NATIONAL BUREAU OF STANDARDS REPORT

8171

A PROGRAMMED FORMALIZER FOR A FRAGMENT OF ENGLISH

by

Sylvan Cappell

To the

National Science Foundation
(Grant No. GN 107)

U. S. DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS

NATIONAL BUREAU OF STANDARDS REPORT

NBS PROJECT
11-11-11404

January 15, 1964

NBS REPORT
8171

A PROGRAMMED FORMALIZER FOR A FRAGMENT OF ENGLISH

by

Sylvan Cappell

Applied Mathematics Division

To the

National Science Foundation
(Grant No. GN 107)

NBS

U. S. DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS

A PROGRAMMED FORMALIZER FOR A FRAGMENT OF ENGLISH

This paper describes a computer program written to translate English into symbolic logic by an algorithm developed by Walter Sillars.[1/] This algorithm, termed "formalizer," has been written for use in a picture language machine and the vocabulary it applies to is directed to that purpose.[2/] As input the program accepts a parsed sentence, one which has been analyzed grammatically. The parser used is described in a paper by Donald Cohen.[3/] The formalizer program is quite general and, like the parser, can be modified or expanded for other grammars than the one discussed here.

Sillars' paper describes a formalizer written for a grammar for a fragment of English, Grammar 12R. A corrected version of the grammar and formalizer appears in Appendix I. Grammar 12R is a subset of Grammar 12 developed by B. Kirk Rankin, III.[4/]

The formalizer program was written in the COMIT programming language, knowledge of which is assumed in this paper. The program appears in Appendix II together with model input and output.

The parsed input describes which rules of Grammar 12R would be used in the generation of the sentence used as input to the parser. The existence of discontinuous rules in a grammar would necessitate the rearrangement of the rule numbers in the output of the parser, in the general case. However, for the special case of Grammar 12R, this is unnecessary.

The formalizer program will accept any number of parser inputs which are separated by a card with the word NEW on it. The program terminates when there are no more input cards to be read in.

The output of the formalizer program is in symbolic logic. A variable is symbolized by a subscripted X. UQ is the symbol used to represent the universal quantifier, EQ the existential quantifier, UN the Sillars U quantifier (Russell and Whitehead's E!), CNJ the "conjunction" and * - represents Church's ꓕ. All other symbols are the same as Sillars'.

Throughout most of the program, what Sillars calls pseudo-wffs are stored in the workspace. The program uses only shelves 1-6.

The first part of the program, the part before the rule named GRAM, reads in the input cards until it reaches one with NEW on it. The program assumes that consecutive digits form part of the same rule number and that non-digital characters separate any two rule numbers. The program assembles the rule numbers in the order that they are to be used on shelf 1 and places an *B after the last rule number. An X_1 (x/.1 in COMIT) is stored on shelf 3. Whenever Sillars' algorithm calls for a new variable the contents of shelf 3 are used and the subscript value of the X in shelf 3 is increased by 1. The program places the initial symbol, TOP, inclosed in brackets in the workspace. Brackets inclose all pseudo-wffs. *B is used internally to represent [and *C represents] . In the output, however, they are converted to "(" and ")" respectively.

The program then branches to the rule named EXIT. It picks up the first grammar rule number on shelf 1 and looks it up in the list GRAM. It then branches to the formalizer instructions corresponding to the grammar rule number, erases the rule number and returns to EXIT. Those grammar rule numbers which have no corresponding formalizer instructions, the ones which Sillars calls not applicable, NA, are simply erased and control returns to EXIT.

After the last grammar rule number has been picked up the program picks up the *B on shelf 1 and finds it in the list GRAM. It then branches to STOP and the symbolic logic statement corresponding to the original sentence is given as output. The program then prepares to read in the next input sentence.

The subroutine named FUNC (the name of its first rule) locates the shortest pseudo-wff containing all occurrences of NV and places an *M immediately before it and an *N immediately after it. It terminates one of the subrules of OUT, which then branches back into the main body of the program.

The program can be modified quite readily. For each new rule assign a new number. Place a card under GRAM with the number on the left and a zero in the right half and branch to a point in the program where the actual formalizer rules are to be executed. Control must finally return to EXIT.

If an instruction requires finding the shortest pseudo-wff containing all occurrences of some constituent α, substitute NV for each α, indicate a subrule of OUT to be used, and branch to FUNC. (Subrules can, of course, be added to OUT.) If so desired, every NV can then be replaced by α.

To substitute an X_i for each NV and to then increase the subscript of the X in shelf 3 by 1, branch to XOG. Control will automatically be returned to EXIT.

The program treats any symbol beginning with PC or SC as if it consisted only of one of those two respective symbols. The processing of a PC rule without a quantifier begins at APC1 and with a quantifier at APC2 and all SC rules at ASC. Each rule number associated with a DEP or DES rule leads in the program to a series of instructions where quantifiers are generated. The rules UNAT, SQUAT, TQUAT and NQUAT substitute the appropriate quantifier and branch to QUADS where CNJ is substituted for $\square$. UQUAT substitutes a UQ for Q and PLY for $\square$. More quantifier rules, accomplishing similar functions, can be added to the program.

A rule can be taken out simply by removing the card with the appropriate rule number under GRAM.

