

## GRAMMAR ANALYSIS -- STANFORD UNIVERSITY

TODAY IS AUGUST 7, 1969.

## P R O D U C T I O N S

\$ THIS IS A TEST GRAMMAR WHICH MAY BE USED IN SKELETON  
 \$ITERATE  
 \$OUTPUT

```

1  <PROGRAM> ::= <STATEMENT LIST>
2  <STATEMENT LIST> ::= <STATEMENT>
3                        | <STATEMENT LIST> <STATEMENT>
4  <STATEMENT> ::= <ASSIGNMENT> ;
5  <ASSIGNMENT> ::= <VARIABLE> = <EXPR>
6  <EXPR> ::= <ARITH EXPR>
7           | <IF CLAUSE> THEN <EXPR> ELSE <EXPR>
8  <IF CLAUSE> ::= IF <BOOLEAN>
9  <BOOLEAN> ::= TRUE
10             | FALSE
11             | <EXPR> <RELATION> <EXPR>
12             | <IF CLAUSE> THEN <BOOLEAN> ELSE <BOOLEAN>
13 <RELATION> ::= =
14             | <
15             | >
16 <ARITH EXPR> ::= <TERM>
17             | <ARITH EXPR> + <TERM>
18             | <ARITH EXPR> - <TERM>
19 <TERM> ::= <PRIMARY>
20         | <TERM> * <PRIMARY>
21         | <TERM> / <PRIMARY>
22 <PRIMARY> ::= <VARIABLE>
23           | <NUMBER>
24           | ( <EXPR> )
25 <VARIABLE> ::= <IDENTIFIER>
26             | <VARIABLE> ( <EXPR> )

```

Fig. 7.1.6

## T E R M I N A L   S Y M B O L S

1 ;  
 2 =  
 3 <  
 4 >  
 5 +  
 6 -  
 7 \*  
 8 /  
 9 (  
 10 )  
 11 IF  
 12 \_ | \_  
 13 THEN  
 14 ELSE  
 15 TRUE  
 16 FALSE  
 17 <NUMBER>  
 18 <IDENTIFIER>

## N O N T E R M I N A L S

19 <EXPR>  
 20 <TERM>  
 21 <PROGRAM>  
 22 <BOOLEAN>  
 23 <PRIMARY>  
 24 <VARIABLE>  
 25 <RELATION>  
 26 <STATEMENT>  
 27 <IF CLAUSE>  
 28 <ASSIGNMENT>  
 29 <ARITH EXPR>  
 30 <STATEMENT LIST>

<PROGRAM> IS THE GOAL SYMBOL.

Fig. 7.1.6 (Contd.)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
																	<NUMBER>													
																	<IDENTIFIER>													
																	<EXPR>													
																	<TERM>													
																	<PROGRAM>													
																	<BOOLEAN>													
																	<PRIMARY>													
																	<VARIABLE>													
																	<RELATION>													
																	<STATEMENT>													
																	<IF CLAUSE>													
																	<ASSIGNMENT>													
																	<ARITH EXPR>													
																	<STATEMENT LIST>													

SENTENTIAL FORM PRODUCTION:

F11 HAS 321 ELEMENTS.  
 THE MAXIMUM DEPTH OF RECURSION WAS 22 LEVELS.  
 894 SENTENTIAL FORMS WERE EXAMINED.

Fig. 7.1.6 (Contd.)

## C1 MATRIX FOR STACKING DECISION: PAGE 1 OF 1

		1111111 11												
		1234567890123456 78												
		+-----+-----+												
1	;											N		N
2	=											#	#	##
3	<											N	N	NN
4	>											N	N	NN
5	+											Y		YY
6	-											Y		YY
7	*											Y		YY
8	/											Y		YY
9	(											Y	Y	YY
10	)													
11	IF											NNNNNNNNNN	NN	
12	_   _											Y	Y	YY
13	THEN											Y	Y	YY
14	ELSE											Y	Y	YY
15	TRUE													NN
16	FALSE													NN
		+-----+-----+												
17	<NUMBER>											NNNNNNNN	N	NN
18	<IDENTIFIER>											NNNNNNNNNN	NN	
19	<EXPR>											N###	#	N#
20	<TERM>											NNNNNNYY	N	NN
21	<PROGRAM>													N
22	<BOOLEAN>													N#
23	<PRIMARY>											NNNNNNNN	N	NN
24	<VARIABLE>											N#NNNNNNYN	NN	
25	<RELATION>											Y	Y	YY
26	<STATEMENT>													N
27	<IF CLAUSE>													Y
28	<ASSIGNMENT>											Y		
29	<ARITH EXPR>											NNNNYY	N	NN
30	<STATEMENT LIST>													N
		+-----+-----+												

## TABLE ENTRIES SUMMARY:

389

47 Y

93 N

11 #

Fig. 7.1.6 (Contd.)

