE IT IS ASSIGNED TO THE PROGRAM FILE AUTOMATICALLY FOR BATCH JOBS), THE CARD USED STARTS WITH

216000

217000 "? FILE PROGRAM", NOT WITH "? FILE READ".

218000

219000

220000

221000

222000 %CP 30

223000 SOME EXAMPLES OF BATCH JOBS:

224000

225000 TO RUN A SIMPLE BATCH JOB USING A CARD DECK, THE FOLLOWING MIGHT

226000 BE USED:

227000

228000 ? USER 1234/5678

229000 ? EXECUTE SNOBOL/SNOBOL

230000 ? FILE PROGRAM = AJONES

231000 ? PROCESS = 3

232000 ? IO = 7

233000 ? DATA AJONES

234000 PROGRAM DECK

235000 .

236000 :

237000 .

238000 END

239000 DATA DECK

240000

241000

242000

243000 TO RUN A SNOBOL PROGRAM FROM THE DISK FILE TMILLR/SDCK:

244000

245000 ? USER 9876/5432

246000 ? EXECUTE SNOBOL/SNOBOL

247000 ? FILE PROGRAM = TMILLR/SDCK SERIAL

248000 ? PROCESS = 5

249000 ? IO = 10

250000 ? CORE = 18000

251000

252000

253000

254000 %CP 25

255000 2.2. SNOBOL CONTROL CARDS.

256000

257000 VARIOUS ACTIONS OF THE SNOBOL COMPILER ARE CONTROLLED BY SNOBOL

258000 CONTROL CARDS, WHICH ARE ANY CARDS WITH A "-" IN COLUMN 1. IF SUCH

259000 A CARD IS NOT A VALID CONTROL CARD, A DIAGNOSTIC MESSAGE WILL BE

260000 PRINTED, AND THE CARD WILL BE IGNORED. THE CONTROL CARDS THAT ARE

261000 CURRENTLY RECOGNIZED ARE:

262000

263000

264000

265000 %CP 4

266000 -LIST <INTEGER>

267000 THIS INITIATES LISTING OF THE PROGRAM ON THE LINE PRINTER FILE,

268000 PRINT. THERE WILL BE <INTEGER> BLANK LINES BETWEEN EACH LINE

269000 OF THE LISTING. IF <INTEGER> IS OMITTED, 0 (ZERO) IS ASSUMED.

270000

271000

272000

273000 %CP 2

274000 -UNLIST

275000 THIS CARD STOPS THE LISTING.

276000

277000

278000

279000 %CP 3

280000 -PCC  
281000 THIS CAUSES ALL CONTROL CARDS TO BE LISTED, EVEN IF THE PROGRAM  
282000 ITSELF IS NOT BEING LISTED.  
283000  
284000  
285000  
286000 %CP 2  
287000 -EJECT  
288000 THIS EJECTS THE LISTING TO THE TOP OF THE NEXT PAGE  
289000  
290000  
291000  
292000 %CP 3  
293000 -SPACE <INTEGER>  
294000 THIS PRODUCES <INTEGER> BLANK LINES IN THE LISTING.  
295000  
296000  
297000  
298000 %CP 6  
299000 -WIDTH <INTEGER>  
300000 THIS SAYS THAT THE COMPILER IS TO USE THE FIRST <INTEGER>  
301000 CHARACTERS OF EACH CARD READ FROM A CARD-IMAGE FILE. IT DOES  
302000 NOT APPLY TO INPUT FROM A TELETYPE. THE COMPILER ASSUMES  
303000 THAT 72 CHARACTERS ARE TO BE USED IF THIS CARD ISNT USED. THE  
304000 MAXIMUM CARD WIDTH IS 80.  
305000  
306000  
307000  
308000 %CP 6  
309000 -26  
310000 THIS SAYS THAT THE DECK WAS PUNCHED ON AN 026 KEYPUNCH, WITH  
311000 THE IBM/CDC CHARACTER SET. A CHARACTER-SET TRANSLATION WILL  
312000 BE MADE BY THE COMPILER BEFORE COMPILING EACH CARD. THE HANDLING  
313000 OF "SPECIAL CHARACTERS" NOT ON THE 026 IS NOT VERY WELL-DEFINED.  
314000 THIS CARD CAN ALSO BE WRITTEN "-026" IF DESIRED.  
315000  
316000  
317000  
318000 %CP 9  
319000 -3600  
320000 THIS CAUSES THE SAME CONVERSION AS THE -26 CARD, AND ALSO TRANSLATES  
321000 THE I/O STRING NAMES USED ON THE CDC3600 SNOBOL AS IMPLEMENTED  
322000 AT THE UNIVERSITY OF WISCONSIN. IN PARTICULAR, THE FOLLOWING  
323000 SUBSTITUTIONS ARE MADE:  
324000  
325000 "SYSPIT" BECOMES "READ "  
326000 "SYSLOK" BECOMES "LOOK "  
327000 "SYSPPT" BECOMES "PUNCH "  
328000  
329000  
330000  
331000 %CP 2  
332000 -B5500  
333000 THIS CANCELS ANY PREVIOUS -26 OR -3600 CARD.  
334000  
335000  
336000

337000 %CP 5  
 338000 -PUNCH  
 339000 THIS CAUSES THE COMPILER TO PUNCH OUT THE PROGRAM DECK ON THE  
 340000 CARD PUNCH, STARTING WITH THE NEXT CARD. THIS CAN BE USED WITH  
 341000 A -26 OR -3600 CARD TO PRODUCE A B5500 DECK FROM A SNOBOL PROGRAM  
 342000 WRITTEN FOR ANOTHER MACHINE.  
 343000  
 344000  
 345000  
 346000 %CP 8  
 347000 -INFORM  
 348000 THIS TURNS ON A SET OF SYSTEM INFORMATIVE DIAGNOSTICS. IT IS  
 349000 EQUIVALENT TO THE RUN-TIME USE OF MODE("INFORM"). SOME MESSAGES  
 350000 THAT ARE PRODUCED BY THE COMPILER IF THIS CARD IS USED ARE:  
 351000 LABELS REFERENCED THAT ARE UNDEFINED (AT THE END OF THE LISTING);  
 352000 INACCESSIBLE INSTRUCTIONS; USING -DEFINE TO RE-DEFINE A PREVIOUSLY  
 353000 DEFINED FUNCTION; USING NON-NUMERIC LITERALS IN ARITHMETIC;  
 354000 AND OTHER THINGS THAT MAY BE ADDED IN THE FUTURE.  
 355000  
 356000  
 357000  
 358000 %CP 2  
 359000 -SILENCE  
 360000 THIS CANCELS ANY PREVIOUS -INFORM CARD.  
 361000  
 362000  
 363000  
 364000 %CP 4  
 365000 -WAIT <INTEGER>  
 366000 THIS SETS THE WAITING TIME FOR TELETYPE I/O TO <INTEGER>  
 367000 SECONDS. IT IS EQUIVALENT TO USING WAIT("<INTEGER>") AT  
 368000 RUN TIME.  
 369000  
 370000  
 371000  
 372000 %CP 12  
 373000 -LIMIT <QUANTITY> <INTEGER>  
 374000 THIS SETS A PROGRAM LIMIT TO THE THING NAMED BY <QUANTITY>  
 375000 TO THE VALUE OF <INTEGER>. IF THIS LIMIT IS EXCEEDED, THE  
 376000 PROGRAM WILL BE TERMINATED ABNORMALLY. SOME POSSIBLE VALUES  
 377000 FOR <QUANTITY> ARE:  
 378000  
 379000 RULES SETS A LIMIT TO THE NUMBER OF RULES THAT CAN BE  
 380000 EXECUTED.  
 381000 PROCESS SETS A TIME LIMIT (IN SECONDS) TO THE PROGRAMS  
 382000 CPU TIME.  
 383000 IO SETS A TIME LIMIT (IN SECONDS) TO THE PROGRAMS  
 384000 I/O TIME.  
 385000  
 386000  
 387000  
 388000 %CP 9  
 389000 -SIZE <INTEGER>  
 390000 THIS TELLS THE COMPILER HOW MANY INSTRUCTIONS TO EXPECT IN THE  
 391000 PROGRAM. IT NEED NOT BE USED; IT ALLOWS THE COMPILER TO SET  
 392000 UP ITS TABLES TO TRY TO MINIMIZE THE TIME FOR BOTH COMPILATION  
 393000 AND EXECUTION. THE VALUE OF <INTEGER> SHOULD BE ABOUT (SAY,



Moore Business Forms, Inc. sv 1412

394000 WITHIN 25 OF) THE NUMBER OF INSTRUCTIONS IN THE PROGRAM. THIS CARD  
395000 CAN CAUSE SAVINGS OF 5-10% IN LARGE PROGRAMS, IF USED, A -SIZE  
396000 CARD MUST APPEAR BEFORE ALL PROGRAM MATERIAL (INCLUDING -SET,  
397000 -DEFINE, -COMPILE, AND -LOAD CARDS).

398000  
399000  
400000

401000 %CP 21

402000 -SET <IDENTIFIER> <LITERALS>  
403000 THIS CARD ASSIGNS AN INITIAL VALUE TO THE VARIABLE NAMED  
404000 <IDENTIFIER>. THE VALUE IS DETERMINED BY SCANNING THE CARD  
405000 FOR ALL LITERALS, AND CONCATENATING THEM. ANYTHING BETWEEN  
406000 THE LITERALS IS TREATED AS COMMENT. THUS, A LONG STRING CAN  
407000 BE BROKEN UP BETWEEN SEVERAL CARDS IN ANY CONVENIENT WAY. SOME  
408000 EXAMPLES OF -SET CARDS ARE:

409000  
410000 -SET ALPHABET "ABCDEFGHIJKLMNOPQRSTUVWXYZ"  
411000 -SET CARDNAME "GEORGE/POSDECK"  
412000 -SET SUITS "HEARTS,CLUBS,SPADES,DIAMONDS"  
413000 -SET CITIES "NEW YORK,N.Y./CHICAGO,ILL./BOSTON,MASS./"  
414000 . "SAN FRANCISCO,CALIF./DENVER,COL./CLEVELAND,OHIO/"  
415000 . "NEW URLEANS,LA./LOS ANGELES,CALIF./ATLANTA,GA./"  
416000 . "PHOENIX,ARIZ./PITTSBURG,PENN./SEATTLE,WASH./"  
417000 -SET OPS " + = x / \* "

418000  
419000 THE MAJOR REASON FOR USING A -SET CARD IS THAT IT SAVES THE  
420000 SPACE THAT WOULD BE REQUIRED FOR THE EQUIVALENT RUN-TIME ASSIGNMENT  
421000 STATEMENT. THIS CAN BE IMPORTANT WHEN THE SMALL MEMORY SIZE  
422000 OF THE B5500 IS TAKEN INTO ACCOUNT.

423000  
424000  
425000

426000 %CP 34

427000 -DEFINE <LIT1> , <LIT2> , <LIT3>  
428000 THIS SETS UP A FUNCTION DURING COMPILATION IN THE SAME WAY THAT  
429000 A CALL OF DEFINE(<LIT1>,<LIT2>,<LIT3>) WOULD DURING EXECUTION.  
430000 THE PARAMETERS MUST ALL BE SINGLE LITERALS (NO CONCATENATION  
431000 WILL BE DONE BY THE COMPILER). LIKE THE -SET CARD, A -DEFINE  
432000 CARD SAVES THE SPACE AND EXTRA TIME THAT WOULD BE REQUIRED FOR  
433000 AN INSTRUCTION TO CREATE THE FUNCTION AT RUN TIME. IT USUALLY  
434000 TURNS OUT EASIER TO USE THE -DEFINE CARD, THOUGH, SINCE THIS  
435000 MAKES IT POSSIBLE TO PLACE THE DEFINITION TOGETHER WITH THE  
436000 FUNCTION CODE WITHOUT THE NEED TO LINK THE DEFINITIONS TOGETHER  
437000 WITH GO-TOS. THE PARAMETERS HAVE THE FOLLOWING MEANINGS:

438000  
439000 <LIT1> IS THE "FUNCTION PROTOTYPE", OF THE FORM:

440000  
441000 <FCT>( <PARAMS> )  
442000

443000 WHERE <FCT> IS THE FUNCTION NAME, AND <PARAMS> IS A  
444000 LIST OF FORMAL PARAMETERS, SEPARATED BY COMMAS.

445000 <LIT2> CONTAINS A LABEL, WHICH IS THE ENTRY POINT OF  
446000 THE FUNCTION. IF <LIT2> IS MISSING (I.E., THE COMMAS  
447000 HAVE NO LITERAL BETWEEN THEM), THEN THE ENTRY POINT IS  
448000 TAKEN TO BE SPELLED THE SAME AS <FCT>.

449000 <LIT3> CONTAINS A LIST OF LOCAL VARIABLES--A LIST OF  
450000 IDENTIFIERS, SEPARATED BY COMMAS.

451000  
452000 FOR FURTHER INFORMATION, SEE SECTIONS 3.2 (FUNCTION CALLS) AND 9  
453000 (INTRINSIC FUNCTIONS--DEFINE()). SOME EXAMPLES OF -DEFINE CARDS:  
454000  
455000 -DEFINE "M(A1,A2)","M1","V,XY,ALPH"  
456000 -DEFINE "MAX(A,B)"  
457000 -DEFINE "MIN(MIN,X)"  
458000 -DEFINE "FCT()",,"X"  
459000 -DEFINE "MRGE(L)","MRGENTRY","L1,L2,L3"  
460000 -DEFINE("GLORCH(A,B,C,D,E,F,G,H,I)","GLORK","J,K,L,M,N,O,P")  
461000  
462000  
463000  
464000 %CP 4  
465000 -DEBUG  
466000 THIS TURNS ON "DEBUG MODE" (SEE SECTION 11.2) DURING COMPILATION, AND  
467000 CAUSES THE JOB TO BE SUSPENDED JUST BEFORE THE FIRST INSTRUCTION IS  
468000 EXECUTED (IF THERE IS A TELETYPE ATTACHED).  
469000  
470000  
471000  
472000 %CP 42  
473000 -COMPILE <NAME1>/<NAME2>  
474000 THIS CARD CAUSES THE COMPILER TO COMPILE THE DISK FILE NAMED  
475000 <NAME1>/<NAME2> INTO THE PROGRAM AT THE POINT WHERE THE  
476000 CARD IS FOUND. THE COMPILER WILL READ EVERYTHING IN THE FILE  
477000 UP TO AN END CARD OR THE END-OF-FILE, AND THEN RETURN TO THE  
478000 ORIGINAL FILE (OR TELETYPE). IF AN END CARD IS FOUND, IT IS  
479000 SEARCHED FOR A LABEL; IF ONE IS FOUND, IT BECOMES THE PROGRAM  
480000 ENTRY POINT (UNLESS A LATER END CARD CHANGES IT). IF THE FILE  
481000 NAMED IS NOT ON THE DISK, A DIAGNOSTIC WILL BE PRINTED AND THE  
482000 CARD WILL BE IGNORED. ANY CARD-IMAGE FILE (WITH 80-CHAR LOGICAL  
483000 RECORDS) CAN BE READ BY THE COMPILER.  
484000 THE MOST OBVIOUS USE OF THE -COMPILE CARD IS TO ENABLE USERS TO  
485000 BUILD UP A SNOBOL "LIBRARY" OF FUNCTIONS, WHICH CAN THEN BE USED  
486000 BY ANYONE BY SIMPLY INCLUDING A -COMPILE CARD FOR EACH FUNCTION  
487000 DESIRED. ALTHOUGH STANDARDS FOR SUCH A LIBRARY ARE CLEARLY NOT  
488000 ENFORCEABLE, SOME GUIDELINES ARE SUGGESTED HERE WHICH SHOULD MAKE  
489000 IT EASIER TO USE LIBRARY FILES:  
490000  
491000 1) THOROUGH DOCUMENTATION SHOULD EITHER BE INCLUDED IN THE CODE OR  
492000 THERE SHOULD BE COMMENTS AT THE START OF THE FILE TELLING THE  
493000 USER WHERE TO FIND DOCUMENTATION.  
494000 2) FUNCTIONS SHOULD BE SET UP BY THE -DEFINE CONTROL CARD, RATHER  
495000 THAN BY THE FUNCTION DEFINE(), SO THAT THE USER DOESNT HAVE TO  
496000 SET UP A LOT OF GO-TOS INTO THE CODE.  
497000 3) THE USER SHOULD GET INTO THE CODE BY CALLING A FUNCTION, RATHER  
498000 THAN BY A GO-TO. THIS MAKES IT MUCH EASIER TO INTEGRATE THE CODE  
499000 WITH THE MAIN PROGRAM.  
500000 4) NO INSTRUCTIONS IN THE LIBRARY FILE SHOULD TRANSFER OUT  
501000 OF THE FILE, WITHOUT EXPLICIT DESCRIPTIONS OF ALL LABELS  
502000 THAT THE CALLING PROGRAM SHOULD CONTAIN. THIS SHOULD  
503000 BE AVOIDED ENTIRELY IF POSSIBLE.  
504000 5) THERE SHOULD BE VERY EXPLICIT COMMENTS DESCRIBING ALL GLOBAL  
505000 VARIABLES USED OR ALTERED BY THE CODE. WHENEVER POSSIBLE,  
506000 THE CODE SHOULD DO ITS OWN INITIALIZATION BY MEANS OF  
507000 THE -SET CONTROL CARD, RATHER THAN ASKING THE USER TO

508000 FIGURE OUT WHAT TO DO.

509000 6) LABELS SHOULD BE SUCH THAT USERS OF THE FILE ARE UNLIKELY

510000 TO DUPLICATE THEM IN THEIR OWN CODE. THIS IS MOST EASILY

511000 HANDLED BY MAKING EACH LABEL INCLUDE EITHER THE NAME OF

512000 THE FILE OR THE NAME OF THE FUNCTION THEY ARE USED IN. NOTE

513000 THAT IF TWO LIBRARY FILES USE THE SAME LABEL, THEY CANT

514000 BE USED IN THE SAME PROGRAM.

515000

516000

517000

518000 %CP 12

519000 -LIBRARY <NAME1>/<NAME2>

520000 THIS CARD CREATES A "LIBRARY" FILE CONTAINING THE COMPILED

521000 VERSION OF THE PROGRAM. THE FILE IS NAMED <NAME1>/<NAME2>.

522000 THIS LIBRARY FILE CAN THEN BE USED IN A LATER RUN WITHOUT

523000 RECOMPILING IT BY USING THE -LOAD CARD (SEE BELOW). A -LIBRARY

524000 CARD IS NORMALLY THE LAST CARD IN THE PROGRAM DECK, COMING

525000 JUST BEFORE THE END CARD, BUT IT CAN APPEAR ANYWHERE IN A

526000 PROGRAM, AND THE PROGRAM STORED IN THE FILE WILL LOOK AS IF

527000 THE -LIBRARY CARD HAD BEEN FOLLOWED BY AN END CARD. SEVERAL

528000 -LIBRARY CARDS CAN BE USED IN THE SAME PROGRAM, IF DESIRED.

529000 IN ADDITION TO THE PROGRAM ITSELF, ANY LIMITS SET BY -LIMIT

530000 CARDS ARE SAVED, AS IS THE ENTRY POINT AS OF THE -LIBRARY CARD.

531000

532000

533000

534000 %CP 13

535000 -LOAD <NAME1>/<NAME2>

536000 THIS LOADS A LIBRARY FILE CREATED BY A -LIBRARY CARD IN A

537000 PREVIOUS RUN. IF THE FILE NAMED IS NOT PRESENT OR IS NOT

538000 A VALID LIBRARY FILE, A DIAGNOSTIC MESSAGE WILL BE PRINTED.

539000 IF THE FILE WAS THERE, BUT CONTAINED DAMAGED CODE, THE COMPILER

540000 WILL OFTEN BE UNABLE TO RECOVER FROM THE BAD INFORMATION LOADED

541000 INTO ITS TABLES WHEN THE PROBLEM IS DISCOVERED; THIS WILL

542000 CAUSE THE COMPILER TO SIMPLY GIVE UP AND THE JOB WILL TERMINATE.

543000 ONLY ONE -LOAD CARD CAN BE USED IN A PROGRAM, AND IT MUST BE

544000 BEFORE ANY OTHER PROGRAM MATERIAL (INCLUDING INSTRUCTIONS, -SET,

545000 -DEFINE CONTROL CARDS). THE -LOAD CARD CAN BE FOLLOWED BY MORE

546000 PROGRAM, WHICH WILL BE COMPILED AND ADDED ONTO THE PROGRAM

547000 LOADED FROM THE FILE. (THE FILE, OF COURSE, WILL NOT BE ALTERED.)

548000

549000

550000

551000 %SECTION 3

552000 %PAGE 1

553000 3. PROGRAM ORGANIZATION.

554000

555000 A SNOBOL PROGRAM CONSISTS OF A SET OF INSTRUCTIONS. THE PROGRAM

556000 CAN BE INPUT FROM A CARD-IMAGE FILE, OR FROM A TELETYPE. EACH CARD

557000 IMAGE FROM AN INPUT FILE STARTS A NEW INSTRUCTION, UNLESS THE FIRST

558000 CHARACTER IS ".,," IN WHICH CASE THE CARD IS A CONTINUATION OF THE

559000 PREVIOUS INSTRUCTION. IN B5500 SNOBOL, IT IS ALSO POSSIBLE TO FIT

560000 SEVERAL INSTRUCTIONS ON ONE CARD, THOUGH THIS IS RARELY DONE.

561000 FOR JOBS RUN FROM A TELETYPE, THE PROGRAM IS READ FROM THE TELETYPE,

562000 WITH EACH INPUT BEING AN INSTRUCTION. CONTINUATIONS ARE NOT VALID

563000 FROM A TELETYPE. THIS IS NOT REALLY A RESTRICTION, SINCE FEW PEOPLE

564000 ACTUALLY ATTEMPT TO INPUT ENTIRE PROGRAMS FROM A TELETYPE. IT IS FAR

565000 EASIER TO CREATE A DISK FILE CONTAINING THE PROGRAM (SAY, WITH AN  
 566000 EDITING PROGRAM), AND USE THE -COMPILE CARD TO DIRECT THE COMPILER  
 567000 TO THE FILE.  
 568000 SCATTERED AMONG THE INSTRUCTIONS IN A PROGRAM CAN BE ANY NUMBER OF  
 569000 SNOBOL CONTROL CARDS, WHICH ARE INSTRUCTIONS TO THE COMPILER, BUT  
 570000 ARE NOT ACTUALLY PART OF THE PROGRAM. IN ADDITION TO THE USUAL  
 571000 CONTROL CARDS, THE B5500 HAS CARDS THAT CAN BE USED TO DEFINE  
 572000 FUNCTIONS DURING COMPILATION, AND TO ASSIGN INITIAL VALUES TO PROGRAM  
 573000 VARIABLES.  
 574000 A PROGRAM DECK CAN ALSO CONTAIN ANY NUMBER OF COMMENT CARDS,  
 575000 WHICH ARE CARDS STARTING WITH "\*" OR "%". SUCH CARDS ARE PRINTED ON A  
 576000 LISTING, BUT OTHERWISE IGNORED. USERS ARE ENCOURAGED TO MAKE LIBERAL  
 577000 USE OF COMMENTS IN THEIR PROGRAMS.  
 578000 AT THE END OF A SNOBOL PROGRAM SHOULD BE AN "END" CARD, WHICH  
 579000 IS A CARD WITH "END " IN COLUMNS 1-4. THE FIRST STRING OF NON-BLANK  
 580000 CHARACTERS ON THE CARD IS TAKEN AS A LABEL WHICH IS THE "ENTRY POINT"  
 581000 TO THE PROGRAM. THAT IS, IT IS THE LABEL OF THE FIRST INSTRUCTION  
 582000 TO BE EXECUTED. IF THE ENTIRE CARD IS BLANK, THE FIRST INSTRUCTION  
 583000 IN THE PROGRAM IS ASSUMED TO BE THE ENTRY POINT.  
 584000 INSTRUCTIONS CAN BE EITHER LABELED OR UNLABELED. LABELS ARE  
 585000 NEEDED WHENEVER AN INSTRUCTION MUST BE TRANSFERRED TO BY ANOTHER ONE,  
 586000 AND OTHER INSTRUCTIONS ARE GENERALLY UNLABELED. A LABEL IS ANY  
 587000 STRING OF NON-BLANK CHARACTERS STARTING IN COLUMN 1. IF COLUMN 1  
 588000 IS A BLANK, THE INSTRUCTION IS UNLABELED. LABELS AREN'T REALLY A  
 589000 PART OF THEIR INSTRUCTIONS, BUT SIMPLY SERVE AS "NAMES" FOR THE  
 590000 INSTRUCTIONS.  
 591000 A SNOBOL PROGRAM IS A SINGLE SET OF INSTRUCTIONS, WITH NO SUB-PARTS  
 592000 (UNLIKE MOST OTHER PROGRAMMING LANGUAGES THAT HAVE SEPARATE BLOCKS  
 593000 OF CODE FOR EACH SUB-ROUTINE OR FUNCTION OR WHATEVER THEY ARE CALLED).  
 594000 ONE CONSEQUENCE OF THIS IS THAT STATEMENT LABELS IN A SNOBOL PROGRAM  
 595000 MUST BE UNIQUE. THERE IS VERY LITTLE CONSISTENCY BETWEEN IMPLEMEN-  
 596000 TATIONS ON HOW DOUBLY-DEFINED LABELS ARE TREATED. THE METHOD USED  
 597000 ON THE B5500 IS THIS: THE SECOND LABEL OVERWRITES THE FIRST AND  
 598000 CAUSES IT TO BE RE-DEFINED, BUT A WARNING MESSAGE IS PRINTED. THIS  
 599000 CAN BE USEFUL AT TIMES ON INTERACTIVE JOBS TO RE-DEFINE AN INSTRUCTION  
 600000 PREVIOUSLY COMPILED.  
 601000  
 602000  
 603000 %PAGE  
 604000 3.1. INSTRUCTIONS.  
 605000 <INSTRUCTION> ::= <LABELED INST>  
 606000 ::= <UNLABELED INST>  
 607000 <LABELED INST> ::= <LABEL> <UNLABELED INST>  
 608000 <UNLABELED INST> ::= <BLANKS> <INST>  
 609000 <INST> ::= <RULE>  
 610000 ::= <GO-TO PART>  
 611000 ::= <RULE> <GO-TO PART>  
 612000  
 613000  
 614000 THE VARIOUS FORMS OF INSTRUCTIONS CAN BE TREATED AS SPECIAL  
 615000 CASES OF THE GENERAL FORM:  
 616000  
 617000 <LABEL> <RULE> <GO-TO PART>  
 618000  
 619000 WHERE IT IS REMEMBERED THAT THERE IS ALWAYS AT LEAST ONE BLANK  
 620000 AFTER <LABEL>, IF THIS IS PRESENT, OR ELSE THE INSTRUCTION  
 621000 STARTS WITH AT LEAST ONE BLANK.

622000 THE <LABEL>, OF COURSE, IS JUST A NAME FOR THE INSTRUCTION,  
623000 A <LABEL> IS ONLY NEEDED IF SOME OTHER PART OF THE PROGRAM IS TO  
624000 TO TRANSFER TO THE INSTRUCTION, THOUGH THERE IS NO OBJECTION TO  
625000 USING LABELS WHICH ARE NEVER REFERENCED. LABELS ARE DESCRIBED  
626000 IN SECTION 7.  
627000 THE <RULE> IS THE PART OF AN INSTRUCTION THAT OPERATES ON DATA;  
628000 IT IS THE PART THAT "DOES THINGS". SEE SECTION 4 FOR DESCRIPTIONS  
629000 OF THE VARIOUS TYPES OF RULES AVAILABLE.  
630000 THE <GO-TO PART> TELLS WHAT INSTRUCTIONS ARE TO FOLLOW NEXT.  
631000 SNOBOL3 ALLOWS CONDITIONAL BRANCHING ON THE SUCCESS OR FAILURE  
632000 OF THE RULE. AN INSTRUCTION CAN ALSO CALCULATE THE LABEL  
633000 OF THE NEXT INSTRUCTION. OMISSION OF THE GO-TO PART CAUSES THE  
634000 INSTRUCTION TO "FALL THROUGH" TO THE NEXT INSTRUCTION IN THE PROGRAM.  
635000 SEE SECTION 7 FOR DETAILED DESCRIPTIONS OF GO-TO PARTS.  
636000  
637000  
638000 %CP 8  
639000 3.2 PROGRAM-DEFINED FUNCTIONS.  
640000  
641000 IN ADDITION TO THE FUNCTIONS PRE-DEFINED BY THE SNOBOL INTER=  
642000 PRETER, A PROGRAMMER CAN DEFINE HIS OWN FUNCTIONS CODED IN SNOBOL  
643000 BY USING EITHER THE FUNCTION DEFINE() (SEE SEC. 9) OR THE -DEFINE  
644000 CONTROL CARD (SEE SEC.2.2). IN SECTION 5.3, THE USE OF FUNCTIONS  
645000 CALLS IS DESCRIBED; THIS SECTION DESCRIBES THE DEFINITION OF THE  
646000 FUNCTIONS.  
647000  
648000  
649000 %CP 15  
650000 3.2.1. FUNCTION CALLS.  
651000  
652000 WHEN A PROGRAM-DEFINED FUNCTION IS CALLED, THE FOLLOWING OCCURS:  
653000  
654000 1) THE STRING WITH THE SAME NAME AS THE FUNCTION IS PUSHED DOWN,  
655000 AND SET NULL.  
656000 2) THE FORMAL PARAMETERS ARE PUSHED DOWN, AND THEN ASSIGNED THE  
657000 VALUES OF THE CORRESPONDING ACTUAL PARAMETERS. IF THERE ARENT  
658000 ENOUGH ACTUAL PARAMETERS, THE EXTRA FORMAL PARAMETERS ARE SET  
659000 NULL.  
660000 3) THE LOCAL VARIABLES ARE PUSHED AND SET NULL.  
661000 4) THE PROGRAM TRANSFERS TO THE FUNCTIONS ENTRY POINT.  
662000  
663000  
664000  
665000 %CP 15  
666000 3.2.2. FUNCTION RETURNS.  
667000  
668000 WHEN A PROGRAM TRANSFERS TO THE "RESERVED LABELS" RETURN AND  
669000 FRETURN, THE LAST FUNCTION CALLED IS TERMINATED, AND THE FOLLOWING  
670000 ACTIONS OCCUR:  
671000  
672000 1) THE VALUE OF THE STRING NAMED THE SAME AS THE FUNCTION IS GIVEN  
673000 TO THE CALLING INSTRUCTION AS THE VALUE OF THE FUNCTION.  
674000 2) THIS STRING, THE FORMAL PARAMETERS, AND THE LOCAL VARIABLES ARE  
675000 ALL POPPED UP, RETURNING THEM TO THEIR VALUES BEFORE THE CALL.  
676000 3) IF THE RETURN WAS MADE BY TRANSFERING TO "FRETURN", THE CALLING  
677000 INSTRUCTION FAILS; IF THE RETURN WAS BY "RETURN", THE CALLING  
678000 INSTRUCTION CONTINUES AS USUAL.