In the final analysis this program is to be used in the predicate evaluator in the picture language machine. The output format in which all propositions which form part of other propositions are inclosed in parentheses should facilitate its use for this application. Thus a proposition of the general form $p \wedge q \supset r$ would be given in output as $(((p) \wedge (q)) \supset r)$.

In determining the truth or falsehood of a proposition which has several component propositions, the predicate evaluator must selectively choose which of the latter to begin with. Thus in evaluating propositions of the form $p \vee q \supset r$ it will usually be advisable to begin by analyzing the truth-value of r. If, however, r is in some sense very complex, it might be preferable to begin with p.

APPENDIX I. The Grammar and Formalizer

Here is the revised grammar for which the program in Appendix II has been written. For the notational conventions, see Sillars.[1/]

TOP	=	CL1	SR
TOP	=	CL3	
TOP	=	CL5	
TOP	=	CL7	
TOP	=	CL11	
TOP	=	CL13	
TOP	=	CL15	
TOP	=	CL17	
AAP1	=	WPN + COM1	AAP1 = WNP + COM1
AAP1	=	COM1	SR
AAP1	=	WOP + J	AAP1 = WOP + J
AAS	=	WN + COM1	AAS = WN + COM1
AAS	=	COM1	SR
AAS	=	WO + J	AAS = WO + J
APN	=	ET + PNBET	SR(2)
AR	=	are	NA
ARNTC	=	aren't	

CL1	=	SUB1X + PRE0	
CL3	=	SUB6X + PRE1	
CL5	=	SUB4X + PRE3	
CL7	=	SUB7W + PRE4	$CL_i = PRE_j(SUB_k)$
CL11	=	SUB10 + PRE11	
CL13	=	SUB11 + PRE13	
CL15	=	SUB13 + PRE15	
CL17	=	SUB16 + PRE17	
CO	=	,	NA
COL	=	Color	COL = Smc
COL	=	Size	COL = Smz
COM1	=	LP	
COM1	=	RE	
COM2	=	COM1	
COM2	=	PNP2	SR
COM3	=	COM1	
COM3	=	SNP2	

DEP12	=	some
DEP12	=	the
DEP12	=	two
DEP12	=	all
DEP12	=	no
DEP2	=	some
DEP2	=	the
DEP2	=	two
DEP2A	=	any
DEP5S	=	some
DEP5S	=	two
DEP5S	=	no
DES1	=	a
DES1	=	one
DES13	=	the
DES13	=	a
DES13	=	every
DES13	=	one
DES13	=	each
DES16	=	the
DES16	=	a
DES16	=	one

Procedure for rewriting (QNV) [NN(NV) □ A(NV)] where NN is either N or NPLUR:

DE is any DEPi or DESi

1. If DES = the, Q = U

   If DE = No or any, (QNV) = $\neg(\exists NV)$

   If quantifier is deleted, Q = $\exists$. In all other cases,

if DE =	then Q =
some	S
the	S
two	T
all	$\forall$
a	$\exists$
one	U
each	$\forall$
every	$\forall$

   Then replace all instances of NV by the next (in alphabetic order) individual variable which has not yet been used.

2. If DE = all, each or every, then □ = $\supset$., otherwise □ = &.

3. Replace every expression of the form Vi(Vj), where Vi and Vj are individual variables, by (Vi = Vj).

DES16	=	each	
DES16	=	every	
DES16	=	no	
DES2	=	the	
DES2	=	a	
DES2	=	one	
DES7	=	a	
DES7	=	one	
DES7	=	no	
DUM1	=	R1 + NPH	DUM1(a,NPH) = R1(a, NPH1, NPH2)
DUM2	=	SNASN	
DUM2	=	PNAPN	SR
DUM2	=	PNBET	
ES	=	ET + SNP2	SR(2)
ET	=	and	NA
F1	=	T2 + G1	
F2	=	T2 + G2	SR(2)
G1	=	right	MT + G1 = Mort; G1 = Rt
G1	=	left	MT + G1 = Molf; G1 = Lf
G1	=	top	MT + G1 = Mtop; G1 = Tp
G1	=	bottom	MT + G1 = Mbot; G1 = Bot
G2	=	center	MT + G2 = Mcen; G2 = Cen
G2	=	middle	MT + G2 = Mmid; G2 = Mid
I	=	is	NA

IA	=	J1	SR
IA	=	J2	
IA	=	J1 + J2	IA(a) = J1(a) & J2 (a)
ISNTC	=	isn't	NA
J	=	J1	SR
J	=	J2	
J1	=	big	J1 = Bg
J1	=	little	J1 = Lt
J1	=	large	J1 = Lg
J1	=	small	J1 = Sm
J2	=	black	J2 = Bk
J2	=	white	J2 = Wh
JER	=	bigger	JER = Bgr
JER	=	littler	JER = Ltr
JER	=	larger	JER = Lgr
JER	=	smaller	JER = Smr
LL	=	Z1	SR
LL	=	Z2	
LP	=	LL	
MO	=	more	NA
MT	=	MO : TN	NA
N	=	triangle	N = Tr
N	=	square	N = Sq
N	=	circle	N = Cir