## C1 TRIPLES FOR STACKING DECISION:

```

1  N FOR + <VARIABLE> =
2  N FOR - <VARIABLE> =
3  N FOR * <VARIABLE> =
4  N FOR / <VARIABLE> =
5  Y FOR ( <EXPR> )
6  Y FOR IF <EXPR> =
7  Y FOR IF <EXPR> <
8  Y FOR IF <EXPR> >
9  N FOR IF <VARIABLE> =
10 Y FOR _ _ <VARIABLE> =
11 Y FOR THEN <EXPR> =
12 Y FOR THEN <EXPR> <
13 Y FOR THEN <EXPR> >
14 Y FOR THEN <EXPR> ELSE
15 Y FOR THEN <BOOLEAN> ELSE
16 N FOR THEN <VARIABLE> =
*** ERROR, STACKING DECISION CANNOT BE MADE WITH (2,1) CONTEXT:
17 # FOR ELSE <EXPR> =
*** ERROR, STACKING DECISION CANNOT BE MADE WITH (2,1) CONTEXT:
18 # FOR ELSE <EXPR> <
*** ERROR, STACKING DECISION CANNOT BE MADE WITH (2,1) CONTEXT:
19 # FOR ELSE <EXPR> >
20 N FOR ELSE <EXPR> )
21 N FOR ELSE <EXPR> ELSE
22 N FOR ELSE <BOOLEAN> ELSE
23 N FOR ELSE <VARIABLE> =
24 N FOR <EXPR> = (
25 N FOR <EXPR> = IF
26 N FOR <EXPR> = <NUMBER>
27 N FOR <EXPR> = <IDENTIFIER>
28 Y FOR <VARIABLE> = (
29 Y FOR <VARIABLE> = IF
30 Y FOR <VARIABLE> = <NUMBER>
31 Y FOR <VARIABLE> = <IDENTIFIER>
32 N FOR <RELATION> <EXPR> ELSE
33 Y FOR <STATEMENT LIST> <VARIABLE> =

```

259 ENTRIES FOR 33 TRIPLES.

## TABLE ENTRIES SUMMARY:

```

15  Y
15  N
3   #

```

Fig. 7.1.6 (Contd.)

## ANALYSIS OF (2,1) CONFLICTS:

THE TRIPLE ELSE <EXPR> = MUST HAVE THE VALUE N FOR

```

7  <EXPR> ::= <IF CLAUSE> THEN <EXPR> ELSE <EXPR>
   IN THE CONTEXT IF ... =
   IN THE CONTEXT THEN ... =
   IN THE CONTEXT ELSE ... =

```

THE TRIPLE ELSE <EXPR> = MUST HAVE THE VALUE Y FOR

```

11 <BOOLEAN> ::= <EXPR> <RELATION> <EXPR>
    IN THE CONTEXT ELSE ... THEN
    IN THE CONTEXT ELSE ... ELSE

```

THE TRIPLE ELSE <EXPR> < MUST HAVE THE VALUE N FOR

```

7  <EXPR> ::= <IF CLAUSE> THEN <EXPR> ELSE <EXPR>
   IN THE CONTEXT IF ... <
   IN THE CONTEXT THEN ... <
   IN THE CONTEXT ELSE ... <

```

THE TRIPLE ELSE <EXPR> < MUST HAVE THE VALUE Y FOR

```

11 <BOOLEAN> ::= <EXPR> <RELATION> <EXPR>
    IN THE CONTEXT ELSE ... THEN
    IN THE CONTEXT ELSE ... ELSE

```

THE TRIPLE ELSE <EXPR> > MUST HAVE THE VALUE N FOR

```

7  <EXPR> ::= <IF CLAUSE> THEN <EXPR> ELSE <EXPR>
   IN THE CONTEXT IF ... >
   IN THE CONTEXT THEN ... >
   IN THE CONTEXT ELSE ... >

```

THE TRIPLE ELSE <EXPR> > MUST HAVE THE VALUE Y FOR

```

11 <BOOLEAN> ::= <EXPR> <RELATION> <EXPR>
    IN THE CONTEXT ELSE ... THEN
    IN THE CONTEXT ELSE ... ELSE

```

Fig. 7.1.6 (Contd.)