679000  
680000  
681000 %CP 10  
682000 3.2.3. FUNCTION CODE.  
683000 IN CONTRAST WITH MOST OTHER PROGRAMMING LANGUAGES, A SNOBOL  
684000 FUNCTION HAS NO "BODY" OF CODE THAT BELONGS TO THE FUNCTION ALONE.  
685000 THAT IS, FROM ITS ENTRY POINT, A FUNCTION CAN TRANSFER TO ANY  
686000 INSTRUCTION IN THE PROGRAM. RETURN OCCURS ONLY WHEN A TRANSFER TO  
687000 RETURN OR FRETURN IS MADE. THIS MEANS THAT SEVERAL SNOBOL  
688000 FUNCTIONS CAN SHARE ALL OR PART OF THEIR CODE. ALSO, NOTE THAT IF  
689000 A STRING IS NOT ONE OF THE STRINGS PUSHED AT THE CALL OF A FUNCTION,  
690000 THEN THAT FUNCTION CAN USE OR ALTER THE STRING FREELY.  
691000  
692000  
693000 %CP 12  
694000 3.2.4. CALLING DEFINE().  
695000  
696000 ONE VERY COMMON PROBLEM WITH NOVICE SNOBOL PROGRAMMERS IS THE  
697000 FAILURE TO REALIZE THAT THE FUNCTION DEFINE() ACTUALLY CREATES A  
698000 FUNCTION WHEN IT IS CALLED. IT IS NOT ENOUGH TO INCLUDE A CALL OF  
699000 DEFINE() IN A PROGRAM--THE PROGRAM MUST EXECUTE THE STATEMENT CON-  
700000 TAINING THE CALL, OR THE FUNCTION WILL NOT BE DEFINED.  
701000 THIS PROBLEM CAN BE AVOIDED ON THE B5500 BY USING THE -DEFINE  
702000 CONTROL CARD (SEC.2.2). SINCE MOST SNOBOL SYSTEMS DONT HAVE THIS  
703000 CONTROL CARD, USING IT WILL MAKE THE PROGRAM INCOMPATIBLE WITH THESE  
704000 OTHER SNOBOLS.  
705000  
706000  
707000  
708000 %CP 25  
709000 3.3. PROGRAM ENTRY POINTS.  
710000  
711000 IN ORDER TO START EXECUTION OF A SNOBOL PROGRAM, IT IS NECESSARY  
712000 TO INDICATE WHERE THE EXECUTION IS TO BEGIN. THE FIRST INSTRUCTION  
713000 TO BE EXECUTED, THE "ENTRY POINT", IS INDICATED BY THE END CARD AT  
714000 THE END OF THE PROGRAM. THE FIRST FOUR CHARACTERS OF THIS CARD ARE  
715000 "END "; THE COMPILER SCANS THE REST OF THE CARD FOR A NON-BLANK  
716000 CHARACTER. IF ONE IS FOUND, THE STRING UP TO THE NEXT BLANK OR TO  
717000 THE END OF THE CARD IS THE LABEL OF THE ENTRY POINT. FOR EXAMPLE:  
718000  
719000 END BEGIN  
720000  
721000 SAYS THAT EXECUTION IS TO START AT THE LABEL BEGIN. IF THERE IS  
722000 NO LABEL ON THE END CARD, THE FIRST INSTRUCTION IN THE PROGRAM IS  
723000 TAKEN TO BE THE ENTRY POINT.  
724000 IF THE PROGRAM CONTAINS -COMPILE CARDS, AND THE FILES THEMSELVES  
725000 CONTAIN END CARDS, THE LABELS (IF ANY) ON THESE END CARDS ARE NOTED,  
726000 AND THE LAST SUCH LABEL IS THE PROGRAM ENTRY POINT.  
727000 IF THE ENTRY POINT IS A LABEL THAT DOESNT OCCUR IN THE PROGRAM,  
728000 A DIAGNOSTIC MESSAGE TO THAT EFFECT IS PRINTED, AND NO EXECUTION  
729000 TAKES PLACE. THIS CAN BE USED TO GET COMPILES FOR SYNTAX CHECKING  
730000 OR A COMPILE TO LIBRARY WITHOUT IMMEDIATE EXECUTION.  
731000  
732000  
733000  
734000 %SECTION 4  
735000 %PAGE 1

1412Z  
Moore Business Forms, Inc. sv

736000 4. RULES.  
737000  
738000 <RULE> ::= <STR REF>  
739000 ::= <STR REF> <PATTERN>  
740000 ::= <STR REF> <REPLACEMENT>  
741000 ::= <STR REF> <PATTERN> <REPLACEMENT>  
742000 <STR REF> ::= <ELEMENT>  
743000 <PATTERN> ::= <ELEMENT>  
744000 ::= <STR VAR>  
745000 ::= <PATTERN> <PATTERN>  
746000 ::= <PATTERN> <BACK REF>  
747000 <REPLACEMENT> ::= <RPL DELIMITER> <EXPRESSION>  
748000 <RPL DELIMITER> ::= +  
749000 ::= =

750000  
751000 THERE ARE SEVERAL VARIETIES OF RULES, AS SHOWN ABOVE, BUT ALL  
752000 CAN BE VIEWED AS SPECIAL CASES OF THE GENERAL FORM:

753000  
754000 <STR REF> <PATTERN> <REPLACEMENT>

755000  
756000 IF THE PATTERN IS OMITTED, AN ARBITRARY STRING VARIABLE  
757000 WILL BE ASSUMED; THIS STRING VARIABLE WILL MATCH THE ENTIRE STRING  
758000 REFERENCE, RESULTING IN AN "ASSIGNMENT STATEMENT". IF THE REPLACEMENT  
759000 IS OMITTED, NO REPLACEMENT TAKES PLACE. IF BOTH PATTERN AND REPLACE-  
760000 MENT ARE OMITTED, THE STATEMENT MERELY EVALUATES THE STRING REFERENCE.  
761000 IN THE LATTER CASE, THE STRING REFERENCE CAN ALSO BE OMITTED, AND  
762000 THE STATEMENT IS A "NO-OP".  
763000 THE FOLLOWING SUB-SECTIONS DESCRIBE MORE THOROUGHLY THE VARIOUS  
764000 PARTS OF A RULE.

765000  
766000  
767000  
768000 %CP 20  
769000 4.1. THE STRING REFERENCE.

770000  
771000 <STR REF>, THE "STRING REFERENCE", IS THE STRING THAT THE RULE  
772000 IS TO OPERATE ON; IT IS THE "SUBJECT" OF THE STATEMENT. THE COMPILER  
773000 ASSUMES THAT THE SHORTEST COMPLETE EXPRESSION AT THE START OF A RULE  
774000 IS THE STRING REFERENCE; IF A STRING REFERENCE CONSISTING OF SEVERAL  
775000 STRINGS CONCATENATED IS DESIRED, IT MUST BE SURROUNDED BY PARENTHESES.  
776000 SOME EXAMPLES OF VALID STRING REFERENCES ARE:

777000  
778000 X  
779000 (ALPHA BETA + GAMMA)  
780000 SIZE(A)  
781000 \$("A" N)  
782000 "123456789"  
783000 ""  
784000 A + B

785000  
786000 EACH TYPE OF RULE INTERPRETS ITS STRING REFERENCE SLIGHTLY DIFFERENTLY,  
787000 AND SOME RULES WILL NOT ACCEPT ALL OF THE ABOVE AS STRING REFERENCE.  
788000 THE FOLLOWING DESCRIPTIONS OF INDIVIDUAL RULE TYPES DESCRIBES EACH  
789000 USAGE IN DETAIL.

790000  
791000  
792000

793000 %CP 20  
794000 4.2. RULES WITH STRING REFERENCE ONLY.  
795000  
796000 <RULE> ::= <STR REF>  
797000  
798000 THIS TYPE OF RULE SIMPLY EVALUATES THE STRING REFERENCE. THE  
799000 USUAL USE OF SUCH A RULE IS TO CALL A FUNCTION. THIS CAN BE DONE  
800000 EITHER FOR THE SIDE-EFFECTS OF THE FUNCTION, OR TO BRANCH ON THE  
801000 SUCCESS OR FAILURE OF THE FUNCTION. AN EXAMPLE OF THE LATTER IS:  
802000  
803000 .GT(SIZE(A),"5") :S(RM)  
804000  
805000 THIS STATEMENT WILL TRANSFER TO RM IF THE STRING A CONTAINS MORE  
806000 THAN FIVE CHARACTERS.  
807000 ANOTHER, RARELY USED, FORM OF THIS TYPE OF RULE IS WITH AN INPUT  
808000 STRING AS THE STRING REFERENCE. THIS TYPE OF RULE WILL INPUT A  
809000 RECORD (IF POSSIBLE), AND THE INPUT WILL BE LOST. FOR EXAMPLE, IF  
810000 THE PROGRAM HAS USED THE STRING LOOK TO READ THE NEXT CARD FROM THE  
811000 FILE READ, THE CARD CAN BE CLEARED FROM THE BUFFER BY:  
812000  
813000 READ  
814000  
815000 AND THE NEXT INPUT WILL GET THE NEXT RECORD.  
816000  
817000  
818000  
819000 %CP 37  
820000 4.3. RULES WITH PATTERNS.  
821000  
822000 <RULE> ::= <STR REF> <PATTERN>  
823000  
824000 THIS CAUSES A "PATTERN MATCH" TO TAKE PLACE. A PATTERN IS A  
825000 DESCRIPTION OF A STRING, AND A PATTERN MATCH IS AN ATTEMPT TO FIND  
826000 A SUBSTRING OF THE STRING REFERENCE WHICH IS DESCRIBED BY  
827000 THE PATTERN. THE VARIOUS MEANS OF BUILDING PATTERNS ARE DESCRIBED  
828000 IN SECTION 6. THE GENERAL WAY IN WHICH THIS TYPE OF RULE IS  
829000 EXECUTED IS AS FOLLOWS:  
830000  
831000 1) EVALUATE THE STRING REFERENCE. IF IT IS AN INPUT STRING, READ  
832000 A RECORD.  
833000 2) EVALUATE THE PATTERN, FROM LEFT TO RIGHT.  
834000 3) ATTEMPT TO MATCH THE PATTERN ON SOME SUBSTRING OF THE STR REF.  
835000  
836000 OF COURSE, IF ANY PART OF THE RULE FAILS, THE RULE IS TERMINATED,  
837000 AND THE FAILURE PART OF THE GO-TO PART IS EVALUATED.  
838000 FOR EXAMPLE, ASSUME THAT X = "ABCDE" AND E = "BC". THE RULE  
839000  
840000 X E "D" /S(M)  
841000 WILL SUCCEED, AND THE PROGRAM WILL TRANSFER TO M, THE STRING REF-  
842000 ERENCE, "ABCDE", CONTAINS THE PATTERN, E "D", OR "BCD". ON THE  
843000 OTHER HAND, THE RULE  
844000  
845000 X E "C" /S(M)  
846000 WILL FAIL, AND TRANSFER TO M WILL NOT BE MADE, BECAUSE THE STRING  
847000 REFERENCE DOES NOT CONTAIN THE PATTERN E "C", OR "BCC".  
848000 FOR ANOTHER EXAMPLE, THE RULE  
849000



850000 "1357" D /S(X)  
851000  
852000 WILL SUCCEED (AND TRANSFER TO X) IF D IS ANY OF THE FOLLOWING: "",  
853000 "1", "13", "135", "1357", "3", "35", "357", "5", "57", "7".  
854000  
855000  
856000

857000 %CP 30  
858000 4.4. RULES WITH REPLACEMENTS.

859000  
860000 <RULE> ::= <STR REF> <REPLACEMENT>  
861000 <REPLACEMENT> ::= <RPL DELIMITER> <EXPRESSION>  
862000 <RPL DELIMITER> ::= ←  
863000 ::= =  
864000

865000 THIS IS AN "ASSIGNMENT STATEMENT". THE STRING REFERENCE MUST  
866000 BE A STRING NAME (AN IDENTIFIER OR AN INDIRECT EXPRESSION). THE  
867000 VALUE OF THE <EXPRESSION> BECOMES THE NEW VALUE OF THE STRING  
868000 NAMED. IF THIS STRING IS AN OUTPUT STRING, OUTPUT OCCURS AFTER  
869000 THE ASSIGNMENT. NOTE THAT THERE ARE TWO DELIMITERS LISTED, "←"  
870000 AND "=". MANY COMPUTERS DO NOT HAVE THE "←" IN THEIR CHARACTER  
871000 SETS, SO THEY WILL ONLY ACCEPT A "=". THE B5500 HAS BOTH, AND  
872000 EITHER CAN BE USED TO INDICATE A REPLACEMENT. SOME EXAMPLES OF  
873000 ASSIGNMENT STATEMENTS ARE:  
874000

875000 X = "123"  
876000 ALPH.1 = BET.3 + GAMM.7  
877000 LIST ← NUM "/" ITEM "," LIST  
878000 \$("X" N) ← \$("X" N) ";" E  
879000 \$\$FR(A,\$B N) = \$A \$("B" N) FL(A1,A2 "," A3)  
880000 \$E ← F G "." EXP  
881000 I = J  
882000 N = .LT(N,MAX) N + "1"  
883000

884000 THE LAST EXAMPLE IS A "CONDITIONAL ASSIGNMENT". IF THE CALL OF .LT()  
885000 FAILS (I.E., IF N ≥ MAX), THE STATEMENT WILL FAIL DURING EVALUATION  
886000 OF THE REPLACEMENT, AND THE ASSIGNMENT WILL NOT BE CARRIED OUT.  
887000  
888000  
889000

890000 %CP 30  
891000 4.5. RULES WITH BOTH PATTERN AND REPLACEMENT.

892000  
893000 <RULE> ::= <STR REF> <PATTERN> <REPLACEMENT>  
894000

895000 THIS IS THE MOST COMPLEX FORM OF A RULE, IN WHICH A PATTERN MATCH  
896000 IS ATTEMPTED, AND IF IT SUCCEEDS, THE MATCHED SUBSTRING OF THE STRING  
897000 REFERENCE IS REPLACED WITH THE VALUE OF THE REPLACEMENT EXPRESSION.  
898000 SPECIFICALLY, THE FOLLOWING ACTIONS OCCUR IN THIS TYPE OF RULE:  
899000

- 900000 1) THE STRING REFERENCE IS EVALUATED, DOING INPUT IF IT IS AN  
901000 INPUT STRING.
- 902000 2) THE PATTERN IS EVALUATED FROM LEFT TO RIGHT.
- 903000 3) A PATTERN MATCH IS ATTEMPTED. IF THIS FAILS (AS WITH ANY  
904000 OTHER FAILURE), THE RULE IS TERMINATED AND THE FAILURE EXIT  
905000 IS TAKEN.
- 906000 4) THE REPLACEMENT IS EVALUATED.

907000 5) THE SUBSTRING OF THE STRING REFERENCE MATCHED BY THE PATTERN  
 908000 IS REPLACED BY THE VALUE OF THE REPLACEMENT EXPRESSION. IF  
 909000 THE STRING REFERENCE IS AN OUTPUT STRING, OUTPUT OCCURS.  
 910000  
 911000 AS USUAL, ANY PART OF THE RULE CAN FAIL. IF THIS HAPPENS, THE RULE  
 912000 IS IMMEDIATELY TERMINATED, AND THE FAILURE PART OF THE GO-TO PART  
 913000 IS EVALUATED.  
 914000 FOR EXAMPLE, IF AB = "COME", THEN THE RULE  
 915000  
 916000 AB "OM" = "RAT"  
 917000  
 918000 WILL SUCCEED, AND GIVE AB THE NEW VALUE "CRATE". IF AB HAD BEEN,  
 919000 SAY, "CAME", THE PATTERN MATCH WOULD HAVE FAILED, AND AB WOULD NOT  
 920000 HAVE BEEN ALTERED.  
 921000 NOTE THAT REPLACEMENT APPLIES ONLY TO THE SUBSTRING MATCHED BY  
 922000 THE PATTERN, NOT TO THE WHOLE STRING REFERENCE, AND THAT THE NEW  
 923000 VALUE THAT REPLACES THE SUBSTRING NEED NOT BE THE SAME SIZE AS THE  
 924000 SUBSTRING.  
 925000 AS A SPECIAL CASE, THE REPLACEMENT PART CAN BE EMPTY (SEE  
 926000 THE BNF DEFINITION OF <EXPRESSION>). IN THIS CASE, THE REPLACE-  
 927000 MENT EXPRESSION IS THE NULL STRING, AND THE MATCHED SUBSTRING IS  
 928000 DELETED. FOR EXAMPLE, THE RULE:  
 929000  
 930000 L A "." = :S(L)F(M)  
 931000  
 932000 WILL DELETE ALL PERIODS FROM A, AND THEN TRANSFER TO M.  
 933000  
 934000  
 935000  
 936000 %SECTION 5  
 937000 %PAGE 1  
 938000 5. EXPRESSIONS.  
 939000  
 940000 <EXPRESSION> ::= <ELEMENT>  
 941000 ::= <EXPRESSION> <CONCATENATE OP> <ELEMENT>  
 942000 ::= <EMPTY>  
 943000 <ELEMENT> ::= <NAMED EXPR>  
 944000 ::= <VALUE EXPR>  
 945000 <NAMED EXPR> ::= <IDENTIFIER>  
 946000 ::= \$ <ELEMENT>  
 947000 <VALUE EXPR> ::= <LITERAL>  
 948000 ::= <GROUPING>  
 949000 ::= <FCT CALL>  
 950000 ::= <ARITH EXPR>  
 951000 <CONCATENATE OP> ::= <BLANKS>  
 952000 <IDENTIFIER> ::= <LETTER>  
 953000 ::= <IDENTIFIER><LETTER>  
 954000 ::= <DIGIT>  
 955000 ::= <IDENTIFIER><DIGIT>  
 956000 ::= .  
 957000 ::= <IDENTIFIER> .  
 958000 <LITERAL> ::= " <STRING> "  
 959000 <GROUPING> ::= ( <EXPRESSION> )  
 960000 <FCT CALL> ::= <FCT NAME>( <PARAMS> )  
 961000 <FCT NAME> ::= <IDENTIFIER>  
 962000 <PARAMS> ::= <EXPRESSION>  
 963000 ::= <PARAMS> , <EXPRESSION>

Moore Business Forms, Inc. sv 14121

```

964000 <ARITH EXPR> ::= <A-TERM>
965000 ::= - <A-TERM>
966000 <A-TERM> ::= <A-TERM> <ADD OP> <M-TERM>
967000 ::= <M-TERM>
968000 <M-TERM> ::= <M-TERM> <MUL OP> <E-TERM>
969000 ::= <E-TERM>
970000 <E-TERM> ::= <E-TERM> <EXP OP> <ARITH ELT>
971000 ::= <ARITH ELT>
972000 <ARITH ELT> ::= <NAMED EXPR>
973000 ::= <GROUPING>
974000 ::= <FCT CALL>
975000 ::= <LITERAL>
976000 <ADD OP> ::= +
977000 ::= -
978000 <MUL OP> ::= *
979000 ::= x
980000 ::= /
981000 <EXP OP> ::= **
982000

```

983000 EXPRESSIONS ARE USED TO GENERATE STRING VALUES, AND TO FORM NEW  
984000 STRINGS FROM OLD. THE VARIOUS TYPES OF EXPRESSIONS ARE DESCRIBED  
985000 IN THE FOLLOWING SECTIONS.

986000  
987000  
988000

989000 %CP 12  
990000 5.1. NAMES AND VALUES.

991000  
992000 IN SNOBOL3, BOTH NAMES AND VALUES OF DATA ARE STRINGS, AND SO IT  
993000 IS QUITE NATURAL THAT THERE SHOULD BE SOME CONFUSION BETWEEN THE  
994000 TWO AMONG PROGRAMMERS; EXPECIALLY SINCE THE INDIRECTION OPERATOR  
995000 ("\$") PROVIDES A WAY TO MANIPULATE STRING NAMES AS WELL AS  
996000 THEIR VALUES. IN GENERAL, ANY STRING CAN BE USED AS BOTH A NAME  
997000 AND AS A VALUE. ONE RESTRICTION TO THIS IS THAT NAMES THAT APPEAR  
998000 AS SUCH IN SNOBOL CODE CAN ONLY BE THOSE THAT ARE "IDENTIFIERS"--  
999000 NON-NULL STRINGS OF LETTERS, DIGITS, AND PERIODS. OTHER NAMES  
1000000 MUST BE HANDLED INDIRECTLY.

1001000  
1002000  
1003000

1004000 %CP 25  
1005000 5.1.1. LITERALS.

1006000  
1007000 THE SIMPLEST WAY TO DESCRIBE A STRING IS TO GIVE THE STRING ITSELF.  
1008000 THIS IS DONE BY USING A "LITERAL", WHICH IS A STRING OF CHARACTERS  
1009000 BOUNDED BY QUOTES. QUOTES THEMSELVES CAN NOT OCCUR IN LITERALS, AND  
1010000 MUST BE HANDLED DIFFERENTLY. SOME EXAMPLES OF LITERALS ARE:

```

1011000
1012000 "1"
1013000 "A+B"
1014000 "THIS IS A STRING."
1015000 "ABCDEFGHIJKLM"
1016000 " GEORGE Q. SMITH "
1017000 " "
1018000 ""
1019000

```

1020000 THE LAST LITERAL IS THE NULL STRING, WHICH HAS A POSITION IN STRING

1021000 MANIPULATION SIMILAR TO ZERO IN ARITHMETIC (IN FACT, IT IS THE IDENTITY  
 1022000 ELEMENT FOR THE OPERATION OF CONCATENATION).  
 1023000 NOTE THAT, AS IN THE NEXT-TO-LAST EXAMPLE, A BLANK IS AS "GOOD"  
 1024000 A CHARACTER AS ANY OTHER, AND THE STRING " " IS NOT THE SAME AS  
 1025000 THE STRING " ". THEY CONTAIN DIFFERENT NUMBERS OF BLANKS.  
 1026000 TO HANDLE QUOTES, SNOBOL3 PROVIDES A SPECIAL STRING NAME, QUOTE,  
 1027000 WHICH IS INITIALIZED TO CONTAIN A SINGLE QUOTE AS ITS VALUE. THIS  
 1028000 CAN, OF COURSE, BE CHANGED BY THE PROGRAM.  
 1029000  
 1030000  
 1031000  
 1032000 %CP 25  
 1033000 5.1.2. NAMES  
 1034000  
 1035000 IT IS NECESSARY, OF COURSE, TO TALK ABOUT STRINGS WITHOUT SPECIFYING  
 1036000 (OR KNOWING) THEIR ACTUAL VALUES. SO SNOBOL PROVIDES THE ABILITY TO  
 1037000 GIVE A STRING A NAME. A NAME IS ITSELF A STRING, WHOSE VALUE IS  
 1038000 ANOTHER (NOT NECESSARILY DIFFERENT) STRING. IN GENERAL, ANY  
 1039000 STRING CAN BE USED AS A NAME, BY USING INDIRECTION (SEE THE NEXT  
 1040000 SECTION). NAMES ARE ASSIGNED VALUES IN TWO DIFFERENT WAYS: BY AN  
 1041000 "ASSIGNMENT", AND BY USING THE NAME WITH A STRING VARIABLE. SOME  
 1042000 EXAMPLES OF ASSIGNMENT STATEMENTS ARE:  
 1043000  
 1044000 X = "1234"  
 1045000 I = J  
 1046000 ALPHA = A B (C+D) FF("=")  
 1047000 N = .GT(N,"0") N = "1"  
 1048000  
 1049000 ANOTHER WAY TO GIVE A STRING NAME A VALUE IS TO ATTACH THE NAME TO  
 1050000 A STRING VARIABLE IN A PATTERN. IF THE PATTERN MATCH SUCCEEDS, THE  
 1051000 SUBSTRING MATCHED BY THE STRING VARIABLE WILL BECOME THE NEW  
 1052000 VALUE OF THE STRING NAME. SOME EXAMPLES ARE:  
 1053000  
 1054000 "1234" \*X\*  
 1055000 TR "," \*I\* "/" \*J\* ","  
 1056000 C \*CH/"1"\* \*C\*  
 1057000 CARD \*R/"72"\* \*SEQ\*  
 1058000 EXP \*(E1)\*  
 1059000  
 1060000 NOTE THAT THE FIRST EXAMPLE IN THIS SET MEANS THE SAME AS THE FIRST  
 1061000 EXAMPLE IN THE PREVIOUS SET.  
 1062000 NAMES THAT ARE USED IN SNOBOL CODE MUST SATISFY THE RESTRICTION  
 1063000 OF BEING AN "IDENTIFIER"--A NON-NULL STRING OF LETTERS, DIGITS, AND  
 1064000 PERIODS. ANY OTHER STRING WHICH IS TO BE USED AS A NAME MUST BE  
 1065000 HANDLED BY INDIRECTION.  
 1066000  
 1067000  
 1068000  
 1069000 %CP 40  
 1070000 5.1.3. INDIRECTION.  
 1071000  
 1072000 SINCE NAMES AND VALUES IN SNOBOL3 ARE BOTH STRINGS, IT IS USEFUL  
 1073000 TO BE ABLE TO CONNECT THE TWO. THIS IS DONE BY MEANS OF "INDIRECTION",  
 1074000 WHICH CONSISTS OF TREATING THE VALUE OF A STRING AS A NAME. THE  
 1075000 OPERATOR WHICH DOES THIS IS "\$", WHICH IS A UNARY PREFIX OPERATOR.  
 1076000 WHEN "\$" IS APPLIED TO AN EXPRESSION, THE VALUE OF THE RESULTING  
 1077000 EXPRESSION IS THE VALUE OF THE STRING NAMED BY THE ORIGINAL EXPRES-

1078000 SIGN. PUT ANOTHER, MORE INTUITIVE WAY, THE "\$" OPERATOR REPLACES  
1079000 ITS ARGUMENT WITH THE THING THAT IT NAMES. A THIRD INTERPETATION IS:  
1080000 INDIRECTION REMOVES ONE LAYER OF QUOTES FROM AROUND ITS ARGUMENT,  
1081000 FOR EXAMPLE, IF A = "B", B = "C", AND C = "D", THEN \$A = \$"B" = B  
1082000 = "C". SIMILARLY, \$\$A = \$B = C = "D". FOR ANOTHER EXAMPLE, SUPPOSE  
1083000 THAT I = "37" AND C37 = "NEW YORK". THEN:

1084000  
1085000       \$( "C" I ) = "NEW YORK"

1086000  
1087000 FOR YET ANOTHER EXAMPLE, IT WILL BE SHOWN HOW "ARRAYS" CAN BE HANDLED  
1088000 IN SNOBOL3, A LANGUAGE THAT DOESNT HAVE ARRAYS AS SUCH AMONG ITS  
1089000 DATA TYPES. ARRAY HANDLING CAN BE SIMULATED, HOWEVER, BY USING  
1090000 INDIRECTION ON AN EXPRESSION THAT CONTAINS THE "ARRAY NAME" AND  
1091000 ALL THE RELEVANT SUBSCRIPTS, SEPARATED BY ANY APPROPRIATE  
1092000 DELIMITERS. TO LOOK ALGOL-LIKE, THE DELIMITERS CAN BE BRACKETS  
1093000 AND COMMAS, THUS, TO USE THE I, J, K ELEMENT OF THE ARRAY AB, THE  
1094000 FOLLOWING EXPRESSION COULD BE USED:

1095000  
1096000       \$( "AB[ " I " , " J " , " K " ] " )

1097000  
1098000 IF I = "3", J = "-2", AND K = "47", THIS WOULD GIVE:

1099000  
1100000       AB[ 3 , -2 , 47 ]

1101000  
1102000 SOME FURTHER EXAMPLES OF EXPRESSIONS INVOLVING INDIRECTION ARE:

1103000  
1104000       \$I  
1105000       \$\$C  
1106000       \$\$\$\$"  
1107000       \$"A=I1"  
1108000       \$( \$X "/" \$B "/" \$( "N" I ) )  
1109000       \$\$F(A, "3")  
1110000       \$\$SIZE(E)  
1111000       \$( "FATHER/" NAME )  
1112000       \$( "NAME-" SOC. SEC. NO. )

1113000  
1114000  
1115000  
1116000

1117000 %CP 45

1118000 5.2. CONCATENATION.

1119000

1120000 THE BASIC STRING OPERATOR IS CONCATENATION, WHICH SIMPLY JOINS ITS  
1121000 ARGUMENTS END-TO-END. CONCATENATION IS ASSUMED IN SNOBOL3 WHEN TWO  
1122000 STRINGS (LITERALS, NAMES, EXPRESSIONS, ETC.) ARE PLACED NEXT TO  
1123000 EACH OTHER WITHOUT AN OPERATOR BETWEEN THEM. SINCE THE OPERATION OF  
1124000 CONCATENATION IS ASSOCIATIVE, IT CAN BE USED AS IF IT WERE AN  
1125000 OPERATION WITH ANY NUMBER OF OPERANDS, AND SO CONCATENATE EXPRES-  
1126000 SIONS CONSIST SIMPLY OF A LIST OF STRINGS TO BE CONCATENATED, WITH  
1127000 JUST BLANKS BETWEEN THEM IF A DELIMITER IS NEEDED. SOME EXAMPLES  
1128000 OF EXPRESSIONS INVOLVING CONCATENATION ARE:

1129000  
1130000       A " , " B  
1131000       (A " , " B)  
1132000       ("ABC" X) "DE"  
1133000       " " ALPHA " "  
1134000       LIST ELT "/" PROPS " , "

1412  
Micro Business Forms, Inc. sv

1135000 X1 "," X2 "," X3 "," X4 "," X5 "," X6  
 1136000  
 1137000 A "GROUPING" IS SIMPLY A STRING EXPRESSION SURROUNDED BY PAREN-  
 1138000 THESES. SOME GROUPINGS ARE INCLUDED IN THE ABOVE EXAMPLES. TO  
 1139000 EVALUATE A GROUPING, ALL THE ELEMENTS WITHIN IT ARE EVALUATED,  
 1140000 AND A TEMPORARY STORAGE LOCATION IS GIVEN THEIR CONCATENATED VALUE.  
 1141000 IF THERE IS ONLY ONE OBJECT WITHIN THE PARENTHESES, A NEW TEMPORARY  
 1142000 VALUE IS CREATED ANYWAY. THIS FACT CAN BE USED IN CASES WHERE LATER  
 1143000 PART OF THE INSTRUCTION MAY ALTER A VALUE OF A STRING, AND THE OLDER  
 1144000 VALUE IS THE ONE THAT SHOULD BE USED. FOR EXAMPLE, IF THE FUNCTION  
 1145000 F() ALTERS THE STRING X, THEN THE EXPRESSION (X F()) WILL USE THE  
 1146000 NEW VALUE IN THE CONCATENATION. TO FORCE THE GROUPING TO USE THE VALUE  
 1147000 THAT X HAD BEFORE THE CALL, THE EXPRESSION SHOULD BE:

1148000  
 1149000 ((X) F())

1150000  
 1151000 SIMILARLY, IF A PROGRAM CALLS F(READ,READ), THE SECOND PARAMETER  
 1152000 WILL ASSIGN A NEW VALUE TO READ, AND THE FIRST INPUT CARD WILL BE  
 1153000 LOST (THE SAME VALUE WILL BE PASSED TO BOTH PARAMETERS). TO CAUSE  
 1154000 THE FUNCTION TO USE THE OLDER VALUE OF ITS FIRST PARAMETER, THE  
 1155000 CALL COULD BE F((READ),READ).