NPH	=	SNP2	SR
NPH	=	PNP2	SR
NPLUR	=	triangles	NPLUR = Tr
NPLUR	=	squares	NPLUR = Sq
NPLUR	=	circles	NPLUR = Cir
NT	=	not	NT = $\neg$
P	=	in	P = In
P	=	near	P = Nr
P	=	below	P = Bel
P	=	above	P = Ab
P	=	touching	P = Tch
P1	=	to	
P1	=	on	
P1	=	at	NA
P2	=	in	
P3	=	between	P3 = Bet
P4	=	of	NA
PA1	=	IA : AAP1	PA1(a) = IA(a)&AAP1(a)
PA1	=	Ø : AAP1	SR
PA1	=	IA	
PA2	=	IA : AAP1	PA2(a) = IA(a)&AAP1(a)
PA2	=	IA	SR

PC12	=	DEP12:NPLUR	A(PCi) = (QNV) [NPLUR(NV)□A(NV)]
PC2	=	DEP2 :NPLUR	
PC2	=	Ø :NPLUR	
PC2A	=	DEP2A : NPLUR	
PC2A	=	Ø : NPLUR	
PC4	=	DEP2 + NPLUR	
PC4	=	NPLUR	
PC412	=	DEP12 + NPLUR	
PC4A	=	DEP2A + NPLUR	
PC4A	=	NPLUR	
PC4S5	=	DEP5S + NPLUR	
PC4S5	=	NPLUR	
PC4S6	=	PC4S5	SR
PNAPN	=	PNBET + APN	A(PNAPN) = A(PNBET,APN)
PNBET	=	PC2 + PA1	A(PNBET) = PA1(PC2)& A(PC2)
PNBET	=	PC4	SR
PNP12	=	PC412	SR
PNP12	=	PC12 + PA1	A(PNP12) = PA1(PC12) & A(PC12)
PNP2	=	PC2 + PA1	A(PNP2) = PA1(PC2) & A(PC2)
PNP2	=	PC4	SR
PNP2A	=	PC2A + PA1	A(PNP2A) = PA1(PC2A) & A(PC2A)
PNP2A	=	PC4A	SR
PNP6S	=	PC4S6	SR

PPLB	=	P3 + DUM2	PPLB(a) = P3(a,DUM2)
PREO	=	VPO	
PRE1	=	VP1	
PRE11	=	VP11	
PRE13	=	VP13	
PRE15	=	VP15	
PRE17	=	VP17	SR
PRE3	=	VP3	
PRE4	=	VP4	
R	=	R1	
R	=	R2	
R	=	R4	
R	=	R5	
R1	=	LL+P4	SR(1)
R2	=	JER + TN	SR(1)
R4	=	P	SR
R5	=	MT + DUM1	R5(a,b) = MT DUM1(a,b)
RE	=	R + NPH	RE(a) = R(a,NPH)
RE	=	PPLB	SR
SA	=	IA : AAS	SA(a) = IA(a) & AAS(a)
SA	=	IA	SR
SA	=	Ø : AAS	SR
SAME	=	same	NA

SC1	=	DES1 : N	
SC13	=	DES13 : N	
SC16	=	DES16 : N	
SC2	=	DES2 : N	
SC3	=	DES1 + N	A(SCi) = (QNV) [N(NV)□A(NV)]
SC37	=	DES7 + N	
SC4	=	DES2 + N	
SC413	=	DES13 + N	
SC416	=	DES16 + N	
SC7	=	DES7 : N	
SNASN	=	SNP2 + ES	A(SNASN) = A(SNP2,ES)
SNP1	=	SC1 + SA	A(SNP1) = SA(SC1) & A(SC1)
SNP1	=	SC3	SR
SNP13	=	SC413	SR
SNP13	=	SC13 + SA	A(SNP13) = SA(SC13) & A(SC13)
SNP16	=	SC416	SR
SNP16	=	SC16 + SA	A(SNP16) = SA(SC16) & A(SC16)
SNP2	=	SC4	SR
SNP7	=	SC37	SR
SNP7	=	SC7 + SA	A(SNP7) = SA(SC7)& A(SC7)

SUB1	=	PNP2A	
SUB10	=	PNP2	
SUB11	=	PNP12	SR
SUB13	=	SNP13	
SUB16	=	SNP16	
SUB1X	=	SUB1Y	
SUB1Y	=	THERE : SUB1	SR(2)
SUB3	=	SNP1	SR
SUB4X	=	SUB4Y	
SUB4Y	=	THERE : SUB3	SR(2)
SUB6	=	PNP6S	SR
SUB6X	=	SUB6Y	SR
SUB6Y	=	THERE : SUB6	SR(2)
SUB7	=	SNP7	SR
SUB7W	=	SUB7Y	SR
SUB7Y	=	THERE : SUB7	SR(2)
T2	=	the	
TH	=	that	NA
TH	=	which	
THERE	=	there	
THSAM	=	T2 + SAME	NA
TN	=	than	

VPO	=	ARNTC : COM1	VPO = ⅂COM1
VP1	=	AR : COM1	SR(2)
VP11	=	ARNTC + COM2	VP11 = ⅂ COM2
VP13	=	AR + COM2	SR(2)
VP15	=	ISNTC + COM3	VP15 = ⅂ COM3
VP17	=	I + COM3	SR(2)
VP3	=	ISNTC : COM1	VP3 = ⅂ COM3
VP4	=	I : COM1	SR(2)
WA	=	TH + AR	NA
WI	=	TH + I	NA
WN	=	WI + NT	WN = NT
WN	=	WI	WN = Ø
WN	=	NT	WN = NT
WNP	=	WA + NT	WNP = NT
WNP	=	WA	WNP = Ø
WNP	=	NT	WNP = NT
WO	=	WI + NT	WO = NT
WO	=	WI	WO = Ø
WOP	=	WA + NT	WOP = NT
WOP	=	WA	WOP = Ø
Z1	=	P1 + F1	SR(2)
Z2	=	P2 + F2	SR(2)

APPENDIX II. The Formalizer Program

The COMIT program which realizes the algorithm is listed below. To illustrate the output format of the program, we present the formalization of the sentence:

There aren't any triangles on the right.