## CONTEXT CHECK FOR EQUAL AND EMBEDDED RIGHT PARTS:

THERE ARE 100 AND 86 VALID CONTEXTS, RESPECTIVELY, FOR  
 26 <VARIABLE> ::= <VARIABLE> ( <EXPR> )  
 24 <PRIMARY> ::= ( <EXPR> )  
 THEY CAN BE RESOLVED BY LENGTH.

THERE ARE 30 AND 30 VALID CONTEXTS, RESPECTIVELY, FOR  
 17 <ARITH EXPR> ::= <ARITH EXPR> + <TERM>  
 16 <ARITH EXPR> ::= <TERM>  
 THEY CAN BE RESOLVED BY LENGTH.

THERE ARE 30 AND 30 VALID CONTEXTS, RESPECTIVELY, FOR  
 18 <ARITH EXPR> ::= <ARITH EXPR> - <TERM>  
 16 <ARITH EXPR> ::= <TERM>  
 THEY CAN BE RESOLVED BY LENGTH.

THERE ARE 64 AND 64 VALID CONTEXTS, RESPECTIVELY, FOR  
 20 <TERM> ::= <TERM> \* <PRIMARY>  
 19 <TERM> ::= <PRIMARY>  
 THEY CAN BE RESOLVED BY LENGTH.

THERE ARE 64 AND 64 VALID CONTEXTS, RESPECTIVELY, FOR  
 21 <TERM> ::= <TERM> / <PRIMARY>  
 19 <TERM> ::= <PRIMARY>  
 THEY CAN BE RESOLVED BY LENGTH.

THERE ARE 2 AND 2 VALID CONTEXTS, RESPECTIVELY, FOR  
 3 <STATEMENT LIST> ::= <STATEMENT LIST> <STATEMENT>  
 2 <STATEMENT LIST> ::= <STATEMENT>  
 THEY CAN BE RESOLVED BY LENGTH.

## C2 PRODUCTION CHOICE FUNCTION:

; AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
 IN THIS ORDER:

4 <STATEMENT> ::= <ASSIGNMENT> ;  
 THERE WILL BE NO CONTEXT CHECK.

= AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
 IN THIS ORDER:

13 <RELATION> ::= =  
 THERE WILL BE NO CONTEXT CHECK.

< AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
 IN THIS ORDER:

14 <RELATION> ::= <  
 THERE WILL BE NO CONTEXT CHECK.

> AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED

Fig. 7.1.6 (Contd.)

IN THIS ORDER:

- 15 <RELATION> ::= >  
THERE WILL BE NO CONTEXT CHECK.
- ) AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
IN THIS ORDER:
- 26 <VARIABLE> ::= <VARIABLE> ( <EXPR> )  
THERE WILL BE NO CONTEXT CHECK.
- 24 <PRIMARY> ::= ( <EXPR> )  
THERE WILL BE NO CONTEXT CHECK.
- TRUE AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
IN THIS ORDER:
- 9 <BOOLEAN> ::= TRUE  
THERE WILL BE NO CONTEXT CHECK.
- FALSE AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
IN THIS ORDER:
- 10 <BOOLEAN> ::= FALSE  
THERE WILL BE NO CONTEXT CHECK.
- <NUMBER> AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
IN THIS ORDER:
- 23 <PRIMARY> ::= <NUMBER>  
THERE WILL BE NO CONTEXT CHECK.
- <IDENTIFIER> AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
IN THIS ORDER:
- 25 <VARIABLE> ::= <IDENTIFIER>  
THERE WILL BE NO CONTEXT CHECK.
- <EXPR> AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
IN THIS ORDER:
- 7 <EXPR> ::= <IF CLAUSE> THEN <EXPR> ELSE <EXPR>  
THERE WILL BE NO CONTEXT CHECK.
- 5 <ASSIGNMENT> ::= <VARIABLE> = <EXPR>  
THERE WILL BE NO CONTEXT CHECK.
- 11 <BOOLEAN> ::= <EXPR> <RELATION> <EXPR>  
THERE WILL BE NO CONTEXT CHECK.
- <TERM> AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
IN THIS ORDER:
- 17 <ARITH EXPR> ::= <ARITH EXPR> + <TERM>

Fig. 7.1.6 (Contd.)