1156000 IF A PARAMETER TO A FUNCTION IS AN EXPRESSION INVOLVING CONCATEN-  
 1157000 ATION, THEN AN IMPLICIT PAIR OF PARENTHESES SURROUNDS THE PARAMETER,  
 1158000 AND THE CONCATENATION IS PERFORMED BEFORE THE NEXT PARAMETER IS  
 1159000 EVALUATED. THUS IN F(READ "\*" ,READ), THE PROBLEM MENTIONED  
 1160000 IN THE PREVIOUS PARAGRAPH IS AVOIDED, SINCE THE FIRST PARAMETER  
 1161000 IS EVALUATED BEFORE THE SECOND READ IS PERFORMED, AND THE RIGHT  
 1162000 VALUE WILL BE USED FOR THE FIRST READ.

1163000 THE GROUPING () IS EQUIVALENT TO "".

1164000  
 1165000

1166000

1167000 %PAGE

1168000 5.3. FUNCTION CALLS.

1169000

1170000 <FCT CALL> ::= <FCT NAME>( <PARAMS> )

1171000 <FCT NAME> ::= <IDENTIFIER>

1172000 <PARAMS> ::= <EXPRESSION>

1173000 ::= <PARAMS> , <EXPRESSION>

1174000

1175000 FUNCTION CALLS ARE USED TO CALL FUNCTIONS. SNOBOL3 PRE-DEFINES

1176000 A LARGE NUMBER OF FUNCTIONS (SEE SECTION 9), AND PROGRAMMERS CAN

1177000 DEFINE OTHERS CODED IN SNOBOL (SEE THE FUNCTION DEFINE() IN SEC. 9

1178000 AND THE -DEFINE CARD IN SEC. 2.2).

1179000 THERE ARE THREE BASIC REASONS FOR CALLING A FUNCTION:

1180000

1181000 1) TO CALCULATE AND RETURN A VALUE.

1182000 2) TO TEST A CONDITION, AND SUCCEED OR FAIL.

1183000 3) TO CAUSE "SIDE EFFECTS"--THAT IS, TO ALTER GLOBAL INFORMATION

1184000 THAT IS TO BE USED BY OTHER PARTS OF THE PROGRAM, OR TO DO

1185000 INPUT OR OUTPUT.

1186000

1187000 A SNOBOL FUNCTION CAN DO ANY COMBINATION OF THESE THINGS.

1188000 THE LATTER IS ESPECIALLY EASY IN SNOBOL, SINCE A PROGRAM-DEFINED

1189000 FUNCTION HAS ACCESS TO ANY VARIABLES IN THE PROGRAM THAT ARE NOT

1190000 EXPLICITLY DECLARED TO BE LOCAL TO IT (THE LIST OF FORMAL PARAMETERS

1191000 AND LOCAL VARIABLES). THIS MEANS ESPECIALLY THAT A FUNCTION GENERALLY

1192000 HAS ACCESS TO ANY OF THE DATA IN USE BY THE PIECE OF CODE THAT CALLED  
 1193000 IT, WHICH MAY ITSELF BE PART OF A HIGHER-LEVEL FUNCTION.  
 1194000 FOR DETAILS OF WHAT HAPPENS WHEN A USER-DEFINED FUNCTION IS  
 1195000 CALLED, SEE SECTION 3.2. SOME SPECIAL DETAILS CONCERNING THE CALL  
 1196000 ITSELF ARE:

1197000  
 1198000 1) THE "(" MUST APPEAR IMMEDIATELY AFTER THE FUNCTION NAME. IF THERE  
 1199000 IS A BLANK BETWEEN THEM, THE COMPILER WILL INTERPRET IT AS THE  
 1200000 CONCATENATION OF A STRING WITH A GROUPING.  
 1201000 2) OMITTED PARAMETERS ARE NULL STRINGS. EXTRA PARAMETERS ARE EVAL-  
 1202000 UATED BUT NOT PASSED TO THE FUNCTION. FOR EXAMPLE:  
 1203000 F(X) = F("",X)  
 1204000 MW(A,B,C) = MW(A,"",B,C,"")  
 1205000 IN THE SECOND EXAMPLE, THE LAST COMMA IS NOT NEEDED.  
 1206000 3) IF THE FUNCTION BEING CALLED IS NOT DEFINED, THE PROGRAM HAS  
 1207000 COMMITTED A FATAL ERROR. IF IN DEBUG MODE, THE PROGRAM WILL  
 1208000 BE SUSPENDED WITH AN EXPLANATORY MESSAGE. IF NOT IN DEBUG  
 1209000 MODE, A MESSAGE GIVING THE STATEMENT NUMBER AND THE FUNCTION  
 1210000 NAME WILL BE PRINTED ON THE LINE PRINTER AND TELETYPE, AND THE  
 1211000 JOB WILL BE TERMINATED ABNORMALLY.

1212000  
 1213000 SOME EXAMPLES OF FUNCTION CALLS ARE:

1214000  
 1215000 MV("A",B L,(", " X ","))  
 1216000 ARRGH(,B " " X1, "CARD")  
 1217000 LEVEL()  
 1218000 .NE(A,"375")  
 1219000 EQUALS(STR,RES)

1220000  
 1221000 NOTE ESPECIALLY THE FOLLOWING:

1222000  
 1223000 F(A) IS A FUNCTION CALL.  
 1224000 F (A) IS A CONCATENATION.  
 1225000 F(B "",",","1") IS A FUNCTION CALL.  
 1226000 F (B "",",","1") IS A SYNTAX ERROR. (WHY?)  
 1227000  
 1228000  
 1229000

1230000 %PAGE  
 1231000 5.4. ARITHMETIC.

1232000  
 1233000 <ARITH EXPR> ::= <A-TERM>  
 1234000 ::= - <A-TERM>  
 1235000 <A-TERM> ::= <A-TERM> <ADD OP> <M-TERM>  
 1236000 ::= <M-TERM>  
 1237000 <M-TERM> ::= <M-TERM> <MUL OP> <E-TERM>  
 1238000 ::= <E-TERM>  
 1239000 <E-TERM> ::= <E-TERM> <EXP OP> <ARITH ELT>  
 1240000 ::= <ARITH ELT>  
 1241000 <ARITH ELT> ::= <NAMED EXPR>  
 1242000 ::= <GROUPING>  
 1243000 ::= <FCT CALL>  
 1244000 ::= <LITERAL>  
 1245000 <ADD OP> ::= +  
 1246000 ::= -  
 1247000 <MUL OP> ::= \*  
 1248000 ::= x

1249000 ::= /  
 1250000 <EXP OP> ::= \*\*  
 1251000  
 1252000 A NUMBER IN SNOBOL3 IS A STRING WHICH CONSISTS ONLY OF DIGITS,  
 1253000 WITH THE POSSIBLE EXCEPTION OF THE FIRST CHARACTER WHICH CAN BE "-".  
 1254000 ONLY INTEGER ARITHMETIC IS PROVIDED IN SNOBOL3. SOME EXAMPLES  
 1255000 OF NUMERIC STRINGS ARE:

1256000  
 1257000 "12" = "00012"  
 1258000 "-3" = "-00000003"  
 1259000 "0" = "0000" = "-0" = "-" = ""

1260000  
 1261000 SOME IMPLEMENTATIONS OF SNOBOL3 INSIST THAT ARITHMETIC BE FULLY  
 1262000 PARENTHESESIZED. THIS IS NOT NECESSARY ON THE B5500, THOUGH USING  
 1263000 UNPARENTHESESIZED ARITHMETIC MAY MAKE A B5500 SNOBOL3 PROGRAM  
 1264000 INCOMPATIBLE WITH ANOTHER IMPLEMENTATION. THE PRECEDENCE RELATIONS  
 1265000 ARE BASICALLY THOSE OF FORTRAN OR ALGOL, WITH THE RANKING:

1266000  
 1267000 EXPONENTIATION \*\* (HIGH)  
 1268000 MULTIPLICATION, DIVISION \*, x, /  
 1269000 ADDITION, SUBTRACTION +, -  
 1270000 CONCATENATION BLANKS (LOW)

1271000  
 1272000 AMONG OPERATORS OF EQUAL PRECEDENCE, LEFT-ASSOCIATION IS USED. SOME  
 1273000 ARITHMETIC EXPRESSIONS, WITH PARENTHESESIZED EQUIVALENTS, ARE:

1274000  
 1275000 A + B = C = ((A + B) = C)  
 1276000 A / B \* C = ((A / B) \* C)  
 1277000 A = B \* C = (A = (B \* C))  
 1278000 = X / R = (= (X / R))  
 1279000 A x B \*\* "2" = (A x (B \*\* "2"))  
 1280000 AL AR = AM = (AL (AR = AM))  
 1281000 C + I "0" = ((C + I) "0")  
 1282000 P \*\* Q \*\* R = ((P \*\* Q) \*\* R) = P \*\* (Q x R)

1283000  
 1284000  
 1285000

1286000 %CP 25  
 1287000 5.4.1. DIVISION MODES.  
 1288000

1289000 THE B5500 HAS THREE WAYS TO HANDLE DIVISION, DEPENDING ON HOW  
 1290000 REMAINDERS ARE TREATED. THE METHOD USED IS SET BY THE FUNCTION  
 1291000 MODE(), AS FOLLOWS:

1292000  
 1293000 MODE("ROUND") CAUSES DIVISION TO BE ROUNDED.  
 1294000 MODE("TRUNCATION") CAUSES REMAINDERS TO BE DISCARDED.  
 1295000 MODE("INTEGER") CAUSES DIVISION TO FAIL IF THE REMAINDER ≠ 0.  
 1296000

1297000 THE SYSTEM ASSUMES ROUNDING UNLESS MODE() IS USED, SINCE THAT IS  
 1298000 HOW THE HARDWARE OPERATES, AND IS FASTEST. SOME EXAMPLES OF THE  
 1299000 DIFFERENCES BETWEEN THE THREE ARE:

EXPRESSION	ROUNDED	TRUNCATED	INTEGER
"4" / "2"	"2"	"2"	"2"
"5" / "2"	"3"	"2"	FAILS
"5" / "3"	"2"	"1"	FAILS
"12" / "4"	"3"	"3"	"3"

1300000  
 1301000  
 1302000  
 1303000  
 1304000  
 1305000



1306000	"13" / "4"	"3"	"3"	FAILS
1307000	"14" / "4"	"4"	"3"	FAILS
1308000	"15" / "4"	"4"	"3"	FAILS
1309000	"-3" / "2"	"-2"	"-1"	FAILS
1310000	"9" / "-6"	"-2"	"-1"	FAILS
1311000	"-5" / "3"	"-2"	"-1"	FAILS

1312000  
1313000  
1314000

1315000 %CP 13

1316000 5.4.2. FAILURE OF ARITHMETIC.

1317000

1318000 THE FOLLOWING CONDITIONS WILL CAUSE FAILURE OF AN ARITHMETIC

1319000 EXPRESSION:

1320000

1321000 1) A NON-NUMERIC ARGUMENT. A STRING IS NUMERIC IF ITS FIRST CHAR-  
1322000 ACTER IS "-" OR A DIGIT AND ALL OTHER CHARACTERS ARE DIGITS.

1323000 BOTH "" AND "-" ARE NUMERIC, AND BOTH EQUAL ZERO.

1324000 2) A NUMBER OCCURS WHOSE MAGNITUDE IS GREATER THAN 2\*\*39-1, THE LARGEST

1325000 INTEGER THAT THE B5500 CAN HANDLE. THE DECIMAL VALUE OF THIS

1326000 NUMBER IS 549,755,813,887.

1327000 3) DIVISION BY ZERO.

1328000 4) THE DIVISION MODE IS "INTEGER" AND DIVISION PRODUCES A NON-ZERO  
1329000 REMAINDER.

1330000

1331000

1332000

1333000 %CP 30

1334000 5.4.3. ARITHMETIC FUNCTIONS.

1335000

1336000 THIS SECTION DESCRIBES A NUMBER OF FUNCTIONS PRE-DEFINED BY  
1337000 SNOBOL WHICH ARE USED WITH NUMBERS. ALL OF THESE FUNCTIONS WILL

1338000 FAIL IF A NON-NUMERIC PARAMETER IS GIVEN THEM.

1339000

1340000 .EQ(A,B) SUCCEEDS IF A AND B REPRESENT THE SAME INTEGER. THIS

1341000 FUNCTION SHOULD BE COMPARED WITH EQUALS(A,B), WHICH DOES A

1342000 STRING COMPARE. SOME EXAMPLES SHOWING THE DIFFERENCES ARE:

1343000

1344000 .EQ("1","01") SUCCEEDS                      EQUALS("1","01") FAILS

1345000 .EQ("00","") SUCCEEDS                      EQUALS("00","") FAILS

1346000 .EQ("A","A") FAILS                      EQUALS("A","A") SUCCEEDS

1347000 .EQ("7","7") SUCCEEDS                      EQUALS("7","7") SUCCEEDS

1348000

1349000 .EQ() ALWAYS RETURNS A NULL VALUE, AS DO THE FOLLOWING:

1350000

1351000 .NE(A,B) SUCCEEDS IF A ≠ B.

1352000 .LT(A,B) SUCCEEDS IF A < B.

1353000 .LE(A,B) SUCCEEDS IF A ≤ B.

1354000 .GT(A,B) SUCCEEDS IF A > B.

1355000 .GE(A,B) SUCCEEDS IF A ≥ B.

1356000 .NUM(A) SUCCEEDS IF A IS NUMERIC.

1357000

1358000 .REMDR(A,B) RETURNS THE REMAINDER OF A / B. THIS IS DEFINED AS:

1359000     A = ((A / B) × B)

1360000     WHERE THE DIVISION IS TRUNCATED.

1361000

1362000

1363000 %CP 10  
 1364000  
 1365000 5.4.4. ADDITIONAL COMMENTS ON ARITHMETIC.  
 1366000  
 1367000 THERE ARE TWO WAYS TO INDICATE MULTIPLICATION, WITH "\*" OR  
 1368000 WITH "x". WHEN USED IN A PATTERN OUTSIDE OF PARENTHESES, "\*" WILL  
 1369000 ALWAYS BE ASSUMED TO BE A STRING VARIABLE ASTERISK, SO EITHER  
 1370000 "x" OR PARENTHESES SHOULD BE USED IF MULTIPLICATION IS DESIRED.  
 1371000 ALSO, IF "\*\*\*" APPEARS OUTSIDE OF PARENTHESES IN A PATTERN, IT WILL  
 1372000 BE AN UNNAMED ARBITRARY STRING VARIABLE, NOT EXPONENTIATION.  
 1373000 NOTE THAT THE ONLY UNARY ARITHMETIC OPERATOR IS "-".  
 1374000 BLANKS ARE NOT NEEDED AROUND ARITHMETIC OPERATORS IN B5500  
 1375000 SNOBOL3, UNLIKE IN MANY OTHER IMPLEMENTATIONS OF THE LANGUAGE.

1376000  
 1377000  
 1378000

1379000 %SECTION 6  
 1380000 %PAGE 1  
 1381000 6. PATTERNS.

1382000  
 1383000 <PATTERN> ::= <ELEMENT>  
 1384000 ::= <STR VAR>  
 1385000 ::= <PATTERN> <PATTERN>  
 1386000 ::= <PATTERN> <BACK REF>  
 1387000 <STR VAR> ::= <ARB VAR>  
 1388000 ::= <BAL VAR>  
 1389000 ::= <FIXED-LENGTH VAR>  
 1390000 <ARB VAR> ::= \*\*  
 1391000 ::= \* <NAMED EXPR> \*  
 1392000 <BAL VAR> ::= \*()\*  
 1393000 ::= \*( <NAMED EXPR> )\*  
 1394000 <FIXED-LENGTH VAR> ::= \*/ <EXPRESSION> \*  
 1395000 ::= \* <NAMED EXPR> / <EXPRESSION> \*  
 1396000 <BACK REF> ::= <NAMED EXPR>

1397000  
 1398000 PROBABLY THE MOST IMPORTANT PART OF A SNOBOL INSTRUCTION IS THE  
 1399000 PATTERN, WHICH IS USED TO IDENTIFY AND ALTER PARTS OF STRINGS ACCORDING  
 1400000 TO CERTAIN PROPERTIES OF THE STRINGS BEING EXAMINED. A PATTERN  
 1401000 IS MADE UP OF A LIST OF "PATTERN ELEMENTS", EACH OF WHICH WILL  
 1402000 "MATCH" A PART OF A STRING IF THAT PART HAS THE PROPERTY DESCRIBED BY  
 1403000 THE PATTERN ELEMENT. A PATTERN WILL MATCH A STRING IF EACH OF ITS  
 1404000 ELEMENTS MATCHES A SUBSTRING, AND THE MATCHED SUBSTRINGS ARE ADJACENT.  
 1405000 IN OTHER WORDS, THE PATTERN IS MATCHED ONE ELEMENT AT A TIME, FROM  
 1406000 LEFT TO RIGHT, AND EACH PATTERN ELEMENT MUST MATCH A SUBSTRING THAT  
 1407000 STARTS JUST AFTER THE SUBSTRING MATCHED BY THE PREVIOUS ELEMENT.  
 1408000 THE BASIC PATTERN ELEMENTS DEFINED IN SNOBOL3 ARE CONSTANTS, STRING  
 1409000 VARIABLES OF THREE TYPES (ARBITRARY, BALANCED, AND FIXED-LENGTH),  
 1410000 AND BACK REFERENCES. THE FOLLOWING SUB-SECTIONS DESCRIBE EACH OF  
 1411000 THESE PATTERN ELEMENTS IN DETAIL.

1412000  
 1413000  
 1414000

1415000 %CP 20  
 1416000 6.1. CONSTANT PATTERN ELEMENTS.

1417000  
 1418000 A "CONSTANT" PATTERN ELEMENT IS ONE WHICH WILL MATCH ONLY ONE  
 1419000 STRING. THAT IS, IT IS AN ELEMENT WHICH EVALUATES TO A STRING OF

14137  
 Moore Business Forms, Inc. sv

1420000 CHARACTERS, AND IT WILL MATCH ONLY THAT STRING. EXAMPLES OF CONSTANT  
1421000 PATTERN ELEMENTS ARE LITERALS, STRING NAMES, FUNCTION CALLS, AND  
1422000 ANY OTHER STRING-VALUED EXPRESSION. FOR EXAMPLE, THE PATTERN

1423000

1424000 A "+" (B C)

1425000

1426000 CONSISTS OF THREE CONSTANT PATTERN ELEMENTS: A, "+", AND (B C).

1427000 IF A = "IS", B = "3", AND C = "K", THEN THIS PATTERN WILL MATCH

1428000 THE STRING "IS+3K". IF EXP = "SQRT(S)=IS+3K-J", THEN THE RULE

1429000

1430000 EXP A "+" (B C)

1431000

1432000 WILL SUCCEED, WITH THE PATTERN MATCHING STARTING AT THE 9TH CHAR-

1433000 ACTER OF THE STRING REFERENCE.

1434000

1435000

1436000

1437000 %CP 45

1438000 6.2. STRING VARIABLES.

1439000

1440000 A STRING VARIABLE IS A PATTERN ELEMENT WHICH WILL MATCH MORE THAN

1441000 ONE STRING, DEPENDING ON THE PROPERTIES OF THE STRING. THERE ARE

1442000 THREE TYPES IN SNOBOL3: "ARBITRARY", "BALANCED", AND "FIXED-LENGTH".

1443000 SEPARATE SUB-SECTIONS BELOW ARE DEVOTED TO EACH TYPE. THIS SECTION

1444000 DESCRIBES SOME PROPERTIES COMMON TO ALL TYPES OF STRING VARIABLES.

1445000 A STRING VARIABLE CAN BE "NAMED" OR "UNNAMED". A NAMED VARIABLE

1446000 IS ONE WHICH CONTAINS A <NAMED EXPR> (SEE THE BNF DEFINITIONS).

1447000 BOTH FORMS ACT THE SAME DURING A PATTERN MATCH; THE DIFFERENCE IS

1448000 THAT IF THE PATTERN MATCH IS SUCCESSFUL, THE STRING NAMED BY THE

1449000 THE STRING VARIABLES ARE ASSIGNED THE MATCHED SUBSTRINGS AS NEW

1450000 VALUES. FOR EXAMPLE, IF X = "1234567", THEN THE RULE

1451000

1452000 X "2" \*\* "5"

1453000

1454000 WILL SIMPLY SUCCEED AND HAVE NO SIDE-EFFECT, WHILE

1455000

1456000 X "2" \*I\* "5"

1457000

1458000 WILL SUCCEED AND ASSIGN TO I THE VALUE "34", WHICH IS THE SUBSTRING

1459000 MATCHED BY THE STRING VARIABLE \*I\*.

1460000 THE ASSIGNMENT OF STRING VARIABLE VALUES IS CARRIED OUT IMMEDIATELY

1461000 AFTER THE PATTERN MATCH, BEFORE THE REPLACEMENT PART (IF ANY) IS

1462000 EVALUATED, SO THE NEW VALUES WILL BE USED IF THE STRING VARIABLE

1463000 NAMES APPEAR IN THE REPLACEMENT. FOR EXAMPLE, IF L = "A, LN, AR, XYZ, CB,"

1464000 THEN THE RULE

1465000

1466000 L ", " \*R\* ", " \*S\* ", " ← RPL(R, S)

1467000

1468000 WILL DO A PATTERN MATCH (WHICH WILL SUCCEED), SET R = "LN", SET

1469000 S = "AR", AND THEN CALL RPL("LN", "AR"), WHOSE VALUE WILL THEN REPLACE

1470000 THE MATCHED SUB-STRING ", LN, AR,".

1471000 THERE MAY BE SEVERAL STRING VARIABLES WITH THE SAME NAME. THE

1472000 ASSIGNMENTS ARE CARRIED OUT IN A STRICT LEFT-TO-RIGHT ORDER, SO

1473000 EACH STRING WILL END UP WITH THE LAST VALUE ASSIGNED TO IT. FOR

1474000 EXAMPLE, WITH L AS ABOVE, THE RULE

1475000

1476000 L ", " \*X\* ", " \*X\* ", " \*X\* ", "

1477000 WILL SUCCEED AND GIVE X THE VALUES "LN", "AR", AND "XYZ". THE LAST  
1478000 VALUE, "XYZ", WOULD BE THE VALUE OF X AT THE END OF THE INSTRUCTION.  
1479000 IF X WERE AN OUTPUT STRING, OUTPUT WOULD OCCUR THREE TIMES  
1480000 DURING THIS RULE.

1481000  
1482000  
1483000

1484000 %CP 40  
1485000 6.2.1. ARBITRARY STRING VARIABLES.

1486000  
1487000 <ARB VAR> ::= \*\*  
1488000 ::= \* <NAMED EXPR> \*  
1489000

1490000 AN ARBITRARY STRING VARIABLE WILL MATCH ANY STRING (INCLUDING  
1491000 THE NULL STRING). AT THE BEGINNING OF A PATTERN (THE LEFT END),  
1492000 AN ARBITRARY VARIABLE MATCHES THE LEFT END OF THE STRING REFERENCE,  
1493000 AT THE RIGHT END OF A PATTERN, IT MATCHES THE "REST" OF THE STRING  
1494000 REFERENCE. FOR EXAMPLE, IF SENT = "WHEN I SAW HER ", THEN  
1495000

1496000 SENT \*A\* " " \*B\* " " \*C\*

1497000  
1498000 WILL SUCCEED, AND ASSIGN A = "WHEN", B = "I", AND C = "SAW HER".  
1499000 NOTE ESPECIALLY THAT AN ARBITRARY STRING VARIABLE WILL MATCH A NULL  
1500000 STRING IF POSSIBLE, SO FOR EXAMPLE, IF X = "++", THEN

1501000  
1502000 X \*I\* "+" \*J\* "+"  
1503000

1504000 WILL SUCCEED AND SET BOTH I AND J TO "".  
1505000 IF SEVERAL ARBITRARY STRING VARIABLES ARE ADJACENT TO EACH OTHER,  
1506000 ALL BUT THE LAST ONE WILL ALWAYS MATCH THE NULL STRING. AN EXAMPLE  
1507000 OF A USE FOR THIS IS:

1508000  
1509000 STR \*V\* \*\* "-"  
1510000

1511000 THIS INSTRUCTION WILL SEARCH STR FOR A MINUS SIGN, AND IF IT FINDS  
1512000 ONE, IT WILL SET V TO NULL. FOR ANOTHER EXAMPLE, TO ASSIGN A TO  
1513000 B AND SIMULTANEOUSLY SET I, J, K, L, AND M NULL, THE FOLLOWING  
1514000 STATEMENT COULD BE USED:

1515000  
1516000 A \*I\* \*J\* \*K\* \*L\* \*M\* \*B\*  
1517000

1518000 THE GENERAL RULE FOR MATCHING AN ARBITRARY VARIABLE IS:

- 1519000
- 1520000 1) MATCH THE PRECEDING ELEMENT, IF THERE IS ONE, AS FAR TO THE  
1521000 LEFT AS POSSIBLE.
  - 1522000 2) MATCH THE FOLLOWING ELEMENT, IF THERE IS ONE, AS CLOSE AFTER  
1523000 THE PRECEDING ONE AS POSSIBLE.
  - 1524000 3) THE ARBITRARY VARIABLE THEN MATCHES EVERYTHING BETWEEN THE STRINGS  
1525000 MATCHED BY THE SURROUNDING ELEMENTS, OR IF EITHER OF THESE IS  
1526000 MISSING, THE VARIABLE EXTENDS TO THE APPROPRIATE END OF THE STRING.

1527000  
1528000  
1529000

1530000 %CP 45  
1531000 6.2.2. BALANCED STRING VARIABLES.

1532000  
1533000 <BAL VAR> ::= \*( )\*

```

1534000 ::= *( <NAMED_EXPR> )*
1535000
1536000 A BALANCED STRING VARIABLE WILL MATCH ANY STRING WHICH IS NON=NULL
1537000 AND HAS BALANCED PARENTHESES. THAT IS, THERE MUST BE THE SAME NUMBER
1538000 OF RIGHT PARENTHESES AS THERE ARE LEFT PARENTHESES (POSSIBLY NONE
1539000 OF EITHER), AND THE FIRST PARENTHESIS MUST BE A "(", THUS A BALANCED
1540000 VARIABLE WILL MATCH ANY OF THE FOLLOWING:
1541000
1542000 "ABCD"
1543000 " "
1544000 "("
1545000 "A(BLMX)+D"
1546000 "A + (B - C)"
1547000 "(F(1.2) + (ALPHA ** 2))"
1548000 "HE SAID (VERY SOFTLY) THAT "
1549000 "((( ))(( ))(( ))(( )))"
1550000
1551000 A BALANCED VARIABLE WILL NOT MATCH ANY OF:
1552000
1553000 ""
1554000 "("
1555000 ")(("
1556000 "A+B)"
1557000 "(A)"
1558000 "VERY SOFTLY) THAT "
1559000 " ) + (A = M) "
1560000
1561000 UNLIKE ARBITRARY STRING VARIABLES, BALANCED VARIABLES ARE NOT
1562000 EXTENDED TO THE RIGHT WHEN THEY END A PATTERN, THOUGH AT THE START
1563000 OF A PATTERN THEY EXTEND AS FAR LEFT AS POSSIBLE. FOR EXAMPLE, IF
1564000
1565000 EXPR = "(A+(LIM(P/Q,Q,1)-SIN(LN(1-ALPH)))x(2**N)x3)-N"
1566000
1567000 THEN THE RULE
1568000
1569000 EXPR *(L)* "x" *(R)*
1570000
1571000 WILL SUCCEED, AND SET L = "A+(LIM(P/Q,Q,1)-SIN(LN(1-ALPH)))",
1572000 AND R = "(2**N)". L WILL INCLUDE THE INITIAL "A+", BUT R
1573000 WILL NOT BE EXTENDED TO INCLUDE THE "x3", THOUGH "(2**N)x3"
1574000 WOULD BE A VALID MATCH FOR A BALANCED VARIABLE.
1575000
1576000
1577000
1578000 %CP 30
1579000 6.2.3. FIXED-LENGTH STRING VARIABLES.
1580000
1581000 <FIXED-LENGTH VAR> ::= */ <EXPRESSION> *
1582000 ::= * <NAMED_EXPR> / <EXPRESSION> *
1583000
1584000 A FIXED-LENGTH STRING VARIABLE WILL MATCH ANY STRING WITH A GIVEN
1585000 NUMBER OF CHARACTERS. THE <EXPRESSION> MUST BE NUMERIC AND NON-
1586000 NEGATIVE, OR THE PATTERN WILL FAIL DURING EVALUATION AND THE PATTERN
1587000 MATCH WILL NOT TAKE PLACE. ONE SORT OF USE OF A FIXED-LENGTH VARIABLE
1588000 IS TO PICK OFF INITIAL STRINGS, USUALLY ONE CHARACTER:
1589000
1590000 STR *CH/"1"* = /F(OUT)

```

1591000  
1592000 THIS WILL ASSIGN TO CH THE FIRST CHARACTER OF STR AND DELETE THAT  
1593000 CHARACTER FROM STR. IF STR IS NULL, THE STATEMENT WILL FAIL AND  
1594000 TRANSFER TO OUT. FOR ANOTHER EXAMPLE, TO GET THE LAST N CHARACTERS  
1595000 OF A STRING S:  
1596000  
1597000 S \*/SIZE(S)-N\* \*TAIL\* /F(TOO,SMALL)  
1598000  
1599000 THE UNNAMED FIXED-LENGTH VARIABLE WILL MATCH ALL BUT THE LAST N  
1600000 CHARACTERS OF S, AND THE ARBITRARY VARIABLE WILL GET THE LAST N.  
1601000 IF S HAS FEWER THAN N CHARACTERS, THE FIXED-LENGTH VARIABLE WILL  
1602000 HAVE A NEGATIVE LENGTH, AND WILL FAIL DURING EVALUATION OF THE PATTERN.  
1603000 NOTE THAT THE FIRST EXAMPLE ABOVE COULD BE WRITTEN:  
1604000  
1605000 STR \*CH/"1" \*STR\* /F(OUT)  
1606000  
1607000  
1608000  
1609000 %CP 30  
1610000 6.3. BACK REFERENCES.  
1611000  
1612000 ANOTHER TYPE OF PATTERN ELEMENT IS THE "BACK REFERENCE". THIS  
1613000 IS A STRING NAME WHICH IS THE SAME AS THE NAME OF A PREVIOUS STRING  
1614000 VARIABLE. A BACK REFERENCE WILL MATCH A STRING WHICH IS IDENTICAL  
1615000 TO THE STRING MATCHED BY THE STRING VARIABLE. FOR EXAMPLE, IN  
1616000  
1617000 X \*H\* "E" H  
1618000  
1619000 THE BACK REFERENCE, H, MUST MATCH THE SAME STRING AS THE ARBITRARY  
1620000 VARIABLE, \*H\*. IF X = "ABEBCA", THE RULE WILL SUCCEED AND SET H = "B".  
1621000 SIMILARLY, IF EXP IS AN ALGEBRAIC EXPRESSION, THEN  
1622000  
1623000 EXP \*(P)\* \*\* P :F(NONE)  
1624000  
1625000 WILL LOOK FOR A SUB-EXPRESSION THAT OCCURS TWICE, AND IF THERE ARE  
1626000 NONE, IT WILL TRANSFER TO NONE.  
1627000 THERE CAN BE SEVERAL BACK REFERENCES TO THE SAME STRING VARIABLE.  
1628000 EACH SUCH BACK REFERENCE MUST MATCH THE SAME STRING. FOR EXAMPLE  
1629000  
1630000 X \*S/"1"\* S S  
1631000 WILL SUCCEED ONLY IF X CONTAINS THREE CONSECUTIVE IDENTICAL CHARACTERS.  
1632000 IF THERE ARE SEVERAL STRING VARIABLES WITH THE SAME NAME, A BACK  
1633000 REFERENCE WILL MATCH THE LAST SUCH VARIABLE THAT PRECEDES IT. FOR  
1634000 EXAMPLE, THE RULE  
1635000  
1636000 AL \*X\* " " \*X\* " " X  
1637000  
1638000 WILL SUCCEED FOR THE STRING AL = "AB CD CDEF", AND SET X = "CD". THE  
1639000 FIRST STRING VARIABLE WILL MATCH THE STRING "AB".  
1640000  
1641000  
1642000  
1643000 %CP 40  
1644000 6.4. PATTERN MATCH MODES.  
1645000  
1646000 NORMALLY, A PATTERN WILL MATCH ANYWHERE IN THE STRING REFERENCE  
1647000 THAT IT FINDS AN APPROPRIATE SUB-STRING. SOMETIMES IT IS DESIRABLE TO

Moore Business Forms, Inc. sv 14137

1648000 FORCE THE PATTERN TO MATCH STARTING AT THE FIRST CHARACTER OF THE  
1649000 STRING REFERENCE. A PATTERN THAT CAN ONLY MATCH AN INITIAL SUB-STRING  
1650000 IS CALLED "ANCHORED". PATTERNS THAT CAN MATCH ANYWHERE (THE USUAL KIND)  
1651000 ARE CALLED "UNANCHORED".