The parse of this sentence is:

2(39(351(352(381(-THERE))(339(221(195(160(-TRIANGLES))))))
(239(394(35(-AREN*T))(65(148(146(427(172(-ON))(119(378(-THE))
(121(-RIGHT)))))))))

(Where the numbers are names of the rules of 12R which were used in generating it.) The output for this sentence from the formalizer program is:

*- + *(+ *(+ EQ + X / .1 + *) + *(+ *(+ TR + *(+ X /
.1 + *) + * + CNJ + *(+ RT + *(+ X / .1 + *) + *) + *)
+ *) +

APPENDIX II

FORMALIZER

```
STAR     $=*B+TOP+*C+X/.1                  //*Q2 1 2 3,*Q3 4       *
BEGIN    $=                                //*RCR1                 BEGINA
*                                                                  FINISH
BEGINA   N+E+W=1                           //*A4 1                 SUB
BEGINB   $                                 //*Q4 1                 BEGIN
SUB      $1                                //*L1                   ZAS
*        $=*B                              //*Q1 1,*A2 1           *
EXIT     $=*A+1                            //*N1 1,*L 1            GRAM
-ZAS     *0                                //*Q4 1                 SUB
         *1                                //*Q4 1                 SUB
         *2                                //*Q4 1                 SUB
         *3                                //*Q4 1                 SUB
         *4                                //*Q4 1                 SUB
         *5                                //*Q4 1                 SUB
         *6                                //*Q4 1                 SUB
         *7                                //*Q4 1                 SUB
         *8                                //*Q4 1                 SUB
         *9                                //*Q4 1                 SUB
*        $1=                               //*A4 1,*K1,*Q1 1       SUB
-GRAM    *2          =0                                            R2
         *4          =0                                            R4
         *6          =0                                            R6
         *8          =0                                            R8
         *1*0        =0                                            R10
         *1*2        =0                                            R12
         *1*4        =0                                            R14
         *1*6        =0                                            R16
         *2*5        =0                                            R25
         *2*6        =0                                            R26
         *2*7        =0                                            R27
         *2*8        =0                                            R28
         *2*9        =0                                            R29
         *3*0        =0                                            R30
         *3*3        =0                                            R33
         *3*9        =0                                            R39
         *5*3        =0                                            R53
         *5*9        =0                                            R59
         *6*1        =0                                            R61
         *4*1        =0                                            R41
         *4*3        =0                                            R43
         *4*5        =0                                            R45
         *4*7        =0                                            R47
         *6*3        =0                                            R63
         *6*4        =0                                            R64
         *6*5        =0                                            R65
         *6*6        =0                                            R66
         *6*7        =0                                            R67
         *6*8        =0                                            R68
         *6*9        =0                                            R69
         *7*0        =0                                            R70
         *7*1        =0                                            SQUAT
         *7*2        =0                                            SQUAT
         *7*3        =0                                            TQUAT
         *7*4        =0                                            UQUAT
```

*7*5	=0	NQUAT
*7*6	=0	SQUAT
*7*7	=0	SQUAT
*7*8	=0	TQUAT
*8*3	=0	NQUAT
*8*9	=0	SQUAT
*9*0	=0	TQUAT
*9*1	=0	NQUAT
*9*2	=0	EQUAT
*9*3	=0	UNAT
*9*4	=0	SQUAT
*9*5	=0	EQUAT
*9*6	=0	UQUAT
*9*7	=0	UNAT
*9*8	=0	UQUAT
*9*9	=0	SQUAT
*1*0*0	=0	EQUAT
*1*0*1	=0	UNAT
*1*0*2	=0	UQUAT
*1*0*3	=0	UQUAT
*1*0*4	=0	NQUAT
*1*0*5	=0	SQUAT
*1*0*6	=0	EQUAT
*1*0*7	=0	UNAT
*1*0*8	=0	EQUAT
*1*0*9	=0	UNAT
*1*1*0	=0	NQUAT
*1*1*1	=0	R111
*1*1*2	=0	R112
*1*1*3	=0	R113
*1*1*4	=0	R114
*1*1*7	=0	R117
*1*1*9	=0	R119
*1*2*0	=0	R120
*1*2*1	=0	R121
*1*2*2	=0	R122
*1*2*3	=0	R123
*1*2*4	=0	R124
*1*2*5	=0	R125
*1*2*6	=0	R126
*1*3*0	=0	R130
*1*3*1	=0	R131
*1*3*2	=0	R132
*1*3*4	=0	R134
*1*3*5	=0	R135
*1*3*6	=0	R136
*1*3*7	=0	R137
*1*3*8	=0	R138
*1*3*9	=0	R139
*1*4*0	=0	R140
*1*4*1	=0	R141
*1*4*2	=0	R142
*1*4*3	=0	R143
*1*4*4	=0	R144
*1*4*5	=0	R145