- THERE WILL BE NO CONTEXT CHECK.
- 18  $\langle \text{ARITH EXPR} \rangle ::= \langle \text{ARITH EXPR} \rangle - \langle \text{TERM} \rangle$   
THERE WILL BE NO CONTEXT CHECK.
- 16  $\langle \text{ARITH EXPR} \rangle ::= \langle \text{TERM} \rangle$   
THERE WILL BE NO CONTEXT CHECK.
- $\langle \text{BOOLEAN} \rangle$  AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
IN THIS ORDER:
- 12  $\langle \text{BOOLEAN} \rangle ::= \langle \text{IF CLAUSE} \rangle \text{ THEN } \langle \text{BOOLEAN} \rangle \text{ ELSE } \langle \text{BOOLEAN} \rangle$   
THERE WILL BE NO CONTEXT CHECK.
- 8  $\langle \text{IF CLAUSE} \rangle ::= \text{IF } \langle \text{BOOLEAN} \rangle$   
THERE WILL BE NO CONTEXT CHECK.
- $\langle \text{PRIMARY} \rangle$  AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
IN THIS ORDER:
- 20  $\langle \text{TERM} \rangle ::= \langle \text{TERM} \rangle * \langle \text{PRIMARY} \rangle$   
THERE WILL BE NO CONTEXT CHECK.
- 21  $\langle \text{TERM} \rangle ::= \langle \text{TERM} \rangle / \langle \text{PRIMARY} \rangle$   
THERE WILL BE NO CONTEXT CHECK.
- 19  $\langle \text{TERM} \rangle ::= \langle \text{PRIMARY} \rangle$   
THERE WILL BE NO CONTEXT CHECK.
- $\langle \text{VARIABLE} \rangle$  AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
IN THIS ORDER:
- 22  $\langle \text{PRIMARY} \rangle ::= \langle \text{VARIABLE} \rangle$   
THERE WILL BE NO CONTEXT CHECK.
- $\langle \text{STATEMENT} \rangle$  AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
IN THIS ORDER:
- 3  $\langle \text{STATEMENT LIST} \rangle ::= \langle \text{STATEMENT LIST} \rangle \langle \text{STATEMENT} \rangle$   
THERE WILL BE NO CONTEXT CHECK.
- 2  $\langle \text{STATEMENT LIST} \rangle ::= \langle \text{STATEMENT} \rangle$   
THERE WILL BE NO CONTEXT CHECK.
- $\langle \text{ARITH EXPR} \rangle$  AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
IN THIS ORDER:
- 6  $\langle \text{EXPR} \rangle ::= \langle \text{ARITH EXPR} \rangle$   
THERE WILL BE NO CONTEXT CHECK.
- $\langle \text{STATEMENT LIST} \rangle$  AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
IN THIS ORDER:
- 1  $\langle \text{PROGRAM} \rangle ::= \langle \text{STATEMENT LIST} \rangle$

Fig. 7.1.6 (Contd.)

THERE WILL BE NO CONTEXT CHECK.

ANALYSIS COMPLETE FOR ITERATION 1

\*\*\* 3 ERRORS WERE DETECTED.

GRAMMAR MODIFICATION TO ATTEMPT TO RESOLVE CONFLICTS:

- 27 <ELSE1> ::= ELSE
- 7 <EXPR> ::= <IF CLAUSE> THEN <EXPR> <ELSE1> <EXPR>
- 28 <ELSE2> ::= ELSE
- 12 <BOOLEAN> ::= <IF CLAUSE> THEN <BOOLEAN> <ELSE2> <BOOLEAN>

Fig. 7.1.6 (Contd.)