1652000 THERE ARE TWO METHODS OF INDICATING WHETHER OR NOT A PATTERN IS TO BE  
1653000 ANCHORED. ONE IS BY USING THE MODE FUNCTION: MODE("ANCHOR") SETS A  
1654000 SWITCH AT CAUSES ALL PATTERNS TO BE ANCHORED; MODE("UNANCH") RESETS  
1655000 THE SWITCH SO THAT PATTERNS ARE UNANCHORED. EITHER OF THESE MODES  
1656000 CAN BE CANCELED FOR A SINGLE INSTRUCTION BY USING THE FUNCTIONS  
1657000 ANCHOR() AND UNANCH(). ANCHOR() CAUSES THE INSTRUCTION THAT IT IS  
1658000 IN TO BE ANCHORED, NO MATTER WHAT THE MODE IS. SIMILARLY, UNANCH()  
1659000 CAUSES ITS PATTERN TO BE UNANCHORED, NO MATTER WHAT THE MODE. IF  
1660000 MODE() IS NOT USED, SNOBOL ASSUMES THAT THE MODE IS UNANCHORED.

1661000 TO SHOW THE DIFFERENCE, SUPPOSE THAT E = "(A+B)(". THEN

1662000

1663000 E \*(F)\*

1664000

1665000 WILL SUCCEED, AND SET F = "(A+B)". HOWEVER, THE RULE

1666000

1667000 E ANCHOR() \*(F)\*

1668000

1669000 WILL FAIL, SINCE THE PATTERN IS RESTRICTED TO MATCHING AN INITIAL  
1670000 SUBSTRING, AND NO BALANCED STRING CAN START WITH ")". IF MODE("ANCHOR")  
1671000 HAD BEEN CALLED, THEN THE FIRST RULE WOULD ALSO HAVE FAILED.

1672000 FOR ANOTHER EXAMPLE, SUPPOSE X = "ABCDE". THEN THE RULE

1673000

1674000 X "C" \*I\*

1675000

1676000 WILL SUCCEED (IF THE MODE IS UNANCHORED), AND SET I = "DE". BUT  
1677000 THE ANCHORED RULE:

1678000

1679000 X ANCHOR() "C" \*I\*

1680000

1681000 WILL FAIL, BECAUSE THE FIRST CHARACTER OF X IS NOT "C".

1682000

1683000

1684000

1685000 %CP 50

1686000 6.5. ASSIGNMENT OF STRING VARIABLES.

1687000

1688000 IF A PATTERN MATCH SUCCEEDS, THE STRING NAMES ASSOCIATED WITH  
1689000 ITS STRING VARIABLES ARE ASSIGNED THE VALUES OF THEIR SUBSTRINGS  
1690000 IN A STRICT LEFT-TO-RIGHT ORDER. THIS MEANS THAT IF SEVERAL STRING

1691000 VARIABLES HAVE THE SAME NAME, THE RIGHT-MOST ONE WILL DETERMINE

1692000 THE VALUE ASSIGNED TO THE NAME; THE OTHERS WILL BE ASSIGNED THEIR

1693000 APPROPRIATE VALUES, AND THEN RE-ASSIGNED. IF THE STRINGS HAVE ANY

1694000 OUTPUT USE, OUTPUT WILL OCCUR FOR EACH ASSIGNMENT. FOR EXAMPLE,

1695000 IF THE RULE

1696000

1697000 X \*PRINT\* "," \*PRINT\* "," \*PRINT\*

1698000

1699000 SUCCEEDS, THE THREE STRINGS ASSIGNED TO PRINT WILL ALL BE WRITTEN

1700000 ON THE LINE PRINTER.

1701000 FURTHERMORE, THE ASSIGNMENT IS DONE IMMEDIATELY AFTER THE PATTERN

1702000 MATCH, BEFORE THE REPLACEMENT AND GO-TO PARTS ARE EVALUATED, TO ANY

1703000 STRING NAMES USED, SAY, BOTH AS A STRING VARIABLE NAME AND IN THE

1704000 REPLACEMENT WILL HAVE THE NEW VALUE WHEN THE REPLACEMENT IS EVALUATED.

1705000 FOR EXAMPLE, IF AST = "XZ,AB", THEN IN THE RULE

1706000

1707000           AST "." \*L\* = "." DBL(I,L)

1708000

1709000 THE VALUE PASSED TO THE SECOND PARAMETER OF DBL() WILL BE "AB".

1710000       A RATHER INTERESTING SITUATION OCCURS WHEN THE STRING REFERENCE

1711000 HAS THE SAME NAME AS A STRING VARIABLE. FOR EXAMPLE, THE RULE

1712000

1713000           I \*I\* "+" \*J\*

1714000

1715000 WILL FIRST ASSIGN A NEW VALUE TO I, THEN ATTEMPT TO ASSIGN A VALUE TO J.

1716000 BUT BY THIS TIME, I HAS BEEN CHANGED, AND I WAS THE STRING REFERENCE.

1717000 THERE IS CLEARLY NO LONGER ANYTHING MEANINGFUL TO ASSIGN TO J, AND

1718000 SUCH INSTRUCTIONS SHOULD BE AVOIDED. IN GENERAL, J WOULD BE EXPECTED

1719000 TO CONTAIN RATHER ARBITRARY GARBAGE. A SIMILAR RULE THAT ACHIEVES

1720000 WHAT MOST PEOPLE WOULD ASSUME TO BE THE BEST MEANING OF THIS RULE IS

1721000

1722000           (I) \*I\* "+" \*J\*

1723000

1724000 IN THIS CASE, THE STRING REFERENCE IS NOT I ITSELF, BUT A GROUPING

1725000 THAT HAS THE SAME VALUE AS I. SINCE THE STRING VARIABLES ARE NO

1726000 LONGER NAMED THE SAME AS THE STRING REFERENCE (THE STRING REFERENCE

1727000 HAS NO NAME AT ALL), THE PROBLEM DOESNT ARISE.

1728000       IT IS AT TIMES USEFUL TO HAVE THE LAST STRING VARIABLE IN A

1729000 PATTERN NAMED THE SAME AS THE STRING REFERENCE. FOR EXAMPLE, THE

1730000 FOLLOWING TWO STATEMENTS ARE EQUIVALENT:

1731000

1732000           ST \*WD\* " " = /F(E)

1733000           ST \*WD\* " " \*ST\* /F(E)

1734000

1735000

1736000

1737000 %CP 35

1738000 6.6. PATTERN MATCH ALGORITHM.

1739000

1740000       THERE ARE USUALLY MANY WAYS FOR A PATTERN MATCH TO SUCCEED, WHEN

1741000 THIS IS TRUE, THERE IS A CLEAR NEED FOR A WAY TO DETERMINE WHICH

1742000 OF THE POSSIBLE MATCHES WILL BE THE ONE CHOSEN BY SNOBOL, SO THE

1743000 PROGRAMMER CAN PREDICT WHAT VALUES WILL BE ASSIGNED TO THE STRING

1744000 VARIABLES AND WHICH SUBSTRING WILL BE REPLACED.

1745000       AS A GENERAL, INTUITIVE RULE, THE PROGRAMMER SHOULD REMEMBER THAT

1746000 A PATTERN MATCH PROCEEDS FROM LEFT TO RIGHT, AND EACH ELEMENT IS MATCHED

1747000 WITH THE SHORTEST SUBSTRING POSSIBLE CONSISTENT WITH THE SUCCESS OF THE

1748000 REST OF THE PATTERN. THAT IS, THE PREFERRED MATCH WILL BE THE ONE

1749000 FOR WHICH AN "EARLIER" PATTERN ELEMENT MATCHES AS FAR LEFT AS POSSIBLE.

1750000       THE GENERAL RULES GOVERNING A PATTERN MATCH ARE:

1751000

1752000 1) TRY TO MATCH THE FIRST PATTERN ELEMENT STARTING AT THE FIRST CHAR-

1753000       ACTER OF THE STRING REFERENCE. AN ARBITRARY VARIABLE STARTS

1754000       BY MATCHING THE NULL STRING. IF THIS MATCH FAILS, GO TO STEP 5.

1755000 2) IF THERE ARE NO MORE ELEMENTS, GO TO STEP 6.

1756000 3) IF THERE IS A NEXT ELEMENT, TRY TO MATCH IT AT THE CHARACTER

1757000       AFTER THE SUBSTRING MATCHED BY THE PRECEDING ELEMENT. IF THIS

1758000       SUCCEEDS, GO TO STEP 2.

1759000 4) IF THERE IS NO PREVIOUS ELEMENT, GO TO STEP 5. IF

1760000       THERE IS A PREVIOUS ELEMENT, TRY TO EXTEND IT TO MATCH A LONGER

1761000       STRING (ONLY ARBITRARY AND BALANCED STRING VARIABLES CAN BE



1762000 EXTENDED). IF THIS SUCCEEDS, GO BACK TO STEP 3 AND TRY TO RE-MATCH  
 1763000 THE CURRENT ELEMENT. IF THE EXTENSION CANT BE MADE, GO TO STEP 4.  
 1764000 5) IF THE PATTERN IS ANCHORED, THE PATTERN MATCH FAILS. IF IT IS  
 1765000 NOT ANCHORED, THEN TRY TO FIND ANOTHER MATCH FOR THE FIRST ELEMENT  
 1766000 STARTING AT A LATER CHARACTER. IF THIS CANT BE DONE, THE MATCH  
 1767000 FAILS. IF IT CAN BE DONE, GO TO STEP 2.  
 1768000 6) IF THE LAST PATTERN ELEMENT IS AN ARBITRARY VARIABLE, EXTEND IT  
 1769000 TO INCLUDE THE REST OF THE STRING REFERENCE. THE PATTERN MATCH  
 1770000 HAS SUCCEEDED, AND THE ASSIGNMENT TO THE STRING VARIABLE NAMES  
 1771000 CAN BE MADE.

1772000  
 1773000  
 1774000

1775000 %SECTION 7

1776000 %PAGE 1

1777000 7. GO-TO PARTS, LABELS.

1778000

1779000 <GO-TO PART> ::= <GO-TO DELIMITER> <GO-TOS>

1780000 <GO-TO DELIMITER> ::= <BLANKS> /

1781000 ::= :

1782000 <GO-TOS> ::= <UNCONDITIONAL GO-TO>

1783000 ::= <SUCCESS GO-TO>

1784000 ::= <FAILURE GO-TO>

1785000 ::= <SUCCESS GO-TO> <FAILURE GO-TO>

1786000 ::= <FAILURE GO-TO> <SUCCESS GO-TO>

1787000 <UNCONDITIONAL GO-TO> ::= ( <LABEL EXPR> )

1788000 <SUCCESS GO-TO> ::= S( <LABEL EXPR> )

1789000 <FAILURE GO-TO> ::= F( <LABEL EXPR> )

1790000 <LABEL EXPR> ::= <RESERVED LABEL>

1791000 ::= <LABEL>

1792000 ::= <COMPUTED LABEL>

1793000 <RESERVED LABEL> ::= END

1794000 ::= RETURN

1795000 ::= FRETURN

1796000 <COMPUTED LABEL> ::= \$ <ELEMENT>

1797000

1798000

1799000

1800000 %CP 10

1801000 7.1 LABELS.

1802000

1803000 ANY INSTRUCTION IN A SNOBOL PROGRAM CAN BE LABELED. A LABEL IS

1804000 A NON-NULL STRING PRECEDING THE FIRST BLANK OF AN INSTRUCTION. IT

1805000 FOLLOWS FROM THIS DEFINITION THAT LABELS CANT CONTAIN BLANKS. IF

1806000 THE FIRST CHARACTER OF AN INSTRUCTION IS A BLANK, THE INSTRUCTION

1807000 IS UNLABELED. ASIDE FROM THE SPECIAL CHARACTERS IN COLUMN ONE, THERE

1808000 ARE NO RESTRICTIONS ON THE CHARACTERS THAT CAN BE USED IN LABELS.

1809000

1810000

1811000

1812000 %CP 45

1813000 7.2. SIMPLE GO-TO PARTS.

1814000

1815000 THE GO-TO PART OF AN INSTRUCTION DETERMINES WHICH INSTRUCTION IS

1816000 TO BE EXECUTED NEXT. IF THERE IS NO GO-TO PART, OR IF THE GO-TO

1817000 PART DOESNT APPLY WHEN THE INSTRUCTION IS EXECUTED, THEN THE NEXT

1818000 INSTRUCTION TO BE EXECUTED IS THE NEXT ONE IN THE PROGRAM.

1819000 BRANCHING IN SNOBOL3 CAN BE EITHER UNCONDITIONAL OR CONDITIONED  
1820000 ON THE SUCCESS OR FAILURE OF THE INSTRUCTION. THE GO-TO PART CAN  
1821000 BE SET OFF BY EITHER A "/" (WHICH MUST BE PRECEDED BY A BLANK) OR BY  
1822000 A ":". MANY IMPLEMENTATIONS OF SNOBOL3 ONLY USE THE "/", USUALLY  
1823000 BECAUSE THEIR CHARACTER SETS DONT INCLUDE ":". THE B5500 WILL  
1824000 ACCEPT EITHER DELIMITER.  
1825000 A SIMPLE GO-TO PART CONSISTS OF A LABEL WITHIN PARENTHESES.  
1826000 FOR EXAMPLE:  
1827000  
1828000       :(L)               MEANS "GO TO L".  
1829000       :S(X1)            MEANS "IF THE RULE SUCCEEDED, GO TO X1".  
1830000       :F(AX)            MEANS "IF THE RULE FAILED, GO TO AX".  
1831000       :S(BY)F(AGAIN) MEANS "IF SUCCEEDED, GO TO BY, OTHERWISE GO  
1832000    TO AGAIN".  
1833000        /F(E)S(BACK)    MEANS "IF FAILED, GO TO E, OTHERWISE GO TO BACK".  
1834000        /S(BLXGG)      MEANS "IF SUCCEEDED, GO TO BLXGG".  
1835000        /(A+T\$)        MEANS "GO TO A+T\$".  
1836000  
1837000 THE B5500 ALLOWS THE USER TO OMIT THE "S" OR "F" ON THE SECOND PART  
1838000 WHEN BOTH SUCCESS AND FAILURE EXITS ARE USED. THUS  
1839000  
1840000       :S(A)(B)            MEANS :S(A)F(B)  
1841000        /F(L1)(L2)       MEANS /F(L1)S(L2)  
1842000  
1843000 THERE ARE A FEW RESTRICTIONS ON LABELS THAT CAN BE USED IN A SIMPLE  
1844000 GO-TO PART. THESE VARY FROM ONE IMPLEMENTATION TO ANOTHER; ON THE  
1845000 B5500 THEY ARE:  
1846000  
1847000       1) THE FIRST CHARACTER MUST BE A LETTER OR A DIGIT.  
1848000       2) THE LABELS CANT CONTAIN PARENTHESES.  
1849000  
1850000 FOR EXAMPLE, THE FOLLOWING LABELS, THOUGH VALID LABELS, CANT BE  
1851000 USED IN SIMPLE GO-TO PARTS:  
1852000  
1853000       AB)  
1854000       +25  
1855000       \$\$A  
1856000       AB(2)  
1857000  
1858000  
1859000  
1860000  
1861000 %CP 20  
1862000 7.3. RESERVED LABELS.  
1863000  
1864000 THERE ARE THREE "RESERVED LABELS" THAT CAN BE USED IN SIMPLE  
1865000 GO-TO PARTS. THESE ARE NOT REALLY LABELS, BUT RATHER COMMANDS  
1866000 TO PERFORM CERTAIN ACTIONS. THEY ARE:  
1867000  
1868000       END            CAUSES THE PROGRAM TO BE TERMINATED NORMALLY.  
1869000       RETURN        CAUSES A RETURN FROM THE LAST PROGRAMMER-DEFINED FUNCTION  
1870000    WHICH WAS CALLED. THE RETURN IS SUCCESSFUL.  
1871000       FRETURN       CAUSES A RETURN ALSO, BUT THE FUNCTION FAILS.  
1872000  
1873000 A PROGRAM CAN USE RETURN AND FRETURN AS LABELS, BUT THEY CANT BE  
1874000 TRANSFERED TO BY A SIMPLE GO-TO; A COMPUTED GO-TO MUST BE USED.  
1875000 IF THE PROGRAMMER DOESNT DEFINE THESE LABELS, THE COMPILER WILL

1876000 DEFINE THEM AS FOLLOWS:

1877000

1878000 RETURN : (RETURN)

1879000 FRETURN : (FRETURN)

1880000

1881000

1882000

1883000 %CP 12

1884000 7.4. COMPUTED GO-TOS.

1885000

1886000 A PROGRAM CAN COMPUTE THE LABEL THAT AN INSTRUCTION IS TO TRANSFER

1887000 TO BY USING A CONSTRUCT THAT LOOKS EXACTLY LIKE INDIRECTION INSTEAD

1888000 OF A LABEL IN A GO-TO PART. THE VALUE OF THE EXPRESSION FOLLOWING

1889000 THE "\$" IS THE LABEL TRANSFERRED TO. FOR EXAMPLE, IF X = "1" AND

1890000 Y = "AL", THEN:

1891000

1892000 /S(\$("A" X)) MEANS /S(A1)

1893000 :F(\$Y) MEANS :F(AL)

1894000 :S(\$ (X Y)) F(\$ (Y X)) MEANS :S(1AL) F(AL1)

1895000 :(\$("M" X "." SIZE(Y))) MEANS :(M1,2)

1896000

1897000

1898000

1899000 %CP 15

1900000 7.5. ERRORS IN GO-TO PARTS.

1901000

1902000 THERE ARE TWO FATAL ERRORS THAT CAN OCCUR IN A GO-TO PART, IF

1903000 A PROGRAM ATTEMPTS TO TRANSFER TO A LABEL THAT ISNT DEFINED, THE

1904000 PROGRAM IS TERMINATED ABNORMALLY WITH A MESSAGE GIVING THE ERROR,

1905000 THE STATEMENT NUMBER, AND THE ERRONEOUS LABEL.

1906000 IF A COMPUTED GO-TO CONTAINS A FUNCTION CALL, AND THE FUNCTION

1907000 FAILS, THE PROGRAM IS AGAIN TERMINATED ABNORMALLY, AND A MESSAGE

1908000 IS PRINTED SAYING THAT THE GO-TO PART FAILED, GIVING THE STATEMENT

1909000 NUMBER.

1910000 OF COURSE, IF THE USER IS USING DEBUG MODE, THE PROGRAM WILL NOT

1911000 BE TERMINATED. INSTEAD, THE ERROR MESSAGES WILL BE PRINTED ON THE

1912000 LINE PRINTER AND TELETYPE AND THE PROGRAM WILL BE SUSPENDED.

1913000

1914000

1915000

1916000 %SECTION 8

1917000 %PAGE 1

1918000 8. FAILURE.

1919000

1920000 AN EXTREMELY IMPORTANT CONCEPT IN SNOBOL IS THAT OF "FAILURE".

1921000 A SNOBOL INSTRUCTION CAN BE VIEWED AS A COMMAND TO ATTEMPT CERTAIN

1922000 ACTIONS; IF THE INSTRUCTION CANT BE CARRIED OUT, IT IS TERMINATED

1923000 AT THE POINT WHERE IT HAS TO BE STOPPED. THE GO-TO PART OF AN INST-

1924000 RUCTION PROVIDES THE PROGRAMMER WITH A WAY TO BRANCH ON THE SUCCESS

1925000 OR FAILURE OF THE INSTRUCTION. MOST BRANCHING IN SNOBOL PROGRAMS

1926000 IS DONE THIS WAY; THE OTHER METHOD (COMPUTING THE LABEL OF THE

1927000 NEXT INSTRUCTION) IS USUALLY RATHER RARE.

1928000 IT SHOULD BE EMPHASIZED THAT FAILURE CAN OCCUR AT VIRTUALLY ANY

1929000 PART OF AN INSTRUCTION, AND WHEN FAILURE OCCURS, THE INSTRUCTION

1930000 TERMINATES IMMEDIATELY AND THE FAILURE EXIT (IF ONE EXISTS) IS TAKEN.

1931000 THERE ARE TWO RATHER VITAL POINTS NECESSARY TO SUCCESSFUL USE OF

1932000 FAILURE: WHAT CAUSES FAILURE, AND WHAT WILL BE LEFT UNDONE WHEN

1933000 AN INSTRUCTION FAILS AT A PARTICULAR POINT.  
1934000  
1935000  
1936000  
1937000 %CP 18  
1938000 8.1. CAUSES OF FAILURE.  
1939000  
1940000 THE FOLLOWING CONDITIONS CAUSE FAILURE OF AN INSTRUCTION:  
1941000  
1942000 1) FAILURE OF THE PATTERN MATCH.  
1943000 2) FUNCTION FAILURE. INTRINSIC FUNCTIONS FAIL WHEN THEIR DESCRIPTIONS  
1944000 SAY THEY SHOULD (ASSUMING NO BUGS); PROGRAM-DEFINED FUNCTIONS  
1945000 FAIL WHEN THEY USE FRETURN IN A GO-TO PART.  
1946000 3) FAILURE OF ARITHMETIC. THERE ARE SEVERAL REASONS FOR THIS:  
1947000 NON-NUMERIC ARGUMENTS, VALUES TOO BIG FOR THE HARDWARE TO HANDLE;  
1948000 DIVISION BY ZERO, DIVISION WITH INTEGER MODE THAT GIVES A  
1949000 NON-ZERO REMAINDER.  
1950000 4) I/O FAILS WHEN THE OPERATION CANT BE COMPLETED. FOR FILES, THIS  
1951000 MEANS END-OF-FILE OR PARITY ERRORS. FOR TELETYPES, FAILURE IS  
1952000 CAUSED BY EXCEEDING THE WAIT TIME OR USING THE BREAK KEY DURING  
1953000 OUTPUT, OR FOR VARIOUS ABNORMAL CONDITIONS.  
1954000 5) A FIXED-LENGTH VARIABLE FAILS DURING EVALUATION IF ITS LENGTH  
1955000 IS NON-NUMERIC OR NEGATIVE.  
1956000  
1957000  
1958000  
1959000 %CP 25  
1960000 8.2. ORDER OF EXECUTION.  
1961000  
1962000 IN ORDER TO KNOW HOW MUCH WILL BE DONE WHEN AN INSTRUCTION FAILS,  
1963000 IT IS NECESSARY TO KNOW THE EXACT ORDER IN WHICH AN INSTRUCTION IS  
1964000 EXECUTED. THE RULES GOVERNING THE ORDER OF EXECUTION ARE:  
1965000  
1966000 1) ALL EXPRESSIONS ARE EVALUATED FROM LEFT TO RIGHT. FUNCTIONS ARE  
1967000 CALLED AS SOON AS ALL PARAMETERS HAVE BEEN EVALUATED. ARITHMETIC  
1968000 OPERATORS ARE APPLIED AS SOON AS BOTH OPERANDS HAVE BEEN EVALUATED.  
1969000 2) THE STRING REFERENCE IS THE FIRST PART OF A RULE THAT IS EVALUATED.  
1970000 3) AFTER THE STRING REFERENCE, THE PATTERN IS EVALUATED FROM LEFT TO  
1971000 RIGHT. NAMES OF STRING VARIABLES ARE EVALUATED WHEN THE STRING VAR-  
1972000 IABLE IS ENCOUNTERED.  
1973000 4) AFTER THE PATTERN IS EVALUATED, THE PATTERN MATCH TAKES PLACE.  
1974000 5) AFTER THE PATTERN MATCH, STRING VARIABLES (IF ANY) ARE ASSIGNED  
1975000 THEIR MATCHING SUBSTRINGS AS NEW VALUES. FAILURE CAN OCCUR HERE  
1976000 IF A STRING VARIABLE NAME IS AN OUTPUT STRING, AND THE OUTPUT  
1977000 OPERATION FAILS.  
1978000 6) THE REPLACEMENT PART (IF ONE EXISTS) IS THEN EVALUATED.  
1979000 7) THE REPLACEMENT OF THE MATCHED SUBSTRING THEN OCCURS.  
1980000 8) LASTLY, THE APPROPRIATE PART OF THE GO-TO PART IS EVALUATED, AND THE  
1981000 PROGRAM TRANSFERS TO THE NEXT INSTRUCTION. FAILURE OF A GO-TO PART  
1982000 IS A FATAL ERROR.  
1983000  
1984000  
1985000  
1986000 %SECTION 9  
1987000 %PAGE 1  
1988000 9. INTRINSIC FUNCTIONS.  
1989000 THIS SECTION DESCRIBES THE FUNCTIONS THAT ARE PRE-DEFINED BY THE

Macre Business Forms, Inc. sv 14117

1990000 B5500 SNOBOL3 SYSTEM. OF COURSE, THE NAMES OF THESE FUNCTIONS ARE IN  
1991000 NO SENSE "RESERVED", AND A PROGRAM CAN RE-DEFINE ANY OF THEM IF THE  
1992000 USER WISHES TO USE THEIR NAMES FOR HIS OWN FUNCTIONS. THE FUNCTIONS  
1993000 ARE LISTED IN ALPHABETICAL ORDER.  
1994000  
1995000  
1996000 %CP 8  
1997000 ANCHOR(N) WHEN USED IN A PATTERN, CAUSES THE PATTERN TO "ANCHORED"  
1998000 AT CHARACTER N. THAT IS, THE FIRST PATTERN ELEMENT MUST MATCH  
1999000 A SUBSTRING THAT STARTS AFTER AN INITIAL SUBSTRING OF N CHARACTERS.  
2000000 THE USUAL FORM, ANCHOR(), THEREFORE FORCES THE PATTERN TO MATCH AN  
2001000 INITIAL SUBSTRING. N MUST BE A NON-NEGATIVE INTEGER, OR ANCHOR(N)  
2002000 WILL FAIL. NOTE THAT MOST IMPLEMENTATIONS OF SNOBOL3 DO NOT HAVE  
2003000 A PARAMETER FOR ANCHOR(); IF A PROGRAM IS TO BE USED ON OTHER  
2004000 SYSTEMS, THE PARAMETER SHOULD NEVER BE USED.  
2005000  
2006000  
2007000 %CP 17  
2008000 ASSIGN(A,B) IS A SPECIAL FAST, "EFFICIENT" ASSIGNMENT FUNCTION THAT  
2009000 TAKES SPECIAL ADVANTAGE OF THE B5500 STRING REPRESENTATION. THIS  
2010000 FUNCTION CAN GENERALLY BE ASSUMED TO BE LACKING IN OTHER VERSIONS  
2011000 OF SNOBOL3. IT IS EQUIVALENT TO THE FOLLOWING TWO STATEMENTS:  
2012000  
2013000       A ← B  
2014000       B ←  
2015000  
2016000 NO CHECKING OF PARAMETERS IS DONE, INCLUDING CHECKING FOR OUTPUT USE.  
2017000 (CHECKS FOR INPUT USE OCCUR BEFORE THE FUNCTION IS CALLED, SO THEY  
2018000 WILL NOT BE SUPPRESSED.) THE FUNCTION SIMPLY MOVES POINTERS AROUND  
2019000 WITHOUT LOOKING AT WHAT THEY POINT TO. NOTE THE FOLLOWING WARNINGS:  
2020000 IF EITHER PARAMETER IS A LITERAL, THE VALUE OF THAT LITERAL WILL BE  
2021000 DESTROYED (THROUGHOUT THE PROGRAM)--LITERALS SHOULD NEVER EVER EVER  
2022000 BE USED AS PARAMETERS TO ASSIGN(). ALSO, ALTHOUGH B CAN BE OMITTED  
2023000 (IT WILL THEN BE ASSUMED NULL), A MUST BE PRESENT, OR THE FUNCTION  
2024000 WILL FAIL.  
2025000  
2026000  
2027000 %CP 13  
2028000 CALLF(F,P1,P2,P3,...) CALLS THE FUNCTION NAMED BY F, WITH PARAMETERS  
2029000 P1, P2, P3, ETC. CALLF RETURNS THE VALUE RETURNED BY THIS FUNCTION,  
2030000 OR FAILS IF THE FUNCTION FAILS. F CAN NAME ANY FUNCTION, INTRINSIC  
2031000 OR PROGRAM-DEFINED. FOR EXAMPLE:  
2032000  
2033000       CALLF("FX",B,"1",AL)                   MEANS   FX(B,"1",AL)  
2034000       CALLF("SIZE",B)                        MEANS   SIZE(B)  
2035000       CALLF("CALLF","MODE","INTEGER") MEANS   MODE("INTEGER")  
2036000  
2037000 IF CALLF IS BEING TRACED, NO OUTPUT WILL APPEAR AT ITS RETURN. IT IS  
2038000 NECESSARY TO TRACE THE FUNCTION BEING CALLED TO GET THE RETURN VALUE.  
2039000 IF THE FUNCTION IS UNDEFINED, AN ABNORMAL TERMINATION WILL OCCUR AS  
2040000 WITH ALL OTHER UNDEFINED FUNCTION CALLS.  
2041000  
2042000  
2043000 %CP 24  
2044000 CLEAR()  
2045000 THIS FUNCTION SETS ALL STRINGS TO NULL. THIS IS ONLY  
2046000 DONE TO THE TOP VALUE ON A PUSH-DOWN STACK, SO VALUES THAT

```

2047000 HAVE BEEN PUSHED DOWN WILL NOT BE LOST. AS AN EXAMPLE OF ITS USE,
2048000 THE FOLLOWING FUNCTION CLEARS ALL STRINGS EXCEPT THOSE WHOSE NAMES
2049000 ARE ON A CERTAIN LIST (WHICH IS THEN CALCULATED BY THE PROGRAM,
2050000 OF COURSE):
2051000
2052000 *****
2053000 =DEFINE "RESET(L)", , "N"
2054000 * L IS A LIST OF NAMES OF VARIABLES. THE NAMES ARE SEPARATED BY
2055000 * SEMICOLONS (";"), AND L SHOULD END WITH A SEMICOLON. ALL THE
2056000 * VARIABLES LISTED IN L ARE PUSHED, EVERYTHING IS SET NULL, AND
2057000 * THE VARIABLES ARE THEN POPPED TO THEIR FORMER VALUES.
2058000 * FOR EXAMPLE, RESET("ALPHA;RESULTS;MEM;") WILL SET EVERYTHING TO
2059000 * NULL EXCEPT FOR ALPHA, RESULTS, AND MEM. NOTE THAT THE STRINGS
2060000 * RESET, L, AND N WILL ALSO BE SAVED.
2061000 RESET PUSH(L,L)
2062000 RESET1 L *N* ";" = :F(RESET2)
2063000 PUSH($N) :(RESET1)
2064000 RESET2 CLEAR()
2065000 POP(L)
2066000 RESET3 L *N* ";" = :F(RETURN)
2067000 POP($N) :(RESET3)
2068000 *****
2069000
2070000
2071000
2072000 %CP 24
2073000 CLOSE(F,V) CLOSES THE FILE ASSOCIATED WITH THE I/O STRING $F, IN
2074000 THE MANNER INDICATED BY V. CLOSE FAILS IF $F IS NOT A FILE I/O
2075000 STRING OR V ISNT ONE OF THE VALUES LISTED BELOW:
2076000
2077000 "SAVE" CLOSES THE FILE AND SAVES THE I/O UNIT FOR THE PROGRAM.
2078000 DISK FILES CREATED BY THE PROGRAM ARE LOST; OTHER DISK FILES
2079000 REMAIN ON THE DISK. TAPES ARE REWOUND AND THE OPERATOR IS
2080000 TOLD TO REMOVE AND SAVE THEM.
2081000 "LOCK" CLOSES AND LOCKS THE FILE. DISK FILES CREATED BY THE
2082000 PROGRAM ARE ENTERED IN THE DISK DIRECTORY, AND ARE SAVED.
2083000 TAPES ARE REWOUND AND LOCKED (THE UNIT IS "NOT READY" UNTIL
2084000 THE OPERATOR READIES IT).
2085000 "RELEASE" CLOSES THE FILE AND RELEASES THE UNIT AND ALL BUFFER
2086000 AREAS TO THE SYSTEM. TAPES ARE REWOUND AND MARKED AS SCRATCH.
2087000 DISK FILES CREATED BY THE PROGRAM ARE LOST.
2088000 "PURGE" CLOSES, PURGES, AND RELEASES THE FILE. DISK FILES ARE
2089000 REMOVED FROM THE DIRECTORY IF THEIR SECURITY STATUS ALLOWS
2090000 THE USER TO PURGE THEM. (IF HE CANT, THE MCP WILL KILL THE
2091000 PROGRAM.)
2092000 "*" IS USED ONLY WITH TAPE FILES. FOR INPUT FILES, THE END-OF-FILE
2093000 FOR THE CURRENT FILE IS LOCATED, AND THE TAPE IS POSITIONED
2094000 JUST BEYOND IT. FOR OUTPUT FILES, AN END-OF-FILE IS WRITTEN.
2095000 IN EITHER CASE, THE TAPE IS NOT REWOUND AND REMAINS UNDER
2096000 CONTROL OF THE PROGRAM.
2097000
2098000
2099000 %CP 12
2100000 COPY(S,N) RETURNS A STRING CONSISTING OF N COPIES OF S. FAILURE
2101000 OCCURS IF N IS NON-NUMERIC OR NEGATIVE. THIS FUNCTION IS QUITE
2102000 USEFUL AS AN AID TO FORMATTING OUTPUT. FOR EXAMPLE:
2103000