*1*4*6	=0	R146
*1*4*7	=0	R147
*1*4*8	=0	R148
*1*5*2	=0	R152
*1*5*3	=0	R153
*1*5*4	=0	R154
*1*5*6	=0	R156
*1*5*8	=0	R158
*1*5*9	=0	R159
*1*6*0	=0	R160
*1*6*1	=0	R161
*1*6*2	=0	R162
*1*6*4	=0	R164
*1*6*6	=0	R166
*1*6*7	=0	R167
*1*6*8	=0	R168
*1*6*9	=0	R169
*1*7*0	=0	R170
*1*7*5	=0	R175
*1*7*7	=0	R177
*1*7*8	=0	R178
*1*7*9	=0	R179
*1*8*0	=0	R180
*1*8*1	=0	R181
*1*8*2	=0	APC2
*1*8*3	=0	APC2
*1*8*4	=0	APC1
*1*8*6	=0	APC2
*1*8*7	=0	APC1
*1*9*0	=0	APC2
*1*9*1	=0	APC1
*1*9*2	=0	APC2
*1*9*4	=0	APC2
*1*9*5	=0	APC1
*2*0*0	=0	APC2
*2*0*1	=0	APC1
*2*0*2	=0	R202
*2*0*5	=0	APC1
*2*0*6	=0	APC2
*2*0*7	=0	R207
*2*0*8	=0	R208
*2*0*9	=0	R209
*2*1*0	=0	R210
*2*1*1	=0	R211
*2*1*2	=0	R212
*2*1*3	=0	R213
*2*1*4	=0	R214
*2*1*5	=0	R215
*2*1*6	=0	R216
*2*2*0	=0	R220
*2*2*1	=0	R221
*2*2*2	=0	R222
*2*3*5	=0	R235
*2*3*6	=0	R236
*2*3*7	=0	R237