## C1 MATRIX FOR STACKING DECISION: PAGE 1 OF 1

		1111111 11		
		1234567890123456 78		
1	;		N	N
2	=	#	#	##
3	<	N	N	NN
4	>	N	N	NN
5	+	Y		YY
6	-	Y		YY
7	*	Y		YY
8	/	Y		YY
9	(	Y	Y	YY
10	)	NNNNNNNNNN	NN	
11	IF	Y	Y	YY YY
12	_			Y
13	THEN	Y	Y	YY YY
14	ELSE	N	N	NN NN
15	TRUE		NN	
16	FALSE		NN	
17	<NUMBER>	NNNNNNNN	N	NN
18	<IDENTIFIER>	NNNNNNNNNN	NN	
19	<EXPR>	N###	#	N#
20	<TERM>	NNNNNNYY	N	NN
21	<PROGRAM>		N	
22	<BOOLEAN>		N#	
23	<PRIMARY>	NNNNNNNN	N	NN
24	<VARIABLE>	N#NNNNNNY	N	NN
25	<RELATION>		Y	Y
26	<STATEMENT>		N	N
27	<IF CLAUSE>		Y	
28	<ASSIGNMENT>	Y		
29	<ARITH EXPR>	NNNNYY	N	NN
30	<STATEMENT LIST>		N	Y
31	<ELSE1>		Y	Y
32	<ELSE2>		Y	Y

## TABLE ENTRIES SUMMARY:

4 15

51 Y  
99 N  
11 #

Fig. 7.1.6 (Contd.)

## C1 TRIPLES FOR STACKING DECISION:

```

1  N FOR + <VARIABLE> =
2  N FOR - <VARIABLE> =
3  N FOR * <VARIABLE> =
4  N FOR / <VARIABLE> =
5  Y FOR ( <EXPR> )
6  Y FOR IF <EXPR> =
7  Y FOR IF <EXPR> <
8  Y FOR IF <EXPR> >
9  N FOR IF <VARIABLE> =
10 Y FOR _|_ <VARIABLE> =
11 Y FOR THEN <EXPR> =
12 Y FOR THEN <EXPR> <
13 Y FOR THEN <EXPR> >
14 Y FOR THEN <EXPR> ELSE
15 Y FOR THEN <BOOLEAN> ELSE
16 N FOR THEN <VARIABLE> =
17 N FOR <EXPR> = (
18 N FOR <EXPR> = IF
19 N FOR <EXPR> = <NUMBER>
20 N FOR <EXPR> = <IDENTIFIER>
21 Y FOR <VARIABLE> = (
22 Y FOR <VARIABLE> = IF
23 Y FOR <VARIABLE> = <NUMBER>
24 Y FOR <VARIABLE> = <IDENTIFIER>
25 N FOR <RELATION> <EXPR> ELSE
26 Y FOR <STATEMENT LIST> <VARIABLE> =
27 N FOR <ELSE1> <EXPR> =
28 N FOR <ELSE1> <EXPR> <
29 N FOR <ELSE1> <EXPR> >
30 N FOR <ELSE1> <EXPR> )
31 N FOR <ELSE1> <EXPR> ELSE
32 N FOR <ELSE1> <VARIABLE> =
33 Y FOR <ELSE2> <EXPR> =
34 Y FOR <ELSE2> <EXPR> <
35 Y FOR <ELSE2> <EXPR> >
36 N FOR <ELSE2> <BOOLEAN> ELSE
37 N FOR <ELSE2> <VARIABLE> =

```

281 ENTRIES FOR 37 TRIPLES.

## TABLE ENTRIES SUMMARY:

```

18  Y
19  N
0   #

```

Fig. 7.1.6 (Contd.)

## CONTEXT CHECK FOR EQUAL AND EMBEDDED RIGHT PARTS:

THERE ARE 108 AND 93 VALID CONTEXTS, RESPECTIVELY, FOR  
 26 <VARIABLE> ::= <VARIABLE> ( <EXPR> )  
 24 <PRIMARY> ::= ( <EXPR> )  
 THEY CAN BE RESOLVED BY LENGTH.

THERE ARE 4 AND 6 VALID CONTEXTS, RESPECTIVELY, FOR  
 27 <ELSE1> ::= ELSE  
 28 <ELSE2> ::= ELSE  
 THEY CAN BE RESOLVED BY (1,0) CONTEXT.