```

2104000 PRINT ← COPY(" ", "10"-SIZE(A)) A :F(TOOBIG)  
2105000  
2106000 WILL PRINT A RIGHT-JUSTIFIED IN A 10-CHARACTER FIELD.  
2107000  
2108000  
2109000  
2110000 %CP 3  
2111000 DATE() RETURNS THE CURRENT DATE IN THE FORM:  
2112000  
2113000 MM/DD/YY  
2114000  
2115000  
2116000  
2117000  
2118000 %CP 25  
2119000 DEFINE(A,B,C) CREATES A PROGRAM-DEFINED FUNCTION AT RUN-TIME.  
2120000 THE PARAMETERS DESCRIBE THE FUNCTION AS FOLLOWS:  
2121000  
2122000 A IS THE "FUNCTION PROTOTYPE", WHICH GIVES THE FUNCTION NAME  
2123000 AND ITS FORMAL PARAMETERS IN THE FORM:  
2124000  
2125000 <FCT NAME>( <PARAMS> )  
2126000  
2127000 WHERE <FCT NAME> IS THE FUNCTION NAME, AND <PARAMS> IS  
2128000 THE LIST OF FORMAT PARAMETERS (POSSIBLY NULL) SEPARATED BY  
2129000 COMMAS.  
2130000 B CONTAINS A LABEL WHICH IS THE ENTRY POINT FOR THE FUNCTION.  
2131000 IF B IS NULL, THE ENTRY POINT IS TAKEN TO SPELLED THE SAME  
2132000 AS <FCT NAME>.  
2133000 C CONTAINS A LIST OF LOCAL VARIABLES FOR THE FUNCTION, IF ANY,  
2134000 SEPARATED BY COMMAS.  
2135000  
2136000 SEE THE DESCRIPTIONS OF FUNCTION CODE AND CALLS FOR FURTHER INFO  
2137000 ON PROGRAM-DEFINED FUNCTIONS. SOME EXAMPLES OF CALLS OF DEFINE():  
2138000  
2139000 DEFINE("M(AB,AC)", "M1", "V,XY,ALPH")  
2140000 DEFINE("MAX(A,B)")  
2141000 DEFINE("MIN(MIN,X)")  
2142000 DEFINE("FCT()", "X")  
2143000 DEFINE("MRGE(LA,LB)", "MRGENTRY", "L1,L2,L3")  
2144000  
2145000  
2146000  
2147000 %CP 13  
2148000 DETACH(S,T) TURNS OFF ANY I/O USE OF THE OBJECT NAMED BY S (A STRING  
2149000 NAME, FUNCTION, OR LABEL). THE TYPE OF THE OBJECT IS INDICATED  
2150000 BY THE FIRST CHARACTER OF T AS FOLLOWS:  
2151000  
2152000 "S" MEANS THAT S NAMES A STRING.  
2153000 "F" MEANS THAT S NAMES A FUNCTION.  
2154000 "L" MEANS THAT S CONTAINS A LABEL.  
2155000  
2156000 IF T IS NULL, "S" IS ASSUMED, SINCE THE MOST COMMON USE OF THIS  
2157000 FUNCTION IS TO TURN OFF TRACING. FOR EXAMPLE, TO TERMINATE ANY  
2158000 I/O USE OF THE STRING ALPHA, EITHER DETACH("ALPHA") OR DETACH(  
2159000 "ALPHA", "S") WOULD WORK.  
2160000

2161000  
 2162000  
 2163000 %CP 5  
 2164000 DUMP(.) CAUSES A DUMP OF ALL STRINGS AND LABELS IN USE TO BE PRINTED  
 2165000 ON THE LINE PRINTER FILE, PRINT. THE OUTPUT IS IN THE SAME  
 2166000 FORM AS THAT USED FOR TRACING. FOR EACH STRING, ITS NAME AND  
 2167000 CURRENT VALUE IS PRINTED. FOR EACH LABEL, THE NUMBER OF TIMES  
 2168000 IT HAS BEEN EXECUTED IS GIVEN.  
 2169000  
 2170000  
 2171000  
 2172000 %CP 2  
 2173000 EOF(S) SUCCEEDS IF \$S IS A FILE I/O STRING AND THE LAST I/O OPERATION  
 2174000 ON THAT FILE FAILED DUE TO AN END-OF-FILE.  
 2175000  
 2176000  
 2177000  
 2178000 %CP 2  
 2179000 .EQ(A,B) SUCCEEDS IF A AND B ARE BOTH NUMERIC AND REPRESENT  
 2180000 THE SAME INTEGER.  
 2181000  
 2182000  
 2183000  
 2184000 %CP 2  
 2185000 EQUALS(A,B) SUCCEEDS IF A AND B HAVE THE SAME STRINGS AS VALUES.  
 2186000 IT CAN BE DEFINED IN SNOBOL AS:  
 2187000  
 2188000       A B       :F(FRETURN)  
 2189000       B A       :F(FRETURN)S(RETURN)  
 2190000  
 2191000  
 2192000  
 2193000 %CP 15  
 2194000 EXECUTE(S) "EXECUTES" THE STRING S. AT PRESENT, THIS FUNCTION IS  
 2195000 RATHER LIMITED, AND WORKS FOR THE FOLLOWING CASES ONLY:  
 2196000  
 2197000 1) S IS AN I/O STRING TO A FILE. THE FILE IS GIVEN TO THE MCP  
 2198000 TO EXECUTE. THIS IS THE SAME AS THE "ZIP WITH <FILE PART>"  
 2199000 CONSTRUCT OF ALGOL. FOR DETAILS OF FILE ORGANIZATION AND  
 2200000 USE OF CONTROL CARDS IN THE FILE, SEE THE ALGOL MANUAL.  
 2201000 2) S STARTS WITH A "-". THE VALUE IS PASSED TO THE SNOBOL CONTROL  
 2202000 CARD ROUTINE, AND INTERPRETED AS A CONTROL CARD. MOST OF  
 2203000 THE CONTROL CARDS ARE RATHER USELESS DURING EXECUTION, BUT  
 2204000 A FEW (LIKE -COMPILE AND -LIBRARY) CAN AT TIMES BE USEFUL.  
 2205000 3) S STARTS WITH "?". THE VALUE OF S IS PASSED TO THE MCP AS  
 2206000 AN MCP CONTROL CARD. EXECUTE(.) THEN RETURNS, AND THE MCP  
 2207000 GIVES NO INDICATION WHAT IT DID WITH THE CARD. THE STRING  
 2208000 SHOULD ALWAYS END WITH ";END."  
 2209000  
 2210000  
 2211000  
 2212000 %CP 42  
 2213000 FILL(F,MFID,FID,UNIT,DATE,REEL,CYCLE) ALTERS CERTAIN MCP INFORMATION  
 2214000 ASSOCIATED WITH THE I/O STRING \$F. THIS FUNCTION IS USUALLY  
 2215000 A NO-OP IF THE FILE IS OPEN (WHEN THE MCP WILL ALLOW CHANGES  
 2216000 TO AN OPEN FILE IS VERY UNCLEAR). IF \$F IS NOT AN I/O STRING TO  
 2217000 A FILE, FILL(.) WILL FAIL. IF ANY OF THE OTHER PARAMETERS IS NULL,



2218000 THEN THE INFORMATION ASSOCIATED WITH THAT PARAMETER IS NOT ALTERED.  
 2219000 NON-NULL PARAMETERS WILL CAUSE THEIR ASSOCIATED QUANTITIES TO BE  
 2220000 ASSIGNED THE VALUE OF THE PARAMETER. THE MEANINGS OF THE PARAMETERS  
 2221000 ARE AS FOLLOWS:

2222000  
 2223000 MFID IS THE "MULTI-FILE I.D." OF THE FILE; THAT IS, IT IS THE FIRST  
 2224000 HALF OF THE FILE NAME. FOR SINGLE-NAME CARD READER FILES, MFID  
 2225000 IS "0000000". IF SIZE(MFID) > "7", ONLY THE FIRST SEVEN  
 2226000 CHARACTERS WILL BE USED.

2227000 FID IS THE "FILE I.D." OF THE FILE--THE LAST HALF OF ITS NAME.  
 2228000 UNIT IS AN INTEGER DESCRIBING THE I/O DEVICE. THE VALID UNIT  
 2229000 NUMBERS ARE GIVEN IN TABLE 9.1. NOTE THAT FOR INPUT FILES, THE  
 2230000 ONLY DISTINCTION THE PROGRAM NEED MAKE IS BETWEEN DISK FILES  
 2231000 AND NON-DISK FILES. THUS, IF UNIT = "0" AND THE FILE IS ON  
 2232000 A MAGNETIC TAPE, THE PROGRAM WILL FIND IT CORRECTLY. FOR OUTPUT  
 2233000 FILES, HOWEVER, IT IS IMPORTANT TO GET THE RIGHT INTEGER.

2234000 DATE IS THE DATE ASSOCIATED WITH THE FILE (NORMALLY ONLY USED WITH  
 2235000 TAPES). TWO FORMS CAN BE USED: MM/DD/YY OR YYDD (WHICH IS  
 2236000 HOW IT IS STORED IN A FILE LABEL).

2237000 REEL IS THE REEL NUMBER, FOR MULTI-REEL TAPES.  
 2238000 CYCLE IS THE CYCLE NUMBER, USED WHEN SEVERAL TAPES HAVE THE SAME  
 2239000 NAME AND DATE.

2240000  
 2241000 FOR EXAMPLE, TO ASSIGN THE I/O STRING CARD TO THE DISK FILE  
 2242000 SMITH/CRD3, THE PROGRAM WOULD USE:

2243000  
 2244000       FILL("CARD","SMITH","CRD3","12")

2245000  
 2246000 TO GIVE THE FILE NEWDISK THE NAME GEO/CALIBR, YOU COULD USE:

2247000  
 2248000       FILL("NEWDISK","GEO","CALIBR")

2249000  
 2250000 NOTE THAT THE UNIT NEED NOT BE USED HERE; IT IS ONLY NEEDED WHEN A  
 2251000 DIFFERENT I/O DEVICE IS DESIRED THAN WAS USED THE LAST TIME THE FILE  
 2252000 WAS OPENED. FOR ANOTHER EXAMPLE, TO ASSIGN THE FILE READ TO THE  
 2253000 CARD READER FILE INPT, THE FOLLOWING WOULD WORK:

2254000  
 2255000       FILL("READ","0000000","INPT","0")

2256000  
 2257000

2258000  
 2259000 %PAGE

TABLE 9.1.

UNIT	DESCRIPTION
0	CARD READER/PUNCH
1	LINE PRINTER (NO BACK-UP)
2	LABELED MAGNETIC TAPE
3	UNUSED
4	LINE PRINTER OR TAPE BACK-UP
5	LABELED "DESIGNATED" FILE (?)
6	PRINTER BACK-UP TAPE
7	UNLABELED "DESIGNATED" FILE (?)
8	UNLABELED PAPER TAPE
9	UNLABELED MAGNETIC TAPE
10	RANDOM DISK FILE

2270000  
 2271000  
 2272000  
 2273000  
 2274000

2275000	11	SUPERVISORY PRINTER (SPO)
2276000	12	SERIAL DISK FILE
2277000	13	UPDATE DISK FILE
2278000	14	DATA COMMUNICATIONS
2279000	15	PRINTER BACK-UP DISK
2280000	16	PRINTER BACK-UP TAPE OR DISK
2281000	17	PRINTER OR BACK-UP DISK
2282000	18	PRINTER OR BACK-UP DISK OR BACK-UP TAPE
2283000	32	SPECIAL FORMS FILE.
2284000		
2285000		
2286000		
2287000		%PAGE
2288000		.GE(A,B) SUCCEEDS IF A AND B ARE BOTH NUMERIC AND $A \geq B$ .
2289000		
2290000		
2291000		
2292000		
2293000		.GT(A,B) SUCCEEDS IF A AND B ARE BOTH NUMERIC AND $A > B$ .
2294000		
2295000		
2296000		.LE(A,B) SUCCEEDS IF A AND B ARE BOTH NUMERIC AND $A \leq B$ .
2297000		
2298000		
2299000		
2300000		
2301000		LEVEL() RETURNS THE LEVEL OF FUNCTION CALLS WHICH THE PROGRAM
2302000		HAS REACHED. THE LEVEL COUNTER STARTS AT 1, INCREASES WHEN
2303000		A FUNCTION IS CALLED, AND DECREASES WHEN A FUNCTION RETURNS.
2304000		
2305000		
2306000		
2307000		%CP 7
2308000		LOOK(S,F) DEFINES THE STRING \$S TO BE A "NON-READING" INPUT STRING
2309000		FOR THE FILE ASSOCIATED WITH THE I/O STRING \$F. THIS MEANS THAT
2310000		USING \$S AS AN INPUT STRING WILL CAUSE THE "NEXT" RECORD TO BE
2311000		READ FROM THE FILE, BUT THIS RECORD WILL REMAIN IN THE BUFFER,
2312000		SO IT WILL BE THE NEXT ONE READ. THUS, CONSECUTIVE USES OF \$S
2313000		WILL GIVE THE SAME INPUT RECORD. THE SYSTEM DEFINES THE I/O STRING
2314000		LOOK AS: LOOK("LOOK","READ").
2315000		
2316000		
2317000		
2318000		.LT(A,B) SUCCEEDS IF A AND B ARE BOTH NUMERIC AND $A < B$ .
2319000		
2320000		
2321000		
2322000		%CP 52
2323000		MODE(V,P) SETS VARIOUS GLOBAL SYSTEM FLAGS, AS INDICATED BY V.
2324000		P IS USUALLY NOT USED, BUT FOR SOME VALUES OF V IT INDICATES A
2325000		VALUE TO BE USED WITH THE FLAG. V MUST BE ONE OF THE VALUES
2326000		BELOW, OR MODE() WILL FAIL. THE CURRENTLY RECOGNIZED VALUES
2327000		OF V ARE AS FOLLOWS:
2328000		
2329000		MODE("ANCHOR",P) CAUSES ALL PATTERNS TO BE ANCHORED AT CHARACTER P.
2330000		THAT IS, ALL PATTERNS MUST MATCH WITH THE FIRST PATTERN ELEMENT
2331000		MATCHING IMMEDIATELY AFTER THE P-TH CHARACTER. FOR P = "0" (OR

2332000 P NULL), THE PATTERN MUST MATCH AN INITIAL SUBSTRING. NOTE THAT  
 2333000 MOST VERSIONS OF SNOBOL DONT ALLOW THE EXTRA PARAMETER P, SO  
 2334000 PROGRAMS THAT ARE TO BE RUN ON OTHER SNOBOL SYSTEMS SHOULD  
 2335000 ONLY USE MODE("ANCHOR"). IF P IS NON-NUMERIC OR NEGATIVE,  
 2336000 FAILURE OCCURS.  
 2337000 MODE("UNANCH") OR MODE("UNANCHOR") RETURNS THE PATTERN MATCH MODE  
 2338000 TO ITS NORMAL, UNANCHORED MODE. THAT IS, THE FIRST ELEMENT  
 2339000 OF A PATTERN CAN MATCH STARTING ANYWHERE IN THE STRING REFERENCE.  
 2340000 MODE("DUMP") CAUSES DUMP() TO BE CALLED WHEN THE PROGRAM TERMINATES.  
 2341000 MODE("ERRDUMP") CAUSES DUMP() TO BE CALLED IF THE PROGRAM TERMINATES  
 2342000 ABNORMALLY.  
 2343000 MODE("INFORM") CAUSES WARNING DIAGNOSTIC MESSAGES TO BE PRINTED  
 2344000 WHEN CERTAIN CONDITIONS OCCUR THAT ARE VALID IN SNOBOL BUT ARE  
 2345000 OFTEN SIGNS OF PROGRAM ERRORS WHEN THEY OCCUR. SOME SUCH THINGS  
 2346000 ARE: USING NON-NUMERIC ARGUMENTS TO ARITHMETIC OPERATORS OR  
 2347000 ARITHMETIC FUNCTIONS; INDIRECTION WITH THE NULL STRING AS THE  
 2348000 STRING NAME; USING POP() TO WIPE OUT THE PUSH-DOWN STACK FOR  
 2349000 A FUNCTIONS FORMAL PARAMETERS OR LOCAL VARIABLES; USING A  
 2350000 FIXED-LENGTH VARIABLE WITH NON-NUMERIC OR NEGATIVE LENGTH;  
 2351000 ETC. THIS LIST MAY BE EXTENDED IN THE FUTURE, AS MORE SUCH  
 2352000 CONDITIONS ARE BROUGHT TO THE ATTENTIONS OF THE SYSTEM AUTHORS  
 2353000 WHICH CAN EASILY BE CHECKED BY THE SNOBOL INTERPRETER.  
 2354000 MODE("SILENCE") TERMINATES MODE("INFORM").  
 2355000 MODE("DEBUG") TURNS ON THE CHECKING FOR DEBUG MODE. THIS IS  
 2356000 THE INTERACTIVE DEBUGGING TOOL THAT IS PART OF THE INTERPRETER;  
 2357000 IT IS DESCRIBED IN SECTION 11.2.  
 2358000 MODE("RUN") TERMINATES DEBUG MODE, AND TURNS OFF THE ASSOCIATED  
 2359000 LIMIT TESTS.  
 2360000 MODE("ROUND") SAYS THAT ALL DIVISION IS TO BE ROUNDED. THAT IS,  
 2361000 IF THE REMAINDER IS  $\geq 1/2$  OF THE DIVISOR, THE QUOTIENT IS  
 2362000 INCREASED BY 1.  
 2363000 MODE("TRUNCATION") SAYS THAT DIVISION IS TO BE TRUNCATED--THE  
 2364000 REMAINDER IS TO BE DISCARDED.  
 2365000 MODE("INTEGER") SAYS THAT DIVISION IS TO FAIL IF THE REMAINDER  
 2366000 IS NOT ZERO.  
 2367000 MODE("OVERFLOW",P) WHERE \$P IS A FILE OUTPUT STRING, SAYS THAT  
 2368000 IF A STRING IS WRITTEN THAT IS LONGER THAN A RECORD, THE  
 2369000 EXTRA CHARACTERS ARE WRITTEN ON THE NEXT RECORD (THIS WILL  
 2370000 BE REPEATED IF THE REMAINING STRING IS STILL TOO LONG).  
 2371000 MODE("TRUNCATE",P) CANCELS MODE("OVERFLOW",P). THIS MEANS THAT  
 2372000 STRINGS THAT ARE TOO LONG FOR ONE RECORD WILL BE TRUNCATED,  
 2373000 AND THE TAIL END WILL NOT BE OUTPUT. THIS IS THE ASSUMED MODE  
 2374000 FOR ALL OUTPUT FILES UNLESS MODE("OVERFLOW",P) IS USED.  
 2375000  
 2376000  
 2377000  
 2378000 .NE(A,B) SUCCEEDS IF A AND B ARE BOTH NUMERIC AND  $A \neq B$ .  
 2379000  
 2380000  
 2381000  
 2382000 %CP 3  
 2383000 .NUM(A) SUCCEEDS IF A IS NUMERIC. THAT IS, THE FIRST CHARACTER  
 2384000 MUST BE A DIGIT OR "-"; ALL OTHER CHARACTERS MUST BE DIGITS.  
 2385000 THE NULL STRING IS NUMERIC, AS IS "-"; BOTH = ZERO.  
 2386000  
 2387000  
 2388000

2389000 %CP 34  
2390000 OPSYN(A,B,T) SETS THE OBJECT (STRING, FUNCTION, LABEL) NAMED  
2391000 BY A TO BE THE SAME AS THE ONE NAMED BY B. THE TYPE OF THE  
2392000 TWO OBJECTS IS GIVEN BY THE FIRST CHARACTER OF T, AS FOLLOWS:  
2393000  
2394000 "S" MEANS THAT A AND B ARE STRING NAMES.  
2395000 "F" MEANS THAT A AND B ARE FUNCTION NAMES.  
2396000 "L" MEANS THAT A AND B CONTAIN LABELS.  
2397000  
2398000 IF T IS NULL, "F" IS ASSUMED--IN ORDER TO BE COMPATIBLE WITH THE  
2399000 MANY SNOBOL SYSTEMS THAT ONLY ALLOW THIS FUNCTION TO BE APPLIED  
2400000 TO FUNCTION NAMES. SOME EXAMPLES OF THE USE OF OPSYN() ARE:  
2401000  
2402000 OPSYN("XA","XB","STR") ASSIGNS THE VALUE OF XB TO XA, AND ALSO  
2403000 GIVES XA ANY I/O USE OF XB.  
2404000 OPSYN("FL","RXA","F") DEFINES THE FUNCTION FL TO BE THE SAME  
2405000 AS THE FUNCTION RXA (INCLUDING TRACING, IF RXA IS BEING  
2406000 TRACED). IF RXA IS UNDEFINED, THEN FL IS ALSO UNDEFINED.  
2407000 OPSYN("REDO","DO","LABEL") DEFINES THE LABEL REDO TO BE THE SAME  
2408000 INSTRUCTION AS DO. ANY INSTRUCTION PREVIOUSLY LABELED REDO  
2409000 IS LOST UNLESS IT HAS BEEN GIVEN ANOTHER NAME.  
2410000  
2411000 NOTE THAT OPSYN ACTS AS AN ASSIGNMENT, NOT A PERMANENT DEFINITION,  
2412000 SO A FURTHER CHANGE IN ONE OF THE VARIABLES WILL NOT AFFECT  
2413000 THE OTHER. IF, AFTER THE FIRST EXAMPLE ABOVE, XB WERE ASSIGNED  
2414000 A NEW VALUE, XA WOULD REMAIN UNALTERED; IF TRACEL("DO") WERE  
2415000 CALLED, TRACING WOULD BE STARTED FOR DO BUT NOT FOR REDO.  
2416000 A TRIVIAL EXAMPLE OF THE USE OF OPSYN IS:  
2417000  
2418000 OPSYN("NULL","EQUALS")  
2419000  
2420000 WHICH WILL DEFINE A FUNCTION NULL() WHICH, WHEN CALLED WITH ONLY  
2421000 ONE ARGUMENT, WILL SUCCEED IF THE ARGUMENT IS NULL. THIS MAY  
2422000 BE DESIRABLE BECAUSE THE MEANING OF NULL(X) IS A BIT MORE OBVIOUS  
2423000 THAN THAT OF EQUALS(X).  
2424000  
2425000  
2426000  
2427000 %CP 16  
2428000 PAGE(N,F) CAUSES A PAGE-EJECT TO BE DONE ON THE FILE \$F, WHICH  
2429000 SHOULD BE A LINE PRINTER FILE. THE EJECTION IS TO "CHANNEL" N,  
2430000 WHICH IS A POSITION SOMEWHERE ON THE PAGE DETERMINED BY A PAPER  
2431000 TAPE CONTROL WITHIN THE PRINTER. THE EJECT WILL OCCUR ON THE  
2432000 NEXT LINE OF OUTPUT, AFTER THE OUTPUT OCCURS. N MUST BE A VALUE  
2433000 BETWEEN "0" AND "11"; "0" AND "1" ARE BOTH THE TOP OF THE NEXT  
2434000 PAGE. IF F IS OMITTED, THEN "PRINT" IS ASSUMED; IF N IS NULL,  
2435000 THE TOP OF THE PAGE ("1") WILL BE ASSUMED. THUS, TO SIMPLY DO  
2436000 A PAGE-EJECT ON THE PRINT FILE, THE FOLLOWING WOULD SUFFICE:  
2437000  
2438000 PRINT ← PAGE()  
2439000  
2440000 ACTUALLY, PAGE() IS A SPECIAL CASE OF THE FUNCTION RECORD(), AND  
2441000 IS MEANT TO BE USED ONLY WITH LINE PRINTERS (THUS THE RESTRICTION  
2442000 ON THE RANGE OF N). THE SAME EFFECT COULD BE ACHEIVED BY USING  
2443000 RECORD(F,N), WITH 1 ≤ N ≤ 11.  
2444000  
2445000

1412  
Moore Business Forms, Inc. sv

2446000  
 2447000 %CP 12  
 2448000 POP(S) REMOVES THE TOP OF THE PUSH-DOWN STACK FOR THE STRING S.  
 2449000 EVERY STRING NAME IN SNOBOL CAN BE CONSIDERED A PUSH-DOWN  
 2450000 STACK, WITH THE TOP OF THE STACK THE CURRENT VALUE OF THE  
 2451000 STRING. ONE WARNING: IF THE FORMAL PARAMETERS OR LOCAL VAR-  
 2452000 IABLES OF A FUNCTION ARE POPPED WITHIN THE FUNCTION, THE RETURN  
 2453000 FROM THE FUNCTION WILL TRY TO POP THEM AGAIN; THIS CAN CAUSE  
 2454000 SOME GRIEF IF CARE IS NOT TAKEN TO RESTORE THE STACKS TO THEIR  
 2455000 CORRECT "DEPTH" IN SUCH CASES. IN SUCH A SITUATION, THE RETURN  
 2456000 WOULD NOT ALTER THE STRING (IF THE INFORM FLAG IS ON, A MESSAGE  
 2457000 WILL BE PRINTED--SEE THE FUNCTION MODE()). POP(S) WILL FAIL  
 2458000 IF S IS NOT A STRING NAME, OR IF S CANT BE POPPED BECAUSE THERE  
 2459000 IS ONLY ONE VALUE IN ITS STACK.  
 2460000  
 2461000  
 2462000  
 2463000 %CP 8  
 2464000 PUSH(S,V) PUSHES THE VALUE V ONTO THE PUSH-DOWN STACK FOR S.  
 2465000 IF THIS IS FOLLOWED BY POP(S), S WILL RETURN TO ITS FORMER  
 2466000 VALUE. PUSH(S,V) WILL FAIL IF S ISNT A NAMED STRING. ALSO,  
 2467000 EACH CALL OF PUSH() REQUIRES ONE MORE ENTRY IN THE SYMBOL  
 2468000 TABLE; THERE IS ROOM IN THE TABLE FOR SOMEWHAT OVER 4000  
 2469000 SYMBOLES (STRINGS, LITERALS, FUNCTIONS, LABELS, TEMPORARY  
 2470000 STORAGE FOR GROUPINGS, ETC.) AND EXHAUSTING THIS SPACE WILL  
 2471000 CAUSE THE PROGRAM TO DIE ABNORMALLY.  
 2472000  
 2473000  
 2474000 %CP 8  
 2475000  
 2476000 .RANF(N) PRODUCES A PSEUDO-RANDOM NUMBER IN THE RANGE:  
 2477000  
 2478000  $1 \leq .RANF(N) \leq N$   
 2479000  
 2480000 THE GENERATOR IS ALWAYS INITIALIZED TO THE SAME NUMBER, SO  
 2481000 THE VALUES RETURNED ARE "PREDICTABLE" IN THE SENSE THAT REPEATING  
 2482000 A RUN WITH THE SAME VALUES FOR N WILL PRODUCE THE SAME SEQUENCE  
 2483000 OF NUMBERS.  
 2484000  
 2485000  
 2486000  
 2487000 %CP 10  
 2488000 RECORD(F,N) SETS THE "RECORD POINTER" FOR THE FILE ASSOCIATED  
 2489000 WITH \$F TO N. \$F MUST BE A FILE I/O STRING; N MUST BE A  
 2490000 NON-NEGATIVE NUMBER, OR RECORD(F,N) WILL FAIL. THIS FUNCTION  
 2491000 IS REALLY ONLY USEFUL FOR DISK FILES. FOR LINE PRINTER FILES,  
 2492000 IT IS THE SAME AS THE FUNCTION PAGE(); FOR OTHER NON-DISK FILES  
 2493000 RECORD() IS A NO-OP. FOR DISK FILES, USING RECORD(F,N) SAYS  
 2494000 THAT THE NEXT I/O OPERATION IS TO BE ON RECORD N. THE RECORDS  
 2495000 OF A DISK FILE ARE NUMBERED STARTING AT 0. IF N IS GREATER THAN  
 2496000 THE NUMBER OF THE LAST RECORD IN THE FILE, THE NEXT I/O OPERATION  
 2497000 WILL FAIL DUE TO END-OF-FILE.  
 2498000  
 2499000  
 2500000  
 2501000 %CP 5  
 2502000 RELEASE(A) WHERE A IS A TELETYPE TERMINAL ADDRESS, REMOVES THE