*2*3*8	=0	R238
*2*3*9	=0	R239
*2*4*0	=0	R240
*2*4*2	=0	R242
*2*4*4	=0	R244
*2*4*6	=0	R246
*2*4*8	=0	R248
*2*5*2	=0	R252
*2*5*7	=0	R257
*2*9*4	=0	R294
*2*9*5	=0	R295
*2*9*7	=0	R297
*2*9*8	=0	R298
*2*9*9	=0	R299
*3*0*0	=0	R300
*3*0*1	=0	R301
*3*0*3	=0	R303
*3*0*4	=0	R304
*3*0*5	=0	R305
*3*0*6	=0	R306
*3*0*7	=0	R307
*3*0*9	=0	R309
*3*1*0	=0	R310
*3*1*1	=0	R311
*3*1*4	=0	ASC
*3*1*5	=0	ASC
*3*1*6	=0	ASC
*3*1*7	=0	ASC
*3*1*8	=0	ASC
*3*1*9	=0	ASC
*3*2*0	=0	ASC
*3*2*1	=0	ASC
*3*2*2	=0	ASC
*3*2*3	=0	ASC
*3*2*4	=0	R324
*3*2*5	=0	R325
*3*2*6	=0	R326
*3*2*7	=0	R327
*3*2*8	=0	R328
*3*2*9	=0	R329
*3*3*0	=0	R330
*3*3*2	=0	R332
*3*3*3	=0	R333
*3*3*4	=0	R334
*3*3*9	=0	R339
*3*4*2	=0	R342
*3*4*4	=0	R344
*3*4*7	=0	R347
*3*5*0	=0	R350
*3*5*1	=0	R351
*3*5*2	=0	R352
*3*5*6	=0	R356
*3*6*3	=0	R363
*3*6*4	=0	R364
*3*6*9	=0	R369

```
          *3*7*1          =0                    R371
          *3*7*2          =0                    R372
          *3*7*4          =0                    R374
          *3*7*5          =0                    R375
          *3*7*7          =0                    R377
          *3*8*6          =0                    R386
          *3*8*7          =0                    R387
          *3*8*8          =0                    R388
          *3*8*9          =0                    R389
          *3*9*0          =0                    R390
          *3*9*1          =0                    R391
          *3*9*2          =0                    R392
          *3*9*3          =0                    R393
          *3*9*4          =0                    R394
          *3*9*5          =0                    R395
          *3*9*7          =0                    R397
          *3*9*9          =0                    R399
          *4*0*1          =0                    R401
          *4*0*3          =0                    R403
          *4*0*6          =0                    R406
          *4*0*7          =0                    R407
          *4*1*7          =0                    WNNT
          *4*1*8          =0                    WNO
          *4*1*9          =0                    WNNT
          *4*2*0          =0                    WNPNT
          *4*2*1          =0                    WNPO
          *4*2*2          =0                    WNPNT
          *4*2*3          =0                    WONT
          *4*2*4          =0                    WOO
          *4*2*5          =0                    WOPN
          *4*2*6          =0                    WOPO
          *4*2*7          =0                    R427
          *4*2*8          =0                    R428
          *B              =0                    STOP
*         $1=0                                  EXIT
R2        TOP=CL*1                              EXIT
R4        TOP=CL*3                              EXIT
R6        TOP=CL*5                              EXIT
R8        TOP=CL*7                              EXIT
R10       TOP=CL*1*1                            EXIT
R12       TOP=CL*1*3                            EXIT
R14       TOP=CL*1*5                            EXIT
R16       TOP=CL*1*7                            EXIT
R25       AAP*1=WN+COM*1                        EXIT
R26       AAP*1=COM*1                           EXIT
R27       AAP*1=WOP+J                           EXIT
R28       AAS=WN+COM*1                          EXIT
R29       AAS=COM*1                             EXIT
R30       AAS=WO+J                              EXIT
R33       APN=PNBET                             EXIT
R39       CL*1=PRE*0+*(+SUB*1X+*)               EXIT
R53       CL*3=PRE*1+*(+SUB*6X+*)               EXIT
R59       CL*5=PRE*3+*(+SUB*4X+*)               EXIT
R61       CL*7=PRE*4+*(+SUB*7W+*)               EXIT
R41       CL*1*1=PRE*1*1+*(+SUB*1*0+*)          EXIT
```

```
R43        CL*1*3=PRE*1*3+*(+SUB*1*1+*)                                   EXIT
R45        CL*1*5=PRE*1*5+*(+SUB*1*3+*)                                   EXIT
R47        CL*1*7=PRE*1*7+*(+SUB*1*6+*)                                   EXIT
R63        COL=SMC                                                        EXIT
R64        COL=SMZ                                                        EXIT
R65        COM*1=LP                                                       EXIT
R66        COM*1=RE                                                       EXIT
R67        COM*2=COM*1                                                    EXIT
R68        COM*2=PNP*2                                                    EXIT
R69        COM*3=COM*1                                                    EXIT
R70        COM*3=SNP*2                                                    EXIT
UNAT       Q=UN                                                           QUADS
SQUAT      Q=S                                                            *
QUADS      QUAD=CNJ                                                       EXIT
TQUAT      Q=T                                                            QUADS
FQUAT      Q=FQ                                                           QUADS
UQUAT      Q=UQ                                                           *
*          QUAD=PLY                                                       EXIT
NQUAT      *(+Q=*-+*(+FQ                                                  QUADS
R111       DUM*1+*(+$+,+NPH+*)=R*1+2+3+4+5+4+5+6                          EXIT
R112       DUM*2=SNASN                                                    EXIT
R113       DUM*2=PNAPN                                                    EXIT
R114       DUM*2=PNBET                                                    EXIT
R117       ES=SNP*2                                                       EXIT
R119       F*1=G*1                                                        EXIT
R120       F*2=G*2                                                        EXIT
R121       MT+G*1=MORT                                                    EXIT
*          G*1=RT                                                         EXIT
R122       MT+G*1=MOLF                                                    EXIT
*          G*1=LF                                                         EXIT
R123       MT+G*1=MTOP                                                    EXIT
*          G*1=TP                                                         EXIT
R124       MT+G*1=MBOT                                                    EXIT
*          G*1=BOT                                                        EXIT
R125       MT+G*2=MCEN                                                    EXIT
*          G*2=CEN                                                        EXIT
R126       MT+G*2=MMID                                                    EXIT
*          G*2=MID                                                        EXIT
R130       IA=J*1                                                         EXIT
R131       IA=J*2                                                         EXIT
R132       *B+IA+*(+$+*)+*C=1+1+J*1+3+4+5+6+CNJ+1+J*2+3+4+5+6+6           EXIT
R134       J=J*1                                                          EXIT
R135       J=J*2                                                          EXIT
R136       J*1=BG                                                         EXIT
R137       J*1=LT                                                         EXIT
R138       J*1=LG                                                         EXIT
R139       J*1=SM                                                         EXIT
R140       J*2=BK                                                         EXIT
R141       J*2=WH                                                         EXIT
R142       JER=BGR                                                        EXIT
R143       JER=LTR                                                        EXIT
R144       JER=LGR                                                        EXIT
R145       JER=SMR                                                        EXIT
R146       LL=Z*1                                                         EXIT
R147       LL=Z*2                                                         EXIT
```