THERE ARE 35 AND 35 VALID CONTEXTS, RESPECTIVELY, FOR  
 17 <ARITH EXPR> ::= <ARITH EXPR> + <TERM>  
 16 <ARITH EXPR> ::= <TERM>  
 THEY CAN BE RESOLVED BY LENGTH.

THERE ARE 35 AND 35 VALID CONTEXTS, RESPECTIVELY, FOR  
 18 <ARITH EXPR> ::= <ARITH EXPR> - <TERM>  
 16 <ARITH EXPR> ::= <TERM>  
 THEY CAN BE RESOLVED BY LENGTH.

THERE ARE 71 AND 71 VALID CONTEXTS, RESPECTIVELY, FOR  
 20 <TERM> ::= <TERM> \* <PRIMARY>  
 19 <TERM> ::= <PRIMARY>  
 THEY CAN BE RESOLVED BY LENGTH.

THERE ARE 71 AND 71 VALID CONTEXTS, RESPECTIVELY, FOR  
 21 <TERM> ::= <TERM> / <PRIMARY>  
 19 <TERM> ::= <PRIMARY>  
 THEY CAN BE RESOLVED BY LENGTH.

THERE ARE 2 AND 2 VALID CONTEXTS, RESPECTIVELY, FOR  
 3 <STATEMENT LIST> ::= <STATEMENT LIST> <STATEMENT>  
 2 <STATEMENT LIST> ::= <STATEMENT>  
 THEY CAN BE RESOLVED BY LENGTH.

## C2 PRODUCTION CHOICE FUNCTION:

; AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
 IN THIS ORDER:

4 <STATEMENT> ::= <ASSIGNMENT> ;  
 THERE WILL BE NO CONTEXT CHECK.

= AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
 IN THIS ORDER:

13 <RELATION> ::= =  
 THERE WILL BE NO CONTEXT CHECK.

< AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
 IN THIS ORDER:

Fig. 7.1.6 (Contd.)

```

14  <RELATION> ::= <
      THERE WILL BE NO CONTEXT CHECK.

> AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED
  IN THIS ORDER:

15  <RELATION> ::= >
      THERE WILL BE NO CONTEXT CHECK.

) AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED
  IN THIS ORDER:

26  <VARIABLE> ::= <VARIABLE> ( <EXPR> )
      THERE WILL BE NO CONTEXT CHECK.

24  <PRIMARY> ::= ( <EXPR> )
      THERE WILL BE NO CONTEXT CHECK.

ELSE AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED
  IN THIS ORDER:

27  <ELSE1> ::= ELSE
      (1,0) CONTEXT WILL BE CHECKED.  LEGAL LEFT CONTEXT:
      <EXPR> ...

28  <ELSE2> ::= ELSE
      THERE WILL BE NO CONTEXT CHECK.

TRUE AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED
  IN THIS ORDER:

9   <BOOLEAN> ::= TRUE
      THERE WILL BE NO CONTEXT CHECK.

FALSE AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED
  IN THIS ORDER:

10  <BOOLEAN> ::= FALSE
      THERE WILL BE NO CONTEXT CHECK.

<NUMBER> AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED
  IN THIS ORDER:

23  <PRIMARY> ::= <NUMBER>
      THERE WILL BE NO CONTEXT CHECK.

<IDENTIFIER> AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED
  IN THIS ORDER:

25  <VARIABLE> ::= <IDENTIFIER>
      THERE WILL BE NO CONTEXT CHECK.

<EXPR> AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED

```

Fig. 7.1.6 (Contd.)

IN THIS ORDER:

7 <EXPR> ::= <IF CLAUSE> THEN <EXPR> <ELSE1> <EXPR>  
THERE WILL BE NO CONTEXT CHECK.

5 <ASSIGNMENT> ::= <VARIABLE> = <EXPR>  
THERE WILL BE NO CONTEXT CHECK.

11 <BOOLEAN> ::= <EXPR> <RELATION> <EXPR>  
THERE WILL BE NO CONTEXT CHECK.

<TERM> AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
IN THIS ORDER:

17 <ARITH EXPR> ::= <ARITH EXPR> + <TERM>  
THERE WILL BE NO CONTEXT CHECK.

18 <ARITH EXPR> ::= <ARITH EXPR> - <TERM>  
THERE WILL BE NO CONTEXT CHECK.

16 <ARITH EXPR> ::= <TERM