2503000 TELETYPE FROM THE LIST OF TELETYPES ATTACHED TO THE PROGRAM.  
 2504000 FURTHER I/O ON THIS TELETYPE BECOMES IMPOSSIBLE UNTIL THE PERSON  
 2505000 AT THE TELETYPE RE-ATTACHES HIMSELF. RELEASE(A) WILL FAIL IF A  
 2506000 IS NOT A VALID ADDRESS OR THE TELETYPE IS NOT ATTACHED.  
 2507000  
 2508000  
 2509000  
 2510000 %CP 4  
 2511000 REWIND(F) CAUSES THE FILE ASSOCIATED WITH \$F TO BE REWOUND. THE  
 2512000 FILE REMAINS OPEN, AND I/O CAN CONTINUE AS USUAL, STARTING  
 2513000 FROM THE FIRST RECORD OF THE FILE. FOR DISK FILES, REWIND(F)  
 2514000 IS EQUIVALENT TO RECORD("0",F).  
 2515000  
 2516000  
 2517000  
 2518000 %CP 7  
 2519000 RULES(N) RETURNS THE NUMBER OF RULES EXECUTED AT LEVEL N ( $\geq 1$ )  
 2520000 SINCE THE LAST TIME LEVEL N WAS REACHED. IF N = "0" (OR NULL)  
 2521000 THE TOTAL FOR THE ENTIRE PROGRAM IS RETURNED. IF N  $\leq$  LEVEL(),  
 2522000 THE CURRENT INSTRUCTION IS INCLUDED IN THE TOTAL. NOTE THAT IF  
 2523000 RULES(N) = "0", THEN LEVEL N HAS NEVER BEEN REACHED. IF N IS  
 2524000 NON-NUMERIC OR NEGATIVE, RULES(N) WILL FAIL.  
 2525000  
 2526000  
 2527000  
 2528000 %CP 30  
 2529000 SEARCH(FILE) WHERE \$FILE IS AN I/O STRING TO A DISK FILE, DOES A  
 2530000 SEARCH OF THE DISK DIRECTORY FOR THE FILE. IF IT IS PRESENT,  
 2531000 A DESCRIPTION OF THE FILE IS RETURNED. FAILURE OCCURS IF \$FILE  
 2532000 IS NOT A DISK FILE, OR IF THE FILE IS NOT IN THE DIRECTORY.  
 2533000 IF THE FILE IS IN THE DIRECTORY, THE RETURN VALUE IS OF THE  
 2534000 FOLLOWING FORM:  
 2535000  
 2536000       <MFID> / <FID> : <SECURITY> : <REC SIZE> : <BLCK SIZE> :  
 2537000       <EOF PNTR> : <OPEN COUNT>  
 2538000  
 2539000 THE EXTERNAL NAME OF THE FILE IS <MFID>/<FID>. THE SECURITY  
 2540000 STATUS, <SECURITY>, IS A STRING OF 3 "0"-S OR "1"-S WITH THE  
 2541000 FOLLOWING MEANING:  
 2542000  
 2543000       1ST = "1" IF THE USER CAN ALTER THE SECURITY STATUS OR  
 2544000               NAME, OR CAN REMOVE THE FILE.  
 2545000       2ND = "1" IF THE USER CAN READ FROM THE FILE.  
 2546000       3RD = "1" IF THE USER CAN WRITE ON THE FILE.  
 2547000  
 2548000 THE SIZE OF THE LOGICAL AND PHYSICAL RECORDS OF THE FILE, IN  
 2549000 CHARACTERS, ARE GIVEN BY <REC SIZE> AND <BLCK SIZE>. <EOF PNTR>  
 2550000 IS THE RELATIVE ADDRESS OF THE LAST RECORD OF THE FILE (THAT  
 2551000 IS, ONE LESS THAN THE NUMBER OF RECORDS IN THE FILE). <OPEN COUNT>  
 2552000 IS THE NUMBER OF PROGRAMS DOING I/O ON THE FILE. NOTE THAT,  
 2553000 UNLIKE ALGOL WHICH DEALS ENTIRELY WITH WORDS, THE RECORD SIZE  
 2554000 AND BLOCK SIZE ARE GIVEN TO A SNOBOL PROGRAM IN CHARACTERS,  
 2555000 IF <SECURITY> = "000", THEN <REC SIZE>, <BLCK SIZE>, ETC.  
 2556000 WILL ALL BE "0".  
 2557000  
 2558000  
 2559000

2560000 SIZE(S) RETURNS THE NUMBER OF CHARACTERS IN S. THUS, SIZE("ABC") = "3".

2561000

2562000

2563000

2564000

2565000 %CP 13

2566000 SPACE(F,N) SETS A COUNTER FOR THE FILE ASSOCIATED WITH \$F TO N, SO

2567000 THAT EACH I/O OPERATION ON THE FILE IS PRECEDED BY SKIPPING N

2568000 RECORDS (ON INPUT) OR WRITING N BLANK RECORDS (ON OUTPUT). FOR

2569000 LINE PRINTER FILES, N = "1" IS NORMAL, N = "0" CAUSES OVERPRINTING,

2570000 N = "2" CAUSES DOUBLE SPACING, ETC. FOR OTHER FILES, A SKIP TO

2571000 THE NEXT RECORD IS PART OF THE I/O OPERATION, AND N = "0" IS THE

2572000 NORMAL SETTING. SPACE(F,N) WILL FAIL IF \$F IS NOT AN I/O STRING

2573000 FOR A FILE, OR IF N IS NON-NUMERIC. FOR EXAMPLE,

2574000

2575000 SPACE("PRINT","2")

2576000

2577000 WILL CAUSE DOUBLE-SPACING ON THE LINE PRINTER FILE PRINT UNTIL

2578000 ANOTHER CALL OF SPACE() CHANGES THE SPACING.

2579000

2580000

2581000

2582000 %CP 6

2583000 SUSPEND(N) HAS TWO USES. IF N IS NON-NULL, IT MUST BE AN INTEGER;

2584000 THE PROGRAM WILL BE SUSPENDED FOR N SECONDS AND THEN WILL RESUME

2585000 EXECUTION FROM THE FUNCTION CALL, WHICH WILL RETURN A NULL STRING.

2586000 IF N ≤ 0, THE CALL IS A NO-OP. IF N IS NULL (OR MISSING), THE

2587000 PROGRAM IS SUSPENDED AND THE INTERACTIVE DEBUGGING ROUTINE IS

2588000 ACTIVATED. SEE SECTION 11.2 FOR DETAILS.

2589000

2590000

2591000

2592000 %CP 50

2593000 STATUS(A,I) GIVES THE STATUS OF ATTACHED TELETYPES. A IS EITHER

2594000 THE ADDRESS OF AN ATTACHED TELETYPE, OR IS NULL, WHICH MEANS

2595000 THAT ALL ATTACHED TELETYPES ARE TO BE CONSIDERED. I IS A NUMBER,

2596000 EITHER "0" OR "1", DEPENDING ON THE TYPE OF STATUS STRING DESIRED.

2597000 THE MEANINGS OF THE TWO VALUES OF I ARE:

2598000

2599000 I = "0" IS A "PASSIVE" STATUS. THAT IS, THE STATUS AS OF THE LAST

2600000 HARDWARE INTERRUPT ASSOCIATED WITH THE TELETYPE IS RETURNED.

2601000 INTERRUPTS OCCUR FOR I/O OPERATIONS AND FOR STATUS INTERROGATES.

2602000 I = "1" IS AN "ACTIVE" STATUS. THIS MEANS THAT A HARDWARE STATUS

2603000 INTERROGATE IS PERFORMED, THE MCP STATUS WORD IS UPDATED, AND

2604000 THE NEW VALUE IS THE VALUE RETURNED. UNFORTUNATELY, THE MCP

2605000 ALSO INTERPRETS AN ACTIVE INTERROGATE AS A COMMAND TO START

2606000 CHARGING THE PROJECT NUMBER OF THE PERSON AT THAT TELETYPE

2607000 FOR THE JOB TIME. FOR SINGLE-USER JOBS, THIS HAS NO EFFECT,

2608000 BUT FOR MULTI-USER JOBS IT CAN BE RATHER ANNOYING.

2609000

2610000 THE FORM OF A STATUS STRING FOR A SINGLE TERMINAL IS:

2611000

2612000 <STATUS> = ( <PROJ#> / <USER#> : <ADDR> : <CONDITIONS> )

2613000

2614000 WHERE <PROJ#>/<USER#> IS THE USERS PROJECT AND I.D. NUMBER.

2615000 (THIS IS THE SECURITY SYSTEM IN USE AT THE UNIVERSITY OF WISCONSIN;

2616000 USERS AT OTHER INSTALLATIONS WILL PROBABLY HAVE TO ALTER THIS

2617000 INTRINSIC TO HANDLE THEIR LOCAL ACCOUNTING PROCEDURES.)  
 2618000 <ADDR> IS THE ADDRESS OF THE TELETYPE, AND <CONDITIONS> IS  
 2619000 A STRING OF "0"-S AND "1"-S WITH THE FOLLOWING MEANINGS:  
 2620000  
 2621000 1ST = "1" IF UNIT IS READY.  
 2622000 2ND = "1" IF UNIT IS READ-READY (I.E., THERE IS INPUT TO READ).  
 2623000 3RD = "1" IF UNIT IS WRITE-READY (OUTPUT IS POSSIBLE)  
 2624000 4TH = "1" IF WRITE IS IN PROGRESS.  
 2625000 5TH = "1" IF BREAK KEY USED ON LAST OUTPUT.  
 2626000 6TH = "1" IF HARDWARE DETECTS ABNORMAL CONDITION.  
 2627000 7TH = "1" IF UNIT IS BUSY.  
 2628000 8TH = "1" IF SOME PROGRAM HAS EXCLUSIVE USE OF UNIT.  
 2629000  
 2630000 MORE CONDITIONS MAY BE ADDED IN THE FUTURE, AS THEY ARE MADE  
 2631000 AVAILABLE BY THE HARDWARE AND BY THE MCP.  
 2632000 WHEN A IS NULL, THE STATUS STRINGS OF ALL ATTACHED TELETYPES  
 2633000 IS RETURNED IN THE FORM:  
 2634000  
 2635000 <STATUS> <STATUS> <STATUS> ... <STATUS>  
 2636000  
 2637000 WITH ONE <STATUS> FOR EACH TELETYPE. IF THERE ARE NO TELETYPES  
 2638000 ATTACHED, OR IF A IS NON-NULL AND NOT THE ADDRESS OF AN ATTACHED  
 2639000 TELETYPE, STATUS() WILL FAIL. FAILURE ALSO OCCURS IF I ISNT  
 2640000 NUMERICALLY "0" OR "1" (NULL OR MISSING IS "0").  
 2641000  
 2642000  
 2643000  
 2644000 %CP 20  
 2645000 TIME(N) RETURNS VARIOUS TIMES ASSOCIATED WITH THE PROGRAM, AS FOLLOWS:  
 2646000  
 2647000 TIME("0") OR TIME() RETURNS THE TIME OF DAY IN THE FORM:  
 2648000 HH:MM:SS  
 2649000  
 2650000 TIME("1") RETURNS THE ELAPSED TIME SINCE MIDNIGHT IN SIXTIETHS  
 2651000 OF A SECOND.  
 2652000  
 2653000 TIME("2") RETURNS THE PROGRAMS TOTAL CPU TIME, IN SIXTIETHS OF  
 2654000 A SECOND.  
 2655000  
 2656000 TIME("3") RETURNS THE PROGRAMS TOTAL I/O CHANNEL TIME, AGAIN IN  
 2657000 SIXTIETHS OF A SECOND.  
 2658000  
 2659000 TIME("4") RETURNS THE VALUE OF A SIX-BIT TIMER IN THE FORM OF  
 2660000 A SINGLE CHARACTER. THE TIMER USED CHANGES EVERY SIXTIETH OF  
 2661000 A SECOND. WHEN CALLED AT WIDELY SPACED INTERVALS (THAT IS,  
 2662000 WITH SEVERAL HUNDRED SNOBOL INSTRUCTIONS IN BETWEEN), IT  
 2663000 BEHAVES MUCH LIKE A RANDOM CHARACTER.  
 2664000  
 2665000  
 2666000  
 2667000 %CP 2  
 2668000 TRACEF(F1,F2,F3,...) TURNS ON TRACING FOR THE FUNCTIONS NAMED BY  
 2669000 F1, F2, F3, ETC. SEE SECTION 11.1 FOR DETAILS.  
 2670000  
 2671000  
 2672000  
 2673000 %CP 2

1312  
 Macroe Business Forums, Inc. sv



2674000 TRACE(L1,L2,L3,...) TURNS ON TRACING FOR THE LABELS CONTAINED IN L1,  
2675000 L2, L3, ETC. SEE SECTION 11.1 FOR DETAILS.  
2676000  
2677000  
2678000  
2679000 %CP 2  
2680000 TRACES(S1,S2,S3,...) TURNS ON TRACING FOR THE STRINGS NAMED BY  
2681000 S1, S2, S3, ETC. SEE SECTION 11.1 FOR DETAILS.  
2682000  
2683000  
2684000  
2685000 %CP 6  
2686000 TRIM(S) RETURNS THE VALUE OF S WITH ALL TRAILING BLANKS (IF ANY)  
2687000 REMOVED. SOME EXAMPLES ARE:  
2688000  
2689000 TRIM(" AB C ") = " AB C"  
2690000 TRIM(" ") = ""  
2691000 TRIM(" XYZ") = " XYZ"  
2692000  
2693000  
2694000  
2695000 %CP 3  
2696000 UNEQL(A,B) SUCCEEDS IF A AND B HAVE DIFFERENT STRINGS AS VALUES.  
2697000 THE USER SHOULD COMPARE UNEQL() WITH THE FUNCTIONS EQUALS() AND  
2698000 .NE().  
2699000  
2700000  
2701000 %CP 7  
2702000 USER(A) HAS TWO USES. IF A IS NULL, THE VALUE IS THE ADDRESS OF  
2703000 THE TELETYPE BEING USED FOR I/O WITH THE STRING TELETYPE. IF  
2704000 A IS NON-NULL, IT MUST BE THE ADDRESS OF AN ATTACHED TELETYPE.  
2705000 THIS TELETYPE BECOMES THE ONE BEING USED FOR I/O, AND ITS ADDRESS  
2706000 IS ALSO THE VALUE RETURNED. NOTE THAT IN THE LATTER CASE, USER(A)  
2707000 IS NOT NECESSARILY THE SAME AS A, SINCE THE ADDRESS IS PUT INTO  
2708000 A STANDARD FORMAT. FOR EXAMPLE, USER("01/03") = "1/3".  
2709000  
2710000  
2711000  
2712000 %CP 6  
2713000 WAIT(N) SETS THE WAITING TIME FOR TELETYPE I/O TO N SECONDS, IF N  
2714000 IS NON-NULL AND NUMERIC. IF N IS NON-NUMERIC, WAIT(N) FAILS.  
2715000 IF N IS NULL, THE WAITING TIME IS NOT ALTERED. IN EITHER CASE,  
2716000 THE RETURN VALUE IS THE WAITING TIME AS IT WAS BEFORE THE FUNCTION  
2717000 WAS CALLED. THIS FUNCTION IS EQUIVALENT TO THE -WAIT CONTROL  
2718000 CARD; IF NEITHER IS USED, 300 SECONDS (5 MINUTES) IS ASSUMED.  
2719000  
2720000  
2721000  
2722000 %SECTION 10  
2723000 %PAGE 1  
2724000 10. INPUT/OUTPUT.  
2725000  
2726000 I/O ON THE B5500 IS NORMALLY ASSOCIATED WITH A FILE, WHICH IS AN  
2727000 ORDERED SET OF I/O "RECORDS". A FILE IS IDENTIFIED BY ITS NAME,  
2728000 WHICH IS OF THE FORM <MULTI-FILE I.D.>/<FILE I.D.>, WHERE BOTH  
2729000 <MULTI-FILE I.D.> AND <FILE I.D.> ARE STRINGS OF FROM 1 TO 7  
2730000 CHARACTERS. NORMALLY, ONLY ALGOL-LIKE IDENTIFIERS ARE USED IN FILE

2731000 NAMES, SINCE MANY MCP FILE-HANDLING COMMANDS CAN ONLY HANDLE SUCH  
2732000 NAMES. A SPECIAL CASE IS MADE FOR CARD-READER FILES, WHICH MAY  
2733000 HAVE ONLY A <FILE I.D.>--THE <MULTI-FILE I.D.> IS THEN ASSUMED  
2734000 TO BE "0000000".  
2735000 TELETYPES ARE HANDLED DIFFERENTLY THAN NON-INTERACTIVE I/O DEVICES.  
2736000 THERE IS NO "FILE" ASSOCIATED WITH A TELETYPE (AS FAR AS SNOBOL  
2737000 USERS ARE CONCERNED). TELETYPE I/O IS DISCUSSED IN SECTION 10.4.  
2738000  
2739000  
2740000  
2741000 %CP 21  
2742000 10.1. I/O STRINGS.  
2743000  
2744000 TO DO I/O, SNOBOL USES "I/O STRINGS" THESE ARE STRING NAMES  
2745000 THAT ARE ASSOCIATED WITH I/O OPERATIONS IN ONE OF TWO WAYS: IF  
2746000 A STRING HAS "INPUT USE", THEN ATTEMPTING TO USE ITS VALUE WILL  
2747000 CAUSE INPUT OF A RECORD FROM ITS ASSOCIATED FILE; IF THE  
2748000 STRING HAS "OUTPUT USE", THEN ASSIGNING A VALUE TO THE STRING WILL  
2749000 CAUSE THE STRING TO BE OUTPUT TO ITS ASSOCIATED FILE. FOR EXAMPLE,  
2750000 THE STRING NAMED "PRINT" IS AN OUTPUT STRING TO THE LINE PRINTER;  
2751000 THE INSTRUCTION:  
2752000  
2753000 PRINT = "A = " QUOTE A QUOTE  
2754000  
2755000 WILL OUTPUT THE VALUE OF A, IDENTIFIED AS SUCH, TO THE PRINTER.  
2756000 BEFORE AN I/O STRING CAN BE USED, IT IS NECESSARY TO ASSIGN IT  
2757000 TO A FILE. THERE ARE TWO WAYS TO DO THIS IN SNOBOL. THE WAY THAT  
2758000 IS RECOMMENDED IS TO USE THE INTRINSIC FUNCTION FILL(), WHICH  
2759000 IS DESCRIBED IN SECTION 9. AN ASSIGNMENT CAN BE MADE BY USING THE  
2760000 MCP "?FILE" CONTROL CARD, THOUGH THIS IS RATHER RESTRICTED. SEE  
2761000 SECTION 2.1 FOR A DESCRIPTION OF THIS CONTROL CARD.  
2762000  
2763000  
2764000  
2765000 %CP 24  
2766000 10.2. OPEN AND CLOSED FILES.  
2767000  
2768000 A FILE IS EITHER "OPEN" OR "CLOSED" WITH RESPECT TO A GIVEN  
2769000 PROGRAM. A FILE IS OPEN IF THE PROGRAM HAS DONE SOME I/O ON IT  
2770000 AND HAS NOT CLOSED IT. IN SNOBOL, THERE ARE TWO WAYS TO CLOSE  
2771000 A FILE: BY CALLING THE INTRINSIC FUNCTION CLOSE(), AND BY TERMINATING  
2772000 THE PROGRAM.  
2773000 WHEN THE FILE ASSOCIATED WITH A GIVEN I/O STRING HAS BEEN CLOSED,  
2774000 THE PROGRAM CAN USE THE FILL() INTRINSIC TO RE-ASSIGN THE I/O STRING  
2775000 TO A DIFFERENT FILE. THE NEW FILE CAN THEN BE OPENED BY DOING I/O  
2776000 ON IT. A PROGRAM CAN CLOSE A FILE, RE-ASSIGN ITS I/O STRING, AND  
2777000 OPEN THE NEW FILE AS OFTEN AS DESIRED. FOR EXAMPLE, IF A PROGRAM  
2778000 HAS BEEN USING THE I/O STRING CARD, WHICH IS ASSIGNED TO, SAY, THE  
2779000 FILE A/B, THIS FILE CAN BE DROPPED AND I/O RE-INITIATED ON THE  
2780000 DISK FILE C/D BY THE FOLLOWING CODE:  
2781000  
2782000 CLOSE("CARD")  
2783000 FILL("CARD","C","D","12")  
2784000 C = CARD  
2785000  
2786000 THIS WILL ASSIGN C THE FIRST RECORD IN THE FILE C/D.  
2787000 SOME I/O STRINGS (AND THEIR ASSOCIATED FILES) ARE OPEN AT THE

2788000 START OF A SNOBOL JOB, WHILE OTHERS ARE CLOSED. SEE THE NEXT SECTION  
 2789000 THE SUCH INFORMATION ON PARTICULAR I/O STRINGS  
 2790000  
 2791000  
 2792000  
 2793000 %CP 10  
 2794000 10.3. STANDARD I/O STRINGS.  
 2795000  
 2796000 THERE ARE A NUMBER OF STANDARD I/O STRINGS PROVIDED BY THE  
 2797000 B5500 SNOBOL SYSTEM, WHICH CAN BE ASSIGNED TO FILES OF THE APPROPRIATE  
 2798000 TYPES. UNFORTUNATELY, THE B5500 OPERATING SYSTEM MAKES IT IMPOSSIBLE  
 2799000 TO PROVIDE THE USER WITH A WAY TO DEFINE HIS OWN FILE TYPES (HOPEFULLY  
 2800000 THIS MAY BE REMEDIED SOMETIME IN THE FUTURE); THE I/O STRINGS THAT  
 2801000 ARE PROVIDED BY SNOBOL SHOULD BE SUFFICIENT FOR THE NEEDS OF MOST  
 2802000 USERS.  
 2803000 THE FOLLOWING I/O STRINGS ARE PRE-DEFINED BY B5500 SNOBOL3:  
 2804000  
 2805000 %CP 8  
 2806000 READ IS INPUT FROM A CARD-IMAGE FILE--THAT IS, ANY FILE (CARD,  
 2807000 TAPE, DISK, ETC.) WITH 80-CHAR LOGICAL RECORDS AND 240-CHAR  
 2808000 PHYSICAL RECORDS. CARD-READER FILES ACTUALLY DO NOT HAVE  
 2809000 240-CHAR PHYSICAL RECORDS, BUT THE SOFTWARE MAKES IT LOOK LIKE  
 2810000 THEY DO. FOR BATCH JOBS, READ IS OPEN AT THE START OF EXECUTION,  
 2811000 AND IS ASSIGNED TO THE FILE THAT CONTAINS THE PROGRAM. THE FIRST  
 2812000 RECORD READ WILL BE THE ONE FOLLOWING THE END CARD. FOR REMOTE  
 2813000 JOBS, READ IS CLOSED AT THE START OF EXECUTION.  
 2814000  
 2815000 %CP 13  
 2816000 LOOK IS A "NON-READING" INPUT STRING FOR THE SAME FILE THAT THE  
 2817000 STRING READ IS ASSIGNED TO. THAT IS, REPEATED USES OF LOOK  
 2818000 WILL GIVE THE SAME RECORD OF THE FILE; USING READ WILL THEN GIVE  
 2819000 THIS SAME RECORD, AND ADVANCE TO THE NEXT ONE. THUS:  
 2820000  
 2821000 A = LOOK  
 2822000 B = LOOK  
 2823000 C = READ  
 2824000 D = READ  
 2825000  
 2826000 WILL ASSIGN THE SAME RECORD TO A, B, AND C; D WILL GET THE NEXT ONE.  
 2827000 THE FUNCTION LOOK() CAN BE USED TO DEFINE STRINGS OF THIS TYPE  
 2828000 FOR OTHER INPUT FILES. THE STRING LOOK CAN IN FACT BE DEFINED  
 2829000 BY LOOK("LOOK","READ").  
 2830000  
 2831000  
 2832000 %CP 5  
 2833000 CARD IS BOTH INPUT AND OUTPUT FOR A CARD-IMAGE FILE WITH 80-CHAR  
 2834000 LOGICAL RECORD AND 240-CHAR PHYSICAL RECORD. OF COURSE, IF IT IS  
 2835000 ASSIGNED TO A CARD-READER FILE, OUTPUT IS IMPOSSIBLE (AND WILL  
 2836000 CAUSE TERMINATION BY THE MCP). CARD IS CLOSED AT THE START OF  
 2837000 EXECUTION.  
 2838000  
 2839000 %CP 28  
 2840000 NEWDISK IS AN I/O STRING FOR A DISK FILE THAT WILL BE CREATED  
 2841000 WHEN THE FILE IS OPENED. THIS FILE CAN BE VIEWED AS A TEMPORARY  
 2842000 DISK FILE, WHICH WILL BE SAVED (PUT IN THE DISK DIRECTORY)  
 2843000 WHEN CLOSE("NEWDISK") OR CLOSE("NEWDISK","LOCK") IS CALLED,  
 2844000 OR WHEN THE PROGRAM TERMINATES. IF ANY OTHER SECOND PARAMETER

2845000 IS USED, THE FILE WILL BE CLOSED AND DISCARDED. THE FILE ASSOC-  
2846000 IATED WITH NEWDISK HAS 80-CHAR LOGICAL RECORDS AND 240-CHAR  
2847000 PHYSICAL RECORDS. WHEN NEWDISK IS CLOSED, IT CAN THEN BE  
2848000 GIVEN A NEW NAME, IF DESIRED, AND RE-OPENED, AND ANOTHER  
2849000 DISK FILE WILL BE CREATED. IF THERE IS ALREADY A FILE IN THE  
2850000 DIRECTORY WITH THE NAME ASSIGNED TO NEWDISK, THEN ATTEMPTING  
2851000 TO ENTER IT IN THE DIRECTORY WILL CAUSE THE PROGRAM TO BE SUSPENDED  
2852000 UNTIL THE OPERATOR TAKES ACTION--USUALLY THIS ACTION CONSISTS OF  
2853000 KILLING THE PROGRAM. THE MAXIMUM NUMBER OF RECORDS THAT CAN  
2854000 BE PUT IN A FILE ASSOCIATED WITH NEWDISK IS 12000. THE END-OF-  
2855000 FILE IS ALWAYS JUST AFTER THE LAST RECORD THAT HAS BEEN WRITTEN  
2856000 ON THE FILE. THE SAVE FACTOR GIVEN TO FILES CREATED BY  
2857000 NEWDISK IS 15 DAYS.

2858000  
2859000

2860000 %CP 9

2861000 PRINT IS AN OUTPUT FILE TO THE LINE PRINTER. IT IS OPEN AT THE  
2862000 START OF EXECUTION, SINCE THIS FILE IS USED BY THE COMPILER  
2863000 TO PRODUCE THE LISTING AND THE OTHER OUTPUT THAT THE COMPILER  
2864000 GENERATES. ON MOST B5500 SYSTEMS, THE LINE PRINTER HAS 132  
2865000 CHARACTERS PER LINE; OUTPUT OF A STRING LONGER THAN THIS WILL  
2866000 CAUSE TRUNCATION UNLESS MODE("OVERFLOW","PRINT") IS CALLED  
2867000 (SEE SECTION 9, THE MODE() FUNCTION). CARRIAGE CONTROL IS  
2868000 HANDLED BY THE FUNCTIONS SPACE() AND PAGE(), WHICH ARE DESCRIBED  
2869000 IN SECTION 9.

2870000

2871000 %CP 12

2872000 SYSPOT IS AN ATTEMPT TO SIMULATE THE FIRST-CHARACTER CARRIAGE CONTROL  
2873000 OF IBM, CDC, ETC. LINE PRINTERS. THE FIRST CHARACTER OF THE  
2874000 OUTPUT IS DELETED AND TRANSLATED INTO THE APPROPRIATE CALLS OF  
2875000 SPACE() AND PAGE(); THE REST OF THE STRING IS THEN PASSED TO  
2876000 THE OUTPUT STRING, PRINT. THE CARRIAGE CONTROL CHARACTERS  
2877000 CURRENTLY RECOGNIZED ARE:

2878000

2879000 " " SINGLE SPACING  
2880000 "0" (ZERO) DOUBLE SPACING  
2881000 "1" PAGE EJECT  
2882000 "\*" NO SKIP AT END OF PAGE  
2883000 "+" NO SPACING--OVERPRINTING

2884000

2885000

2886000 %CP 3

2887000 PUNCH IS OUTPUT TO THE CARD PUNCH. THE LOGICAL AND PHYSICAL  
2888000 RECORD SIZES ARE BOTH 80 CHARACTERS. PUNCH IS CLOSED AT THE  
2889000 START OF EXECUTION.

2890000

2891000 %CP 2

2892000 TELETYPE IS AN I/O STRING TO ALL ATTACHED TELETYPES. IT IS  
2893000 DESCRIBED IN THE NEXT SECTION.

2894000

2895000

2896000

2897000 %CP 12

2898000 10.4. TELETYPES.

2899000

2900000 TELETYPES ARE HANDLED SOMEWHAT DIFFERENTLY FROM THE NON-INTERACTIVE  
2901000 I/O DEVICES. THERE IS NO "FILE" OF PRE-DEFINED "RECORDS" WHEN

2902000 USING A TELETYPE. ALSO, TELETYPE I/O OPERATIONS, ESPECIALLY INPUT,  
2903000 CANT ALWAYS BE DONE IMMEDIATELY; IT IS OFTEN NECESSARY FOR THE  
2904000 PROGRAM TO WAIT, ESPECIALLY FOR INPUT. THE INTRINSIC FUNCTIONS  
2905000 THAT ARE USED WITH FILE I/O STRINGS WILL FAIL IF A TELETYPE I/O  
2906000 STRING IS GIVEN TO THEM, AND THERE ARE A FEW FUNCTIONS THAT ARE  
2907000 USED ONLY WITH TELETYPES.

2908000 THIS SECTION (AND ITS SUB-SECTIONS) GIVES A GENERAL PICTURE OF  
2909000 HOW I/O IS DONE ON TELETYPES.

2910000

2911000

2912000

2913000 %CP 28

2914000 10.4.1. TELETYPE INPUT AND OUTPUT.

2915000

2916000 THERE IS A SINGLE I/O STRING USED WITH TELETYPES; IT IS CALLED

2917000 "TELETYPE". WHEN TELETYPE IS USED AS AN INPUT STRING, A STRING

2918000 IS READ FROM A TELETYPE. THE INPUT IS GIVEN TO THE PROGRAM WHEN THE

2919000 USER TYPES THE "END-OF-MESSAGE" CHARACTER, AN ARROW, "<".

2920000 (THE CTRL-Q COMBINATION IS ALSO END-OF-MESSAGE.) IF NO INPUT IS

2921000 RECEIVED WITHIN THE WAITING TIME FOR TELETYPE I/O, THEN TELETYPE

2922000 WILL FAIL.

2923000 WHEN A PROGRAM ASSIGNS A VALUE TO TELETYPE, THE VALUE IS WRITTEN

2924000 ON THE TELETYPE. THE OUTPUT OPERATION CAN FAIL IF, FOR SOME REASON,

2925000 THE OUTPUT CANT BE INITIATED WITHIN THE WAITING TIME, OR IF THE

2926000 USER PUSHES THE BREAK KEY DURING OUTPUT. THE OUTPUT STARTS WHEREVER

2927000 THE WRITE HEAD MAY HAPPEN TO BE, AND IF THE END OF THE LINE IS

2928000 REACHED, OVERPRINTING WILL OCCUR. THE PROGRAM MUST INDICATE THE

2929000 "CARRIAGE CONTROL" BY USING THE FOLLOWING CHARACTERS:

2930000

2931000 "<" (SINGLE QUOTE ON THE TELETYPE) CAUSES A RETURN TO THE

2932000 FIRST CHARACTER POSITION OF THE CURRENT LINE.

2933000 ">" (UP-ARROW ON THE TELETYPE) CAUSES THE PAPER TO ADVANCE

2934000 ONE LINE ("LINE FEED").

2935000

2936000 IT SHOULD BE NOTED THAT ASSIGNING A VALUE TO TELETYPE ONLY

2937000 INITIATES OUTPUT. AS SOON AS THE OUTPUT STARTS, THE OUTPUT

2938000 ROUTINE RETURNS TO THE PROGRAM AND EXECUTION CONTINUES. SINCE

2939000 A TELETYPE IS SUCH A SLOW OUTPUT DEVICE, THE PROGRAM CAN GET QUITE

2940000 A LOT OF COMPUTING DONE BY THE TIME THE OUTPUT IS FINISHED. IT

2941000 IS QUITE COMMON FOR A PROGRAM TO HAVE A SECOND OUTPUT STRING READY

2942000 BY THE TIME A PREVIOUS OUTPUT IS FINISHED TYPING.

2943000 THERE CAN BE SEVERAL TELETYPES ATTACHED TO A SNOBOL PROGRAM AT

2944000 THE SAME TIME. AT A GIVEN INSTANCE, THE PROGRAM CAN DO I/O WITH

2945000 ONLY ONE OF THEM, BUT THE ONE BEING TALKED TO CAN BE CHANGED BY

2946000 USING THE INTRINSIC FUNCTION USER(). FOR OTHER FUNCTIONS RELEVANT

2947000 TO TELETYPE, SEE THE DESCRIPTIONS OF STATUS(), WAIT(), RELEASE(),

2948000 AND SUSPEND().

2949000

2950000

2951000

2952000 %CP 7

2953000 10.4.2. TELETYPE WAITING TIMES.

2954000

2955000 AN I/O OPERATION ON A TELETYPE WILL FAIL IF IT DOES NOT SUCCEED

2956000 WITHIN A FIXED PERIOD OF TIME CALLED THE "WAITING TIME". THIS TIME

2957000 LIMIT CAN BE SET BY USING THE -WAIT CONTROL CARD OR THE INTRINSIC

2958000 FUNCTION WAIT(). IF NEITHER IS USED, A WAITING TIME OF 300 SECONDS

2959000 (5 MINUTES) IS ASSUMED BY SNOBOL.

2960000

2961000

2962000

2963000 %CP 40

2964000 10.4.3. TELETYPE ADDRESSES.

2965000

2966000 THIS SECTION IS ONLY OF INTEREST TO USERS WISHING TO USE

2967000 SEVERAL TELETYPES AT THE SAME TIME. OTHER USERS SHOULD PROBABLY

2968000 IGNORE IT.

2969000 SINCE IT IS POSSIBLE FOR A PROGRAM TO BE COMMUNICATING WITH

2970000 SEVERAL TELETYPES, AND ALL TELETYPE I/O IS DONE THROUGH THE ONE

2971000 I/O STRING, TELETYPE, THERE MUST BE A WAY TO DISTINGUISH BETWEEN

2972000 THE INDIVIDUAL TELETYPES. THIS IS DONE BY GIVING EACH TELETYPE

2973000 AN ADDRESS. THE FUNCTION USER() CAN THEN BE USED TO SWITCH BACK

2974000 AND FORTH BETWEEN THE VARIOUS TELETYPES ATTACHED.

2975000 A TELETYPE ADDRESS CONSISTS OF A PAIR OF INTEGERS SEPARATED

2976000 BY A "/". THE FIRST INTEGER IS CALLED THE "TERMINAL UNIT" NUMBER;

2977000 THE SECOND IS CALLED THE "BUFFER ADDRESS" (THE USER NEED NOT

2978000 REMEMBER THESE TERMS, SINCE THEY ARE RATHER WORTHLESS FOR MOST

2979000 PURPOSES). SOME POSSIBLE ADDRESSES, DEPENDING ON THE HARDWARE

2980000 AVAILABLE, ARE: "1/5", "1/15", "3/7", "10/13", "2/2", ETC.

2981000 A SNOBOL PROGRAM CAN ONLY COMMUNICATE WITH TELETYPES THAT ARE

2982000 "ATTACHED" TO IT. A TELETYPE BECOMES ATTACHED TO A SNOBOL PROGRAM

2983000 IN ONE OF TWO WAYS:

2984000

2985000 1) THE USER INITIATES THE JOB FROM A TELETYPE, WITH AN MCP

2986000 CONTROL CARD STARTING:

2987000 ??EXECUTE SNOBOL/SNOBOL ...

2988000 2) THE USER ATTACHES HIS TELETYPE TO A RUNNING JOB WITH

2989000 ONE OF THE FOLLOWING EQUIVALENT COMMANDS:

2990000 ??RUN SNOBOL/SNOBOL←

2991000 ??RUN SNOBOL←

2992000

2993000 A PROGRAM CAN FIND OUT WHICH TELETYPES, IF ANY, ARE ATTACHED BY

2994000 USING THE STATUS() FUNCTION. ANY OF THE TELETYPES GIVEN BY THIS

2995000 FUNCTION CAN BE USED AS AN I/O DEVICE BY CALLING USER(A), WHERE

2996000 A IS THE TELETYPE ADDRESS. FOR EXAMPLE, IF IT IS KNOWN THAT A

2997000 TELETYPE WITH THE ADDRESS "1/5" IS ATTACHED, I/O CAN BE DONE ON

2998000 IT IN THE FOLLOWING MANNER:

2999000

3000000 USER("1/5")

3001000 TELETYPE = "<?" MESSAGE "<?"

3002000 ANSWER = TELETYPE /F(NO.RESPONSE)

3003000

3004000 OF COURSE, IF THERE IS ONLY ONE TELETYPE ATTACHED (AS IS ALMOST

3005000 ALWAYS THE CASE), THERE IS NO NEED TO WORRY ABOUT WHICH ONE IT IS,

3006000 SINCE ALL TELETYPE I/O WILL BE WITH THAT ONE TELETYPE.

3007000

3008000

3009000

3010000 %CP 40

3011000 10.4.4. SPECIAL CHARACTERS FOR THE TELETYPE.

3012000

3013000 THE CHARACTERS LISTED BELOW REQUIRE SPECIAL ATTENTION WHEN USED

3014000 WITH TELETYPE I/O. SOME ARE INTERPRETED AS CONTROL CHARACTERS BY

3015000 THE HARDWARE, AND OTHERS ARE PRINTED DIFFERENTLY ON THE LINE PRINTER

3016000 THAN ON THE TELETYPE.

3017000

3018000

3019000

PRINTER TELETYPE

DESCRIPTION

3020000

3021000

"←"

"←"

"GROUP MARK", OR "END-OF-MESSAGE" CHARACTER.

3022000

THIS CAN NOT BE READ OR WRITTEN. THE

3023000

CTRL-Q COMBINATION IS THE SAME CHARACTER,

3024000

BUT DOESNT PRINT.

3025000

3026000

"≤"

APDSTROPHE

THIS CAN BE INPUT, BUT WHEN OUTPUT, IT

3027000

CAUSES A "CARRIAGE RETURN"--THE WRITE

3028000

HEAD RETURNS TO COLUMN 1, AND NO CHAR

3029000

IS PRINTED.

3030000

3031000

"↕"

UP-ARROW

THIS CAN ALSO BE INPUT, BUT WHEN OUTPUT,

3032000

IT CAUSES THE PAPER TO ADVANCE BY ONE LINE.

3033000

3034000

"≥"

EXCLAMATION

WHEN TYPED, THIS CAUSES ERASURE OF THE

3035000

THE CURRENT "BUFFER"; WHEN OUTPUT, IT

3036000

CAUSES A DISCONNECT.

3037000

"<"

"<"

IS IGNORED ON OUTPUT; WHEN TYPED, IT

3038000

SOMETIMES CAUSES ERASURE OF THE PREVIOUS

3039000

CHARACTER TYPED, AND SOMETIMES ACTS

3040000

LIKE A "+". THE RULES THAT SAY WHICH

3041000

WILL OCCUR ARE TOO COMPLICATED TO BE

3042000

OF ANY PRACTICAL USE.

3043000

3044000

"x"

BACK-SLASH

NORMAL EXCEPT FOR THE DIFFERENT CHARACTER

3045000

USED ON THE TELETYPE. MOST TELETYPES

3046000

ALSO DONT INDICATE THAT THIS CHARACTER

3047000

IS TYPED BY USING SHIFT-L.

3048000

3049000

NOTE THAT "[" AND "]" ARE ON THE TELETYPE (SHIFT-J AND SHIFT-M);

3050000

THEY JUST ARENT INDICATED ON THE KEYS OF MOST TELETYPES.

3051000

3052000

3053000

3054000 %CP 50

3055000

10.5. CAUSES OF I/O.

3056000

3057000

TO EFFECTIVELY USE SNOBOL I/O, IT IS NECESSARY TO KNOW WHAT

3058000

CONSTRUCTS CAUSE INPUT OR OUTPUT TO OCCUR. THESE ARE DESCRIBED

3059000

HERE IN DETAIL:

3060000

3061000

INPUT OCCURS WHEN:

3062000

1) AN INPUT STRING IS USED AS A STRING REFERENCE IN RULES

3063000

OF THE FORMS:

3064000

3065000

<STR REF>

3066000

<STR REF> <PATTERN>

3067000

<STR REF> <PATTERN> <REPLACEMENT>

3068000

3069000

INPUT OCCURS IMMEDIATELY AFTER THE STRING REFERENCE IS

3070000

EVALUATED. THUS IN THE RULE:

3071000

3072000

CARD F(B) /F(L3)

3073000  
3074000 THE INPUT OCCURS BEFORE F(B) IS CALLED.  
3075000 2) AN EXPRESSION USES THE VALUE OF AN INPUT STRING. INPUT  
3076000 OCCURS WHEN THE NAME IS ENCOUNTERED. THUS IN THE RULE:  
3077000  
3078000 STR F(READ "\*",AL("1"))  
3079000  
3080000 THE INPUT OCCURS BEFORE AL("1") IS CALLED.  
3081000  
3082000 OUTPUT OCCURS WHEN:  
3083000 1) AN OUTPUT STRING IS USED AS A STRING REFERENCE IN RULES  
3084000 OF THE FORMS:  
3085000  
3086000 <STR REF> <REPLACEMENT>  
3087000 <STR REF> <PATTERN> <REPLACEMENT>  
3088000  
3089000 OUTPUT OCCURS IMMEDIATELY AFTER THE REPLACEMENT IS DONE,  
3090000 FOR EXAMPLE, TO OUTPUT A LINE TO THE PRINTER:  
3091000  
3092000 PRINT = "CRD = " QUOTE CRD QUOTE  
3093000  
3094000 NOTE THAT IN RULES OF THE FORM <STR REF> <PATTERN> <REPLACEMENT>  
3095000 BOTH INPUT AND OUTPUT WILL OCCUR IF THE STRING REFERENCE IS  
3096000 BOTH AN INPUT STRING AND AN OUTPUT STRING.  
3097000 2) A STRING VARIABLE HAS AN OUTPUT STRING FOR ITS NAME, AND  
3098000 THE PATTERN MATCH SUCCEEDS. OUTPUT OCCURS WHEN THE ASSIGNMENT  
3099000 TO THAT VARIABLE IS MADE.  
3100000 3) A FUNCTION IS CALLED, AND AN OUTPUT STRING IS EITHER THE  
3101000 STRING NAMED THE SAME AS THE FUNCTION, A FORMAL PARAMETER,  
3102000 OR A LOCAL VARIABLE. OUTPUT OCCURS AT BOTH THE CALL AND  
3103000 THE RETURN OF THE FUNCTION.  
3104000  
3105000  
3106000  
3107000 %SECTION 11  
3108000 %PAGE 1  
3109000 11. DEBUGGING AIDS.  
3110000  
3111000 THERE ARE CURRENTLY TWO TYPES OF DEBUGGING AIDS PROVIDED IN  
3112000 B5500 SNOBOL3. THERE IS A TRACING FACILITY THAT CAN BE USED TO  
3113000 MAKE A RECORD OF VALUES ASSIGNED TO VARIABLES, CALLS OF FUNCTIONS,  
3114000 AND TRANSFERS TO LABELS. THERE IS ALSO AN INTERACTIVE DEBUGGING  
3115000 TOOL THAT ALLOWS A USER AT A TELETYPE TO SUSPEND A PROGRAM, ASK  
3116000 QUESTIONS ABOUT ITS STATE, CORRECT ERRORS MADE BY THE PROGRAM,  
3117000 AND RESUME EXECUTION.  
3118000  
3119000  
3120000  
3121000 %CP 5  
3122000 11.1. TRACING.  
3123000  
3124000 TRACING IS PROVIDED IN B5500 SNOBOL FOR STRINGS, FUNCTIONS, AND  
3125000 LABELS. OUTPUT FROM TRACING GOES TO THE LINE PRINTER FILE, PRINT.  
3126000 THE FUNCTIONS THAT CONTROL TRACING ARE:  
3127000  
3128000  
3129000 %CP 21



3130000 TRACES(S1,S2,S3,...) TURNS ON TRACING FOR THE STRINGS NAMED BY  
3131000 S1, S2, S3, ETC. THIS MEANS THAT WHENEVER A VALUE IS ASSIGNED  
3132000 TO ONE OF THESE STRINGS, OUTPUT OCCURS GIVING THE STATEMENT  
3133000 NUMBER, THE STRING NAME, AND ITS VALUE, IN THE FORM:  
3134000  
3135000 \*S <STMT#> <NAME> = " <VALUE> "  
3136000  
3137000 FOR EXAMPLE, TO TURN ON TRACING FOR ALPHA, XV2, AND QVAR, CALL:  
3138000  
3139000 TRACES("ALPHA","XV2","QVAR")  
3140000  
3141000 IF THE VARIABLE ALPHA WERE ASSIGNED THE VALUE "THIS ONE" IN  
3142000 STATEMENT 135, THE FOLLOWING WOULD BE PRINTED:  
3143000  
3144000 \*S 135 ALPHA = "THIS ONE"  
3145000  
3146000 STRING TRACING IS CONSIDERED AN OUTPUT USE OF THE STRING, AND  
3147000 TURNING ON TRACING WILL TERMINATE ANY PREVIOUS I/O USE THE STRING  
3148000 MAY HAVE HAD. TRACING OUTPUT OCCURS AT ALL PLACES WHERE OUTPUT  
3149000 OCCURS FOR STRINGS WITH OUTPUT USE--THAT IS, WHEREVER A VALUE  
3150000 IS ASSIGNED.  
3151000  
3152000  
3153000 %CP 27  
3154000 TRACEF(F1,F2,F3,...) TURNS ON TRACING FOR THE FUNCTION NAMED BY  
3155000 F1, F2, F3, ETC. WHEN ANY OF THESE FUNCTIONS IS CALLED, OUTPUT  
3156000 OCCURS GIVING THE FUNCTION NAME, THE STATEMENT NUMBER, AND THE  
3157000 VALUES OF THE PARAMETERS, IN THE FORM:  
3158000  
3159000 \*C <STMT#> <FCT NAME>( <PARAMS> )  
3160000  
3161000 WHEN THE FUNCTION RETURNS, MORE OUTPUT OCCURS (STARTING WITH  
3162000 "\*R" RATHER THAN "\*C") GIVING THE RETURN VALUE OR INDICATING  
3163000 FAILURE. FOR EXAMPLE, SUPPOSE A PROGRAM HAD CALLED:  
3164000  
3165000 TRACEF("BLOCK","TIME","FLXGQ")  
3166000  
3167000 THIS WILL START TRACING FOR THE FUNCTIONS BLOCK(), TIME(), AND  
3168000 FLXGQ(). SUPPOSE THAT STATEMENT 93 THEN CALLS BLOCK(A,"3"), AND  
3169000 A = "ITO"; AND THE VALUE RETURNED IS "X.3". THEN THE FOLLOWING  
3170000 OUTPUT WILL OCCUR:  
3171000  
3172000 \*C 93 BLOCK("ITO","3")  
3173000 \*R 93 BLOCK() = "X.3"  
3174000  
3175000 IF BLOCK HAD FAILED, THE LAST LINE WOULD HAVE BEEN:  
3176000  
3177000 \*R 93 BLOCK() FAILED.  
3178000  
3179000 NOTE THAT TRACING CAN BE USED WITH BOTH SYSTEM-DEFINED AND  
3180000 PROGRAM-DEFINED FUNCTIONS.  
3181000  
3182000  
3183000 %CP 13  
3184000 TRACEL(L1,L2,L3,...) INITIATES TRACING FOR THE LABELS CONTAINED  
3185000 IN L1, L2, L3, ETC. THIS MEANS THAT WHENEVER ANY OF THESE LABELS  
3186000 IS ENCOUNTERED, OUTPUT OCCURS GIVING THE LABEL, THE NUMBER OF

3187000 TIMES IT HAS BEEN EXECUTED (INCLUDING THE CURRENT ONE), AND  
 3188000 THE NUMBER OF THE PREVIOUS INSTRUCTION. THUS, IF A PROGRAM  
 3189000 HAD CALLED:  
 3190000  
 3191000 TRACE("BACK","NONE","B1","B2")  
 3192000  
 3193000 THEN IF STATEMENT 216 TRANSFERS TO BACK, WHICH HAS PREVIOUSLY  
 3194000 BEEN EXECUTED 17 TIMES, THE FOLLOWING WILL BE PRINTED:  
 3195000  
 3196000 \*L BACK 18 FROM 216  
 3197000  
 3198000  
 3199000 %CP 17  
 3200000 DETACH(N,T) CAN BE USED TO TURN OFF TRACING. THE NAME OF THE  
 3201000 OBJECT CONCERNED IS CONTAINED IN N; T GIVES THE TYPE AS FOLLOWS:  
 3202000  
 3203000 "S" MEANS THAT N CONTAINS A STRING NAME,  
 3204000 "F" MEANS THAT N CONTAINS A FUNCTION NAME.  
 3205000 "L" MEANS THAT N CONTAINS A LABEL.  
 3206000  
 3207000 ONLY THE FIRST CHARACTERS OF T IS LOOKED AT; IF T IS NULL, "S"  
 3208000 IS ASSUMED. THUS, TO TURN OFF TRACING FOR THE STRING NAMED ALPHA,  
 3209000 THE PROGRAM COULD CALL ANY OF:  
 3210000  
 3211000 DETACH("ALPHA")  
 3212000 DETACH("ALPHA","S")  
 3213000 DETACH("ALPHA","STRING")  
 3214000 DETACH("ALPHA","SLORKLE")  
 3215000  
 3216000  
 3217000  
 3218000 %PAGE  
 3219000 11.2. INTERACTIVE DEBUGGING.  
 3220000  
 3221000 AN INTERACTIVE DEBUGGING AID HAS BEEN IMPLEMENTED FOR USE WITH  
 3222000 A TELETYPE. BASICALLY, IT CONSISTS OF SUSPENDING A PROGRAM AND  
 3223000 ASKING THE INTERPRETER QUESTIONS ABOUT ITS STATE. IF ERRORS HAVE  
 3224000 BEEN MADE, VALUES OF STRINGS CAN BE CHANGED, AND THE PROGRAM CAN  
 3225000 BE RE-INITIATED EITHER WHERE IT WAS SUSPENDED OR AT ANY OTHER  
 3226000 LABELED INSTRUCTION.  
 3227000 A PROGRAM CAN BE SUSPENDED, AND THE DEBUGGING ROUTINE INITIATED,  
 3228000 BY ANY OF THE FOLLOWING METHODS:  
 3229000  
 3230000 1) THE PROGRAM CAN CALL SUSPEND()--WITH NULL PARAMETER.  
 3231000 2) IF A -DEBUG CARD OCCURRED DURING COMPILATION, SUSPENSION  
 3232000 WILL OCCUR JUST BEFORE THE FIRST INSTRUCTION STARTS.  
 3233000 3) SUSPENSION OCCURS WHEN THE PROGRAM IS RUNNING IN "DEBUG"  
 3234000 MODE AND THE PROGRAM ENCOUNTERS A LIMIT SET DURING A PREVIOUS  
 3235000 SUSPENSION.  
 3236000 4) THE PROGRAM COMMITS ANY OF THE VARIOUS PROGRAM ERRORS THAT  
 3237000 ARE NORMALLY FATAL. THESE ARE ERRORS SUCH AS CALLING UNDEFINED  
 3238000 FUNCTIONS, TRYING TO EXECUTE A STATEMENT WITH A SYNTAX ERROR,  
 3239000 TRANSFERRING TO AN UNDEFINED LABEL, ETC.  
 3240000  
 3241000 A PROGRAM RUNS EITHER IN NORMAL ("RUN") MODE OR IN "DEBUG" MODE.  
 3242000 "DEBUG" MODE CAUSES CHECKING FOR A NUMBER OF PROGRAM CONDITIONS  
 3243000 WHOSE OCCURENCE WILL CAUSE THE PROGRAM TO BE SUSPENDED. FOR A

3244000 LIST OF THE LIMITS THAT CAN BE SET BY THE PROGRAMMER, SEE THE "RUN"  
 3245000 COMMAND BELOW. A PROGRAM CAN BE IN DEBUG MODE FOR ANY OF THE  
 3246000 FOLLOWING REASONS:  
 3247000  
 3248000 1) THE PROGRAM CONTAINED A -DEBUG CONTROL CARD.  
 3249000 2) THE PROGRAM WAS SUSPENDED EARLIER.  
 3250000 3) THE PROGRAM HAS CALLED MODE("DEBUG").  
 3251000  
 3252000 DEBUG MODE IS TURNED OFF BY CALLING MODE("RUN"). IF THIS IS DONE,  
 3253000 THE ONLY WAY TO GET BACK TO THE DEBUGGING ROUTINE IS FOR THE PROGRAM  
 3254000 TO CALL SUSPEND() OR MODE("DEBUG"), SINCE MODE("RUN") TURNS OFF ALL  
 3255000 THE LIMIT CHECKING ASSOCIATED WITH DEBUG MODE.  
 3256000 WHEN A PROGRAM IS SUSPENDED, THE SYSTEM TYPES OUT "SUSPENDED"  
 3257000 AND IS THEN WAITING FOR A COMMAND. THE COMMANDS THAT HAVE BEEN  
 3258000 IMPLEMENTED AS OF THIS WRITING ARE:  
 3259000  
 3260000  
 3261000 %CP 23  
 3262000 TYPE <NAME>←  
 3263000 THIS CAUSES THE VALUE OF THE VARIABLE <NAME> TO BE TYPED IN  
 3264000 THE FOLLOWING FORM:  
 3265000  
 3266000 <NAME> = " <VALUE> "  
 3267000  
 3268000 IF THE PROGRAM DOESNT CONTAIN <NAME>, AND IT HASNT BEEN CREATED  
 3269000 BY INDIRECTION, THE RESPONSE IS:  
 3270000  
 3271000 <NAME> NOT IN USE  
 3272000  
 3273000 IN ORDER TO ALLOW THE USER TO USE ANY STRING NAME THAT CAN  
 3274000 BE ENTERED FROM A TELETYPE, THE FOLLOWING CONVENTION IS MADE:  
 3275000 THERE IS EXACTLY ONE BLANK AFTER "TYPE"; THE REST OF THE INPUT  
 3276000 STRING IS THE <NAME>. THUS ANY CHARACTER READABLE FROM A  
 3277000 TELETYPE CAN BE USED IN <NAME>, INCLUDING BLANKS. FOR EXAMPLE,  
 3278000 THE COMMAND:  
 3279000  
 3280000 TYPE \*AA/3 B1←  
 3281000  
 3282000 MIGHT GET THE RESPONSE:  
 3283000  
 3284000 \*AA/3 B1 = "437 A+B 4"  
 3285000  
 3286000  
 3287000  
 3288000 %CP 19  
 3289000 SET <NAME>←  
 3290000 <VALUE>←  
 3291000 THIS COMMAND, REQUIRING TWO INPUT STRINGS, ASSIGNS THE STRING  
 3292000 <VALUE> TO <NAME>. THE FIRST INPUT, SET <NAME>←, PRODUCES  
 3293000 A RETURN/LINE FEED FROM SNOBOL, AND IT IS THEN WAITING FOR THE  
 3294000 <VALUE>. THE CONVENTIONS FOR <NAME> ARE THE SAME AS FOR THE  
 3295000 TYPE COMMAND. FOR EXAMPLE, THE THREE COMMANDS:  
 3296000  
 3297000 SET AB3←  
 3298000 N.47.-27.C1←  
 3299000 TYPE AB3←  
 3300000

3301000 WILL GET THE RESPONSE:  
3302000  
3303000 AB3 = "N.47.-27.C1"  
3304000  
3305000 BECAUSE TRANSMISSION ERRORS DO OCCUR FAIRLY OFTEN WITH TELETYPES,  
3306000 IT IS ADVISABLE TO FOLLOW SET COMMANDS WITH TYPE COMMANDS TO  
3307000 VERIFY THAT THE CORRECT VALUE HAS BEEN ASSIGNED.  
3308000  
3309000  
3310000  
3311000 %CP 9  
3312000 LABEL <LABEL>←  
3313000 THIS COMMAND SIMPLY TELLS THE USER HOW MANY TIMES THE LABEL HAS  
3314000 BEEN EXECUTED SO FAR. IF THE LABEL ISNT DEFINED, THE RESPONSE  
3315000 WILL BE:  
3316000  
3317000 <LABEL> NOT DEFINED  
3318000  
3319000 LIKE WITH THE TYPE AND SET COMMANDS, THERE IS EXACTLY ONE BLANK  
3320000 BETWEEN "LABEL" AND <LABEL>.  
3321000  
3322000  
3323000  
3324000 %CP 41  
3325000 RUN <INTEGER> FROM <LABEL> TO <LABEL> <LABEL> <LABEL> ...←  
3326000 THIS COMPLICATED COMMAND RE-INITIATES EXECUTION, AND ALSO SETS  
3327000 UP THE CONDITIONS THAT ARE TO CAUSE SUSPENSION LATER. ALL OF  
3328000 THE COMMAND AFTER "RUN" IS OPTIONAL, AND BREAKS DOWN INTO SEVERAL  
3329000 SECTIONS, THAT ARE REALLY PARAMETERS.  
3330000 IF <INTEGER> IS USED, THEN THE PROGRAM WILL BE SUSPENDED  
3331000 AFTER IT EXECUTES <INTEGER> INSTRUCTIONS, IF IT MANAGES TO  
3332000 GET THAT FAR AT ALL.  
3333000 IF FROM <LABEL> IS USED, THE CURRENT INSTRUCTION WILL BE  
3334000 TERMINATED (AS IF IT HAD FAILED), AND EXECUTION WILL RESUME AT  
3335000 THE STATEMENT LABELED <LABEL>. IF <LABEL> ISNT DEFINED, THE  
3336000 RESPONSE WILL BE:  
3337000  
3338000 <LABEL> NOT DEFINED  
3339000  
3340000 AND THE PROGRAM WILL REMAIN SUSPENDED.  
3341000 IF THE TO <LABEL> <LABEL> ... PART IS INCLUDED, IT IS  
3342000 INTERPRETED AS A LIST OF LABELS THAT ARE TO CAUSE SUSPENSION IF  
3343000 THE PROGRAM TRIES TO EXECUTE THEM. NOTE THAT THE LABELS ARE  
3344000 SEPARATED BY BLANKS--ANY OTHER DELIMITERS (SUCH AS COMMAS) WILL  
3345000 BE INTERPRETED AS PART OF A LABEL. IF A LABEL IS USED THAT  
3346000 ISNT DEFINED, THE "NOT DEFINED" RESPONSE WILL BE GIVEN, AND  
3347000 THE PROGRAM WILL REMAIN SUSPENDED. THERE IS A LIMIT TO THE  
3348000 NUMBER OF LABELS THAT CAN BE GIVEN. THIS LIMIT IS CURRENTLY  
3349000 SET TO 16, THOUGH IT MAY BE INCREASED IN THE FUTURE IF THERE  
3350000 IS DEMAND FOR MORE. IF TOO MANY LABELS ARE GIVEN, THE RESPONSE  
3351000 WILL BE "TOO MANY LABELS", AND THE PROGRAM WILL REMAIN SUSPENDED.  
3352000 AS A SPECIAL CASE, IF "TO" IS USED WITHOUT ANY LABELS FOLLOWING  
3353000 IT, THE LIMITS WILL BE THE SAME AS THOSE USED THE LAST TIME (IF  
3354000 ANY).  
3355000 SOME EXAMPLES OF RUN COMMANDS ARE:  
3356000  
3357000 RUN←

3358000 RUN 75←  
 3359000 RUN FROM HNT3 TO HNT5 HNT8←  
 3360000 RUN 40 FROM GIN.7←  
 3361000 RUN 500 FROM A3 TO B1 AA1 APPLY3 ERROR EOJ ALTRY←  
 3362000 RUN FROM FORR1 TO←  
 3363000 RUN 1000 TO←  
 3364000 RUN 1←  
 3365000 RUN FROM START←  
 3366000  
 3367000  
 3368000  
 3369000 %CP 2  
 3370000 END←  
 3371000 THIS CAUSES THE PROGRAM TO TERMINATE NORMALLY.  
 3372000  
 3373000  
 3374000  
 3375000 %CP 2  
 3376000 ABORT←  
 3377000 THIS CAUSES ABNORMAL TERMINATION OF THE PROGRAM.  
 3378000  
 3379000  
 3380000  
 3381000 %CP 7  
 3382000 WHY←  
 3383000 THIS CAUSES THE REASON(S) FOR SUSPENSION TO BE TYPED. SOME  
 3384000 POSSIBLE RESPONSES TO THIS ARE:  
 3385000  
 3386000 RULE LIMIT HIT  
 3387000 LABEL LIMIT HIT  
 3388000 SUSPEND() CALLED  
 3389000 FATAL ERROR  
 3390000  
 3391000  
 3392000  
 3393000 %CP 4  
 3394000 WHERE←  
 3395000 THIS CAUSES THE SYSTEM TO TYPE OUT WHERE THE PROGRAM WAS SUSPENDED  
 3396000 (STATEMENT NUMBER AND LAST LABEL) AND HOW MANY RULES HAVE BEEN  
 3397000 EXECUTED SO FAR.  
 3398000  
 3399000  
 3400000  
 3401000 %SECTION 12  
 3402000 %PAGE 1  
 3403000 12. MISCELLANEOUS.  
 3404000  
 3405000 %CP 6  
 3406000 THE FOLLOWING CHARACTERS HAVE SPECIAL MEANING IN COLUMN 1:  
 3407000 "?" MCP CONTROL CARD  
 3408000 "- " SNOBOL CONTROL CARD  
 3409000 ". " CONTINUATION CARD (INVALID FROM TELETYPE)  
 3410000 "\* " COMMENT CARD  
 3411000 "% " COMMENT CARD  
 3412000  
 3413000  
 3414000

3415000 %CP 3  
3416000 A LABEL IS ANY NON-NULL STRING PRECEDING THE FIRST BLANK ON A CARD,  
3417000 THE FIRST CHARACTER CAN BE ANY EXCEPT THOSE LISTED ABOVE; THE REST  
3418000 OF THE LABEL CAN USE ANY CHARACTERS BUT BLANKS.  
3419000  
3420000  
3421000  
3422000 %CP 4  
3423000 WHEN ENTERING A PROGRAM FROM A TELETYPE, ANY INPUT ENDING WITH  
3424000 "#" WILL BE TREATED AS AN ERROR--THE COMPILER WILL DISCARD IT  
3425000 TYPE "TRY AGAIN", AND ASK FOR MORE INPUT.  
3426000  
3427000  
3428000  
3429000 %CP 5  
3430000 THE GO-TO PART CAN BE SET OFF BY EITHER "/" OR ":", IF "/" IS  
3431000 USED (NOTE THE BLANK), THE NEXT CHARACTER MUST BE EITHER "(", "S",  
3432000 OR "F". A COLON (":") CAN BE SURROUNDED BY ANY NUMBER OF BLANKS,  
3433000 INCLUDING NONE, AND CAN STAND ALONE WITH GO GO-TO PART FOLLOWING  
3434000 IT, IN WHICH CASE IT IS IGNORED.  
3435000  
3436000  
3437000  
3438000 %CP 2  
3439000 TO SEPARATE A PATTERN OR STRING REFERENCE FROM A REPLACEMENT, EITHER  
3440000 "=" OR "<" CAN BE USED. THE COMPILER THINKS THEY ARE EQUIVALENT.  
3441000  
3442000  
3443000  
3444000 %CP 7  
3445000 THE "ILLEGAL CHARACTER", "?", CANT BE READ FROM A CARD READER EXCEPT  
3446000 IN COLUMN 1, IN WHICH CASE IT INDICATES AN MCP CONTROL CARD. ANY  
3447000 CARD CONTAINING "?" ANYWHERE ELSE WILL BE THROWN OUT BY THE CARD  
3448000 READER, AND THE JOB WILL USUALLY NOT BE RUN. THIS CHARACTER CANT  
3449000 BE INPUT FROM A TELETYPE, EITHER, SINCE ANY INPUT CONTAINING "?"  
3450000 WILL BE INTERPRETED AS A MESSAGE TO THE MCP. THERE IS NO PROBLEM  
3451000 WITH WRITING A "?" ON THE LINE PRINTER OR ON A TELETYPE.  
3452000  
3453000  
3454000  
3455000 %CP 9  
3456000 THERE ARE THREE STRINGS WHICH B5500 SNOBOL3 INITIALIZES TO HAVE  
3457000 NON-NULL VALUES. THEY ARE:  
3458000  
3459000 QUOTE CONTAINS A QUOTE  
3460000 QMARK = "?"  
3461000 ARROW = "<"  
3462000  
3463000 THE STRINGS QMARK AND ARROW ARE PROVIDED MAINLY FOR TELETYPE USERS;  
3464000 SINCE NEITHER CAN BE INPUT FROM A TELETYPE.  
3465000  
3466000  
3467000  
3468000 %CP 2  
3469000 THE SYSTEM LIMIT ON STRING SIZE IS 8181 CHARACTERS. ATTEMPTING TO  
3470000 CREATE A STRING LONGER THAN THIS WILL CAUSE ABNORMAL TERMINATION.  
3471000

3472000  
 3473000  
 3474000 %CP 9  
 3475000 THE CONCATENATE OPERATOR IS A BLANK. BLANKS ARE USUALLY OPTIONAL,  
 3476000 WITH THE FOLLOWING EXCEPTIONS:  
 3477000  
 3478000 1) BEFORE THE STRING REFERENCE.  
 3479000 2) BETWEEN TWO IDENTIFIERS.  
 3480000 3) BETWEEN AN IDENTIFIER AND A FOLLOWING "(", TO DISTINGUISH  
 3481000 A CONCATENATION OF A STRING AND A GROUPING FROM A FUNCTION  
 3482000 CALL (WHICH HAS NO BLANK).  
 3483000 4) BEFORE A "/" USED AS A GO-TO DELIMITER.  
 3484000  
 3485000  
 3486000  
 3487000 %CP 4  
 3488000 ANY STRING CAN BE USED AS A STRING NAME, INCLUDING THE NULL STRING.  
 3489000 IN PARTICULAR, THE FOLLOWING IS A VALID EXPRESSION:  
 3490000  
 3491000 \$"  
 3492000  
 3493000  
 3494000  
 3495000 %CP 5  
 3496000 IF A GO-TO PART HAS BOTH A SUCCESS AND A FAILURE EXIT, THEN THE "S"  
 3497000 OR "F" CAN BE OMITTED FROM THE SECOND PART, THUS:  
 3498000  
 3499000 /S(A)(B) MEANS /S(A)F(B)  
 3500000 :F(XA)(X7) MEANS :F(XA)S(X7)  
 3501000  
 3502000  
 3503000  
 3504000 %CP 11  
 3505000 SEVERAL STATEMENTS CAN BE PUT ON ONE CARD. THEY ARE SEPARATED BY  
 3506000 SEMICOLONS (";"). CONTINUATION CARDS CAN BE USED AS USUAL. OF  
 3507000 COURSE, SINCE A LABEL MUST START IN COLUMN 1, ONLY THE FIRST STATEMENT  
 3508000 ON A CARD CAN BE LABELED. FOR EXAMPLE,  
 3509000  
 3510000 L A ".,." = ".,." /S(L); B = A ".,." B /(LOOP)  
 3511000  
 3512000 MEANS THE SAME AS:  
 3513000  
 3514000 L A ".,." = ".,." /S(L)  
 3515000 B = A ".,." B /(LOOP)  
 3516000  
 3517000  
 3518000  
 3519000 %CP 15  
 3520000 NO INTERNAL DISTINCTION IS MADE BETWEEN FORMAL PARAMETERS AND LOCAL  
 3521000 VARIABLES OF PROGRAM-DEFINED FUNCTIONS. IF A FUNCTION IS CALLED WITH  
 3522000 MORE ACTUAL PARAMETERS THAN THERE ARE FORMAL PARAMETERS, THE EXTRA  
 3523000 PARAMETERS WILL BE ASSIGNED TO LOCAL VARIABLES. FOR EXAMPLE, IF F()  
 3524000 IS DEFINED BY:  
 3525000  
 3526000 DEFINE("F(A,B)", "F1", "C,D")  
 3527000  
 3528000 THEN IF THE PROGRAM CALLS F("+", "-", "x"), THE VALUES AT THE ENTRY

3529000 POINT, F1, WILL BE:

3530000

3531000 A = "+"

3532000 B = "-"

3533000 C = "x"

3534000 D = ""

3535000

3536000

3537000

3538000 %APPENDIX A

3539000 %PAGE 1

3540000 APPENDIX A. BNF DEFINITIONS FOR SNOBOL3.

3541000

3542000 <INSTRUCTION> ::= <LABELED INST>

3543000 ::= <UNLABELED INST>

3544000 <LABELED INST> ::= <LABEL> <UNLABELED INST>

3545000 <UNLABELED INST> ::= <BLANKS> <INST>

3546000 <INST> ::= <RULE>

3547000 ::= <GO-TO PART>

3548000 ::= <RULE> <GO-TO PART>

3549000 <RULE> ::= <STR REF>

3550000 ::= <STR REF> <PATTERN>

3551000 ::= <STR REF> <REPLACEMENT>

3552000 ::= <STR REF> <PATTERN> <REPLACEMENT>

3553000 <STR REF> ::= <ELEMENT>

3554000 <PATTERN> ::= <ELEMENT>

3555000 ::= <STR VAR>

3556000 ::= <PATTERN> <PATTERN>

3557000 ::= <PATTERN> <BACK REF>

3558000 <REPLACEMENT> ::= <RPL DELIMITER> <EXPRESSION>

3559000 <RPL DELIMITER> ::= ←

3560000 ::= =

3561000 <EXPRESSION> ::= <ELEMENT>

3562000 ::= <EXPRESSION> <CONCATENATE OP> <ELEMENT>

3563000 ::= <EMPTY>

3564000 <ELEMENT> ::= <NAMED EXPR>

3565000 ::= <VALUE EXPR>

3566000 <NAMED EXPR> ::= <IDENTIFIER>

3567000 ::= \$ <ELEMENT>

3568000 <VALUE EXPR> ::= <LITERAL>

3569000 ::= <GROUPING>

3570000 ::= <FCT CALL>

3571000 ::= <ARITH EXPR>

3572000 <CONCATENATE OP> ::= <BLANKS>

3573000 <IDENTIFIER> ::= <LETTER>

3574000 ::= <IDENTIFIER><LETTER>

3575000 ::= <DIGIT>

3576000 ::= <IDENTIFIER><DIGIT>

3577000 ::= .

3578000 ::= <IDENTIFIER> .

3579000 <LITERAL> ::= " <STRING> "

3580000 <GROUPING> ::= ( <EXPRESSION> )

3581000 <FCT CALL> ::= <FCT NAME>( <PARAMS> )

3582000 <FCT NAME> ::= <IDENTIFIER>

3583000 <PARAMS> ::= <EXPRESSION>

3584000 ::= <PARAMS> , <EXPRESSION>

3585000 <ARITH EXPR> ::= <A-TERM>



```

3586000 ::= " <A-TERM>
3587000 <A-TERM> ::= <A-TERM> <ADD OP> <M-TERM>
3588000 ::= <M-TERM>
3589000 <M-TERM> ::= <M-TERM> <MUL OP> <E-TERM>
3590000 ::= <E-TERM>
3591000 <E-TERM> ::= <E-TERM> <EXP OP> <ARITH ELT>
3592000 ::= <ARITH ELT>
3593000 <ARITH ELT> ::= <NAMED EXPR>
3594000 ::= <GROUPING>
3595000 ::= <FCT CALL>
3596000 ::= <LITERAL>
3597000 <ADD OP> ::= +
3598000 ::= =
3599000 <MUL OP> ::= *
3600000 ::= x
3601000 ::= /
3602000 <EXP OP> ::= **
3603000 <PATTERN> ::= <ELEMENT>
3604000 ::= <STR VAR>
3605000 ::= <PATTERN> <PATTERN>
3606000 ::= <PATTERN> <BACK REF>
3607000 <STR VAR> ::= <ARB VAR>
3608000 ::= <BAL VAR>
3609000 ::= <FIXED-LENGTH VAR>
3610000 <ARB VAR> ::= **
3611000 ::= * <NAMED EXPR> *
3612000 <BAL VAR> ::= *()*
3613000 ::= *( <NAMED EXPR> )*
3614000 <FIXED-LENGTH VAR> ::= */ <EXPRESSION> *
3615000 ::= * <NAMED EXPR> / <EXPRESSION> *
3616000 <BACK REF> ::= <NAMED EXPR>
3617000 <GO-TO PART> ::= <GO-TO DELIMITER> <GO-TOS>
3618000 <GO-TO DELIMITER> ::= <BLANKS> /
3619000 ::= :
3620000 <GO-TOS> ::= <UNCONDITIONAL GO-TO>
3621000 ::= <SUCCESS GO-TO>
3622000 ::= <FAILURE GO-TO>
3623000 ::= <SUCCESS GO-TO> <FAILURE GO-TO>
3624000 ::= <FAILURE GO-TO> <SUCCESS GO-TO>
3625000 <UNCONDITIONAL GO-TO> ::= ( <LABEL EXPR> )
3626000 <SUCCESS GO-TO> ::= SC ( <LABEL EXPR> )
3627000 <FAILURE GO-TO> ::= FC ( <LABEL EXPR> )
3628000 <LABEL EXPR> ::= <RESERVED LABEL>
3629000 ::= <LABEL>
3630000 ::= <COMPUTED LABEL>
3631000 <RESERVED LABEL> ::= END
3632000 ::= RETURN
3633000 ::= FRETURN
3634000
3635000
3636000
3637000 %APPENDIX B
3638000 %PAGE 1
3639000 APPENDIX B. BNF NOTATION.
3640000
3641000 FOR THOSE NOT FAMILIAR WITH "BNF" (BACKUS-NAUR FORM OR
3642000 BACKUS NORMAL FORM), A BRIEF DESCRIPTION FOLLOWS. BNF IS A

```

3643000 SOMEWHAT STANDARDIZED METHOD OF PRECISELY NOTATING SYNTACTIC  
3644000 DEFINITIONS OF CERTAIN TYPES.  
3645000 QUANTITIES WHICH MUST APPEAR AS A CERTAIN STRING OF CHARACTERS  
3646000 ARE REPRESENTED BY THAT STRING OF CHARACTERS. VARIABLE STRINGS  
3647000 ARE GIVEN A NAME, WHICH IS SURROUNDED BY "BNF BRACKETS", "<"  
3648000 AND ">". FOR EXAMPLE, THE MCP CONTROL CARD:  
3649000

3650000 ?DATA <NAME>

3651000  
3652000 MUST CONTAIN THE LITERAL STRING "?DATA" FOLLOWED BY A <NAME>,  
3653000 WHICH MUST BE DEFINED ELSEWHERE. SOME EXAMPLES OF CARDS THAT  
3654000 SATISFY THIS FORM ARE:  
3655000

3656000 ?DATA ALPHA

3657000 ?DATA XXZZYY

3658000 ?DATA DATA

3659000 ?DATA AB13X47

3660000

3661000 A VARIABLE ITEM MAY BE DEFINED BY USING A "BNF EQUATION" TO  
3662000 DEFINE IT IN TERMS OF OTHER ITEMS. A DEFINITION IS OF THE FORM:  
3663000

3664000 <NAME> ::= <DEFINITION>

3665000

3666000 FOR EXAMPLE, A FUNCTION CALL MAY BE DEFINED BY:

3667000

3668000 <FCT CALL> ::= <FCT NAME>( <PARAMS> )

3669000

3670000 WHERE <FCT NAME> AND <PARAMS> ARE DEFINED BY OTHER EQUATIONS.

3671000 IF THERE ARE SEVERAL ALTERNATE DEFINITIONS OF AN OBJECT, SEVERAL  
3672000 EQUATIONS ARE GIVEN. IF A VARIABLE HAS SEVERAL DEFINITIONS, THEN  
3673000 THAT VARIABLE HAS ALTERNATE DEFINITIONS, ANY OF WHICH CAN

3674000 APPLY IN A GIVEN INSTANCE. THIS WAY, RECURSIVE DEFINITIONS ARE

3675000 SIMPLE (AND QUITE COMMON). FOR EXAMPLE:  
3676000

3677000 <PARAMS> ::= <EXPRESSION>

3678000 <PARAMS> ::= <PARAMS> , <EXPRESSION>

3679000

3680000 THIS MEANS THAT <PARAMS> CAN BE A SINGLE EXPRESSION, OR IT CAN

3681000 BE ANY NUMBER OF EXPRESSIONS SEPARATED BY COMMAS. WHEN SEVERAL

3682000 CONSECUTIVE DEFINITIONS ARE GIVEN FOR THE SAME VARIABLE, IT IS

3683000 RATHER CUSTOMARY (TO MAKE READING EASIER) TO OMIT THE LEFT-HAND

3684000 SIDE OF THE EQUATION IN ALL BUT THE FIRST, SO THE ABOVE DEFINITION

3685000 COULD HAVE BEEN:  
3686000

3687000 <PARAMS> ::= <EXPRESSION>

3688000 ::= <PARAMS> , <EXPRESSION>

3689000

3690000 FOR LOTS OF EXAMPLES OF BNF DEFINITIONS, SEE THE PREVIOUS APPENDIX.

3691000 NOTE THAT A FEW ITEMS ARENT DEFINED--IN PARTICULAR, <BLANKS> AND

3692000 <EMPTY> ARE CONSIDERED TOO OBVIOUS TO BOTHER DEFINING. IT IS

3693000 RATHER COMMON FOR SOME TERMS TO BE DEFINED BY ENGLISH-LANGUAGE

3694000 DEFINITIONS, WHEN BNF IS UNSUITED FOR THE JOB. THIS IS DONE FOR

3695000 A FEW SYMBOLS IN THE SNOBOL3 DEFINITIONS--SEE THE APPROPRIATE

3696000 SECTION OF THE MANUAL FOR THEIR DEFINITIONS.  
3697000

3698000

3699000

3700000 %APPENDIX C  
 3701000 %PAGE 1  
 3702000 APPENDIX C. 85500 CHARACTER SET.  
 3703000

3704000	CHAR	OCT	DEC	PUNCH	COMMENTS
3705000	0	0	0	0	
3706000	1	1	1	1	
3707000	2	2	2	2	
3708000	3	3	3	3	
3709000	4	4	4	4	
3710000	5	5	5	5	
3711000	6	6	6	6	
3712000	7	7	7	7	
3713000	8	10	8	8	
3714000	9	11	9	9	
3715000	#	12	10	3-8	
3716000	@	13	11	4-8	
3717000	?	14	12	2-8	OR ANY ILLEGAL PUNCH
3718000	:	15	13	5-8	
3719000	>	16	14	6-8	
3720000	≥	17	15	7-8	TELETYPE: DISCONNECT
3721000	+	20	16	12-2-8	
3722000	A	21	17	12-1	
3723000	B	22	18	12-2	
3724000	C	23	19	12-3	
3725000	D	24	20	12-4	
3726000	E	25	21	12-5	
3727000	F	26	22	12-6	
3728000	G	27	23	12-7	
3729000	H	30	24	12-8	
3730000	I	31	25	12-9	
3731000	.	32	26	12-3-8	
3732000	[	33	28	12-4-8	
3733000	&	34	29	12	
3734000	(	35	30	12-5-8	
3735000	<	36	30	12-6-8	
3736000	←	37	31	12-7-8	TELETYPE: END-OF-MESSAGE
3737000	x	40	32	11-2-8	TELETYPE: BACKWARD SLASH
3738000	J	41	33	11-1	
3739000	K	42	34	11-2	
3740000	L	43	35	11-3	
3741000	M	44	36	11-4	
3742000	N	45	37	11-5	
3743000	O	46	38	11-6	
3744000	P	47	39	11-7	
3745000	Q	50	40	11-8	
3746000	R	51	41	11-9	
3747000	\$	52	42	11-3-8	
3748000	*	53	43	11-4-8	
3749000	-	54	44	11	
3750000	)	55	45	11-5-8	
3751000	;	56	46	11-6-8	
3752000	≤	57	47	11-7-8	TELETYPE: APOSTROPHE, CARRIAGE RETURN
3753000	BLANK	60	48	NONE	
3754000	/	61	49	0-1	
3755000	S	62	50	0-2	
3756000	T	63	51	0-3	

Moore Business Forms, Inc. sv 1412T

3757000	U	64	52	0-4	
3758000	V	65	53	0-5	
3759000	W	66	54	0-6	
3760000	X	67	55	0-7	
3761000	Y	70	56	0-8	
3762000	Z	71	57	0-9	
3763000	,	72	58	0-3-8	
3764000	%	73	59	0-4-8	
3765000	≠	74	60	0-2-8	TELETYPE: UP-ARROW, LINE FEED
3766000	=	75	61	0-5-8	
3767000	]	76	62	0-6-8	
3768000	"	77	63	0-7-8	

3769000  
3770000 %APPENDIX D  
3771000 %PAGE 1  
3772000 APPENDIX D. RUNNING JOBS FROM A TELETYPE.

3773000  
3774000 THIS APPENDIX IS INTENDED TO DESCRIBE THE SOMEWHAT INTRICATE  
3775000 METHODS OF RUNNING A PROGRAM (IN PARTICULAR, A SNOBOL PROGRAM) FROM  
3776000 A TELETYPE, AND HOPEFULLY TO SOMEWHAT AMELIORATE THE INITIAL SHOCK  
3777000 OF TRYING TO USE THIS RATHER PRIMITIVE AND FRUSTRATING INTERACTIVE  
3778000 I/O DEVICE. THIS DESCRIPTION IS FAR FROM COMPLETE, AND DUE TO THE  
3779000 CONSTANT REVISION (AND OCCASIONALLY IMPROVEMENT) OF THE MCP, DETAILS  
3780000 CAN BE EXPECTED TO CHANGE AT ANY TIME, USUALLY WITHOUT WARNING. THE  
3781000 NOVICE IS WARNED THAT THE MCP WAS DESIGNED FOR THE USE OF PEOPLE WHO  
3782000 ARE VERY FAMILIAR WITH IT, AND DOES VERY LITTLE TO HELP THOSE WHO DONT  
3783000 KNOW HOW IT BEHAVES.

3784000 THE FIRST THING NECESSARY IS TO "LOG IN", THAT IS, TO TELL THE MCP  
3785000 WHO YOU ARE. THIS IS DONE BY TYPING:

3786000  
3787000 ?LI <I.D.>←

3788000  
3789000 AT THE UNIVERSITY OF WISCONSIN, <I.D.> IS THE USER AND PROJECT  
3790000 NUMBERS, IN THE FORM:

3791000  
3792000 <I.D.> = <USER#> / <PROJ#>  
3793000

3794000 THE MCP WILL RESPOND TO THE LOG-IN MESSAGE WITH SOME SORT OF ACKNOWLED-  
3795000 GEMENT, AND THE USER CAN THEN INDICATE WHAT PROGRAM HE WANTS TO RUN  
3796000 BY TYPING A ??EXECUTE CONTROL CARD. THIS MESSAGE CONTAINS THE SAME  
3797000 INFORMATION AS THE MCP CONTROL CARDS DESCRIBED IN SECTION 2.1, WITH  
3798000 THE FOLLOWING CHANGES:

- 3799000
- 3800000 1) ALL CONTROL CARDS MUST BE SENT TOGETHER AS ONE MESSAGE. THIS
  - 3801000 MESSAGE CAN BE LONGER THAN ONE LINE, SINCE THE RETURN AND LINE-FEED
  - 3802000 KEYS ON THE TELETYPE DO NOT TRANSMIT ANYTHING.
  - 3803000 2) THE FIRST CONTROL CARD IS THE EXECUTE CARD, WHICH IS PRECEDED
  - 3804000 BY TWO "?"-S. ALL OTHER CONTROL CARDS ARE PRECEDED BY ";"
  - 3805000 RATHER THAN "?". THE USER CARD IS NOT USED, SINCE THAT INFORMA-
  - 3806000 TION WAS GIVEN IN THE LOG-IN MESSAGE.
  - 3807000 3) THE FILE PROGRAM NEED NOT BE ASSIGNED, SINCE THE TELETYPE IS THE
  - 3808000 INPUT DEVICE FOR THE COMPILER.

3809000  
3810000 %CP 9  
3811000 SOME EXAMPLES OF EXECUTE MESSAGES ARE:

3812000  
3813000 ??EXECUTE SNOBOL/SNOBOL←

Moore Business Forms, Inc. sv

3814000  
3815000 ??EXECUTE SNOBOL/SNOBOL;PROCESS=5;IO=10;  
3816000 FILE CARD=BLEEK/GRUNK SERIAL←  
3817000  
3818000 ??EXECUTE SNOBOL/SNOBOL;CORE=19000;PROCESS=20;IO=30←  
3819000  
3820000  
3821000 %CP 21  
3822000 WHEN THE EXECUTE MESSAGE IS TYPED, THE MCP WILL RESPOND IN ONE OF  
3823000 TWO WAYS. IF THERE IS ROOM FOR THE PROGRAM, THE RESPONSE WILL BE:  
3824000  
3825000 I:SNOBOL/SNOBOL=J BOJ <TIME>  
3826000  
3827000 IF THERE IS NOT ENOUGH ROOM, WHICH IS MUCH MORE LIKELY, THE RESPONSE  
3828000 WILL BE OF THE FORM:  
3829000  
3830000 I:SNOBOL/SNOBOL=J SCHEDULED <TIME>  
3831000  
3832000 THE USER MUST THEN FIND SOMETHING ELSE TO OCCUPY HIS TIME UNTIL SPACE  
3833000 BECOMES AVAILABLE AND THE "BOJ" MESSAGE IS TYPED BY THE MCP. USERS ARE  
3834000 ADVISED TO BRING SOMETHING TO READ WHEN ATTEMPTING TO RUN DURING  
3835000 BUSY PARTS OF THE DAY.  
3836000 WHEN THE BOJ MESSAGE APPEARS, THE COMPILER IS RUNNING AND WAITING  
3837000 FOR INPUT. THE USER CAN THEORETICALLY TYPE ANYTHING THAT IS VALID  
3838000 PROGRAM MATERIAL EXCEPT FOR CONTINUATION CARDS. IN PRACTICE, USERS  
3839000 ARE ADVISED TO FIRST GET THEIR PROGRAM IN A DISK FILE (THERE ARE  
3840000 SEVERAL EDITING PROGRAMS AROUND, AS WELL AS A PROGRAM TO COPY CARD  
3841000 DECKS TO DISK FILES), AND NOT ATTEMPT TO TYPE THE PROGRAM ITSELF  
3842000 DIRECTLY TO THE COMPILER.  
3843000  
3844000 %CP 21  
3845000 WHEN A PROGRAM "CARD" IS TYPED, THE COMPILER WILL COMPILE IT AND  
3846000 THEN RESPOND WITH A RETURN/LINE FEED, AND THE USER CAN THEN TYPE  
3847000 THE NEXT LINE. ONE EXCEPTION TO THIS IS THAT WHEN A -COMPILE CONTROL  
3848000 CARD IS TYPED, THE COMPILER TYPES OUT A NUMBER EVERY TENTH INSTRUCTION  
3849000 AS IT COMPILES, AND AT THE END TYPES A MESSAGE SAYING IT IS DONE. THIS  
3850000 WAY THE USER CAN TELL HOW FAST THE COMPILER IS RUNNING, AND IF IT IS  
3851000 VERY SLOW, HE CAN DO SOMETHING ELSE (LIKE GET A CUP OF COFFEE) WHILE  
3852000 HE WAITS FOR IT TO FINISH.  
3853000 USUALLY, THE USER WILL TYPE IN A SERIES OF SNOBOL CONTROL CARDS  
3854000 THAT WILL CAUSE THE PROGRAM TO BE LOADED OR COMPILED FROM ONE OR  
3855000 MORE DISK FILES, AND THEN INITIATE EXECUTION BY TYPING AN END CARD.  
3856000 AS SOON AS THE END CARD IS TYPED, THE COMPILER TRANSFERS TO THE  
3857000 INTERPRETER, AND THE PROGRAM IS RUNNING. SEE SECTION 10.4 FOR METHODS  
3858000 OF DOING I/O WITH A TELETYPE.  
3859000 IF A MISTAKE IS MADE AND DISCOVERED BEFORE IT IS TRANSMITTED TO  
3860000 THE COMPILER, THE WHOLE THING CAN BE ERASED AND RETYPED BY TYPING  
3861000 "#←". THE COMPILER WILL RESPOND WITH "TRY AGAIN", AND THE CARD  
3862000 CAN THEN BE TYPED OVER. THIS USUALLY WORKS BETTER THAN TRYING TO  
3863000 USE THE TELETYPES RATHER AWKWARD (AND NOT ALWAYS SUCCESSFUL) METHODS  
3864000 FOR ERASING MISTAKES.  
3865000  
3866000 %CP 25  
3867000 SOME EXAMPLES OF INPUTS TO THE COMPILER ARE:  
3868000  
3869000  
3870000

```

3871000 -LIST←
3872000 -LOAD MRG/GROUP←
3873000 -COMPILE SLIB/DIFF←
3874000 -LIMIT RULES TO 5000←
3875000 END INIT←
3876000
3877000
3878000
3879000 -LIST←
3880000 -SIZE 370←
3881000 -COMPILE MARK/SINT←
3882000 -LIST 5←
3883000 -COMPILE MARK/INT1←
3884000 -COMPILE MARK/DIF3←
3885000 -DEBUG←
3886000 END←
3887000
3888000
3889000
3890000 -LIST 2←
3891000 -COMPILE GEO/TRAJ3←
3892000 -LIBRARY GEO/TRJ3←
3893000 END GD←
3894000
3895000
3896000
3897000 -LIST 1←
3898000 -COMPILE ML/TRY←
3899000 -DEBUG←
3900000 INIT TRACES("N","J","AL1","PN")←
3901000 TRACE("HINT","REF") : (BEGIN)←
3902000 END INIT←
3903000
3904000
3905000
3906000
3907000 %SECTION "INDEX."
3908000 %PAGE 1
3909000 INDEX.
3910000
3911000 THIS IS RATHER INCOMPLETE; USERS ARE CORDIALLY INVITED TO SUBMIT
3912000 THEIR SUGGESTIONS AS TO WHAT SHOULD BE ADDED.
3913000
3914000
3915000 ARITHMETIC...5.4
3916000 BACK REFERENCING...6.3
3917000 CONTROL CARDS...2
3918000 MCP...2.1
3919000 FROM TELETYPE...APP.0
3920000 SNOBOL...2.2
3921000 DEBUGGING AIDS...11
3922000 INTERACTIVE...11.2
3923000 TRACING...11.1
3924000 ENTRY POINTS
3925000 FUNCTIONS...2.2(-DEFINE),9(DEFINE),3.2.1
3926000 PROGRAM...3.2,3.0
3927000 FAILURE...8

```

Moore Business Forms, Inc. sv 14121

3928000 OF PATTERNS...6.6  
 3929000 OF RULES...4  
 3930000 OF I/O...10,10.5,10.4.1  
 3931000 OF FUNCTIONS  
 3932000 INTRINSICS...9  
 3933000 DEFINED...3.2  
 3934000 FUNCTIONS  
 3935000 ARITHMETIC...5.4.3,9  
 3936000 CALLS ON...5.3  
 3937000 DEFINED...3.2,5.3  
 3938000 INTRINSIC...9  
 3939000 ANCHOR()...6.4,9  
 3940000 DEFINE()...9,2.2(-DEFINE),3.2,3.2.4  
 3941000 TRACING...11.1  
 3942000 INDIRECTION...5.1.3  
 3943000 IN GO-TO PART...7.4  
 3944000 INPUT...SEE "I/O"  
 3945000 I/O...10  
 3946000 FAILURE...10.4.1,8.1  
 3947000 FILES...10.1,10.2,10.3  
 3948000 TELETYPES...10.4  
 3949000 LABELS...7,3.0,3.1 (SEE ALSO "ENTRY POINTS")  
 3950000 IN GO-TO PARTS...7.2  
 3951000 COMPUTED...7.4  
 3952000 RESERVED...7.3  
 3953000 OUTPUT...SEE "I/O"  
 3954000 PATTERNS...6  
 3955000 ANCHORED...6.4  
 3956000 FAILURE...8.1,4.3,4.5,6.6  
 3957000 I/O WITHIN...10.5  
 3958000 MATCHING...4.3,4.5  
 3959000 STRING VARIABLES...6.2,6.5  
 3960000 ARBITRARY...6.2.1,6.6  
 3961000 ASSIGNMENT TO...6.5,6.6  
 3962000 BALANCED...6.2.2,6.6  
 3963000 FIXED-LENGTH...6.2.3  
 3964000 NAMES OF...6.2  
 3965000 OUTPUT ASSOCIATED WITH...6.5,10.5  
 3966000  
 3967000  
 3968000

LABEL 00000000LPAPRNT00176177?USER=SPO ; EXECUTE COPY /SPO

COPY /SPO