```
R148         LP=LL                                                              EXIT
R152         N=TR                                                               EXIT
R153         N=SQ                                                               EXIT
R154         N=CIR                                                              EXIT
R156         NO=NPLUR                                                           EXIT
R158         NPH=SNP*2                                                          EXIT
R159         NPH=PNP*2                                                          EXIT
R160         NPLUR=TR                                                           EXIT
R161         NPLUR=SQ                                                           EXIT
R162         NPLUR=CIR                                                          EXIT
R164         *B+NT=*-+1                                                         EXIT
*            NT=*-                                                              EXIT
R166         P=IN                                                               EXIT
R167         P=NR                                                               EXIT
R168         P=BEL                                                              EXIT
R169         P=AB                                                               EXIT
R170         P=TCH                                                              EXIT
R175         P*3=BET                                                            EXIT
R177         PA*1+*(+$+*)=*B+IA+*(+3+*)+*C+CNJ+*B+AAP*1+2+3+4+*C               EXIT
R178         PA*1=AAP*1                                                         EXIT
R179         PA*1=IA                                                            EXIT
R180         PA*2+*(+$+*)=*B+IA+*(+3+*)+*C+CNJ+*B+AAP*1+2+3+4+*C               EXIT
R181         PA*2=IA                                                            EXIT
APC1         PC=NV                                                              APC1
*            $                                 //BACH A,OUT A                   FUNC
APC2         PC=NV                                                              APC2
*            $                                 //BACH B,OUT A                   FUNC
BACH    A                                                                       BACHA
        B                                                                       BACHB
BACHA        Q+$+QUAD=EQ+2+CNJ                                                  BACHB
BACHB        NV=                               //*X3                            CBACH
EQUA         X+*(+X+*)=2+1+*=+3+4                                               EQUA
*            $                                 //*X3                            *
*            $1=1/.I1                          //*X3                            EXIT
CBACH        $1=1+1                            //*X3                            *
*            NV=                               //*N3 1                          BACHB
R202         PC*4S*6=PC                                                         EXIT
R207         PC*6S=PC                                                          EXIT
R208         $1+*(+PNAPN+*)=1+2+PNBET+,+APN+4                                   EXIT
R209         PNBET=NV                                                           R209
*            $                                 //OUT B                          FUNC
XOB          NV=PC                                                              XOB
*            $                                                                  EXIT
R210         PNBET=PC                                                           EXIT
R211         PNP*1*2=NV                                                         R211
*            $                                 //OUT C                          FUNC
XOC          NV=V                                                               XOC
*            $                                                                  EXIT
R212         PNP*1*2=PC                                                         EXIT
R213         PNP*1*2=NV                                                         R213
*            $                                 //OUT B                          FUNC
R214         PNP*2=NV                                                           R214
*            $                                 //OUT B                          FUNC
R215         PNP*2=PC                                                           EXIT
R216         PNP*2=NV                                                          R216
```

```
*        $                                       //OUT C              FUNC
R220     PNP*2A=NV                                                    R220
*        $                                       //OUT B              FUNC
R221     PNP*2A=PC                                                    EXIT
R222     PNP*2A=NV                                                    R222
*        $                                       //OUT C              FUNC
R235     PNP*6S=NV                                                    R235
*        $                                       //OUT C              FUNC
R236     PNP*6S=PC*4S*6                                               EXIT
R237     PNP*6S=NV                                                    R237
*                                            //OUT B                  FUNC
R238     PPLB+*(+$+*)=P*3+2+3+,+DUM*2+4                               EXIT
R239     PRE*0=VP*0                                                   EXIT
R240     PRE*1=VP*1                                                   EXIT
R242     PRE*1*1=VP*1*1                                               EXIT
R244     PRE*1*3=VP*1*3                                               EXIT
R246     PRE*1*5=VP*1*5                                               EXIT
R248     PRE*1*7=VP*1*7                                               EXIT
R252     PRE*3=VP*3                                                   EXIT
R257     PRE*4=VP*4                                                   EXIT
R294     R=R*1                                                        EXIT
R295     R=R*2                                                        EXIT
R297     R=R*4                                                        EXIT
R298     R=R*5                                                        EXIT
R299     R=R*6                                                        EXIT
R300     R*1=IL                                                       EXIT
R301     R*2=JER                                                      EXIT
R303     R*4=P                                                        EXIT
R304     R*5+*(+$+*)=MT+DUM*1+2+3+4                                   EXIT
R305     R*6=COL                                                      EXIT
R306     RE+*(+$+*)=R+2+3+,+NPH+4                                     EXIT
R307     RE=PPLB                                                      EXIT
R309     SA+*(+$+*)=*B+IA+2+3+4+*C+CNJ+*B+AAS+2+3+4+*C                EXIT
R310     SA=IA                                                        EXIT
R311     SA=AAS                                                       EXIT
ASC      SC=NV                                                        ASC
*        $                                       //OUT D  ,BACH B     FUNC
R324     $1+*(+SNASN+*)=1+2+SNP*2+.+ES+4                              EXIT
R325     SNP*1=NV                                                     R325
*        $                                       //OUT E              FUNC
XOE      NV=SC                                                        XOE
*        $                                                            EXIT
R326     SNP*1=SC                                                     EXIT
R327     SNP*1*3=SC                                                   EXIT
R328     SNP*1*3=NV                                                   R328
*        $                                       //OUT E              FUNC
R329     SNP*1*6=SC                                                   EXIT
R330     SNP*1*6=NV                                                   R330
*        $                                       //OUT E              FUNC
R332     SNP*2=SC                                                     EXIT
R333     SNP*7=SC                                                     EXIT
R334     SNP*7=NV                                                     R334
*        $                                       //OUT E              FUNC
R339     SUB*1=PNP*2A                                                 EXIT
R342     SUB*1*0=PNP*2                                                EXIT
```

```
R344        SUB*1*1=PNP*1*2                                    EXIT
R347        SUB*1*3=SNP*1*3                                    EXIT
R350        SUB*1*6=SNP*1*6                                    EXIT
R351        SUB*1X=SUB*1Y                                      EXIT
R352        SUB*1Y=SUB*1                                       EXIT
R356        SUB*3=SNP*1                                        EXIT
R363        SUB*4X=SUB*4Y                                      EXIT
R364        SUB*4Y=SUB*3                                       EXIT
R369        SUB*6=PNP*6S                                       EXIT
R371        SUB*6X=SUB*6Y                                      EXIT
R372        SUB*6Y=SUB*6                                       EXIT
R374        SUB*7=SNP*7                                        EXIT
R375        SUB*7W=SUB*7Y                                      EXIT
R377        SUB*7Y=SUB*7                                       EXIT
R386        V=NV                                               R386
*           $                              //OUT F             FUNC
XOF         NV=V*1                                             XOF
*           $                                                  EXIT
R387        V*1=V*2                                            EXIT
R388        V*1=V*3                                            EXIT
R389        V*2=V*4                                            EXIT
R390        V*3=NV                                             R390
*           $                              //OUT G             FUNC
XOG         NV=                            //*X3               ZOG
*           $=                             //*X3               *
*           $1=1/.I1                       //*X3               EXIT
ZOG         $=1+1                          //*X3               *
*           NV=                            //*N3 1             XOG
R391        V*4=NV                                             R391
*           $                              //OUT H             FUNC
R392        V*4=NV                                             R392
*           $                              //OUT I             FUNC
R393        V*5=V*3                                            EXIT
R394        *B+VP*0=*-+1+COM*1                                 EXIT
*           VP*0=*-+COM*1                                      EXIT
R395        VP*1=COM*1                                         EXIT
R397        *B+VP*1*1=*-+1+COM*2                               EXIT
*           VP*1*1=*-+COM*2                                    EXIT
R399        VP*1*3=COM*2                                       EXIT
R401        *B+VP*1*5=*-+1+COM*1                               EXIT
*           VP*1*5=*-+COM*3                                    EXIT
R403        VP*1*7=COM*3                                       EXIT
R406        *B+VP*3=*-+1+COM*3                                 EXIT
*           VP*3=*-+COM*3                                      EXIT
R407        VP*4=COM*1                                         EXIT
WNNT        WN=NT                                              EXIT
WNO         WN=0                                               EXIT
WNPNT       WNP=NT                                             EXIT
WNPO        WNP=0                                              EXIT
WONT        WO=NT                                              EXIT
WOO         WO=0                                               EXIT
WOPN        WOP=NT                                             EXIT
WOPO        WOP=0                                              EXIT
R427        Z*1=F*1                                            EXIT
R428        Z*2=F*2                                            EXIT
```

```
ZZ2          $                                         //*Q2 1,*A6 1          ZZ1
FUNC         $+NV                                      //*Q2 1                *
ZY1          $+NV                                      //*Q5 1 2              ZY1
*            $                                         //*X5                  *
ZY2          *B+$+*C=*D+2+*E                                                  ZY2
*            $                                         //*X2                  *
ZZ1          $+*B+$+*B                                 //*Q6 1 2 3            ZZ1
*            $+*B                                      //*Q6 1                *
*            *B+$+*C=*Z+*D+2+*E                        //*A6 1                ZZ1
*            $=1+*A                                    //*A2 2                *
*            *B+$+*C=*D+2+*E                                                  ZZ2
*            $                                         //*X5                  *
ZZ5          $+*C                                      //*Q2 1 2,*X2          *
*            *B+$+*C=*D+2+*E+*A                        //*A2 4                ZZ5
*            $=*Z+1                                    //*A5 1                *
*            $1+$+*B+$+*C=1+2+*D+4+*E                  //*Q5 1 2 3 4 5,*A2 1  ZZ5
*            $=*A+*M+1+*N+*A                           //*A6 1,*A2 5          *
DOF          *D+$+*E=*B+2+*C                                                  DOF
OUT     A    *M+$+*N=*B+*(+Q+NV+*)+*B+*B+NPLUR+*(+NV+*)+*C+QUAD+2+-
              *C+*C                                                           BACH
        B                =*B+*B+PA*1+*(+PC+*)+*C+CNJ+2+*C                     XOB
        C                =*B+*B+PA*2+*(+V+*)+*C+CNJ+2+*C                      XOC
        D                =*B+*(+Q+NV+*)+*B+*B+N       +*(+NV+*)+*C+QUAD+2+-
              *C+*C                                                           BACH
        E                =*B+*B+SA+*(+SC+*)+*C+CNJ+2+*C                       XOE
        F                =*B+2+CNJ+*B+NO+*(+V*1+*)+*C+*C                      XOF
        G                =*B+2+CNJ+*B+NPLUR+*(+NV+*)+*C+*C                    XOG
        H                =*B+2+CNJ+*B+NPLUR+*(+NV+*)+*C+1+2+3+*C              XOH
        I                =*B+2+CNJ+*B+NPLUR+*(+NV+*)+*C+1+2+3+*C              XOI
XOH          *M+$+NV+$+*N=1+2+V*5+4+5                                         XOH
*            *M+$+*N=2                                                        XOG
XOI          *M+$+NV+$+*N=1+2+V*2+4+5                                         XOI
*            *M+$+*N=2                                                        XOG
STOP         *B+$+*C=*(+2+*)                                                  STOP
*            $                                         //*WSL 1               *
*            $=*A+*A+*A+*A+*A+*A                       //*A1 1,-
*A2 2,*A3 3,*A4 4,*A5 5,*A6 6                                                 *
*            $=0                                                              STAR
FINISH                                                                        *
     END
```

REFERENCES

1. Sillars, Walter. An Algorithm for Representing English Sentences in a Formal Language. NBS Report 7884.

2. Cohen, Donald. Picture Processing in a Picture Language Machine. NBS Report 7885.

3. Cohen, Donald. A Recognition Algorithm for a Grammar Model. NBS Report 7883.

4. Rankin, B. K., III. A Programmable Grammar for a Fragment of English for Use in an Information Retrieval System. NBS Report 7352.

DEPARTMENT OF COMMERCE ★ NATIONAL BUREAU OF STANDARDS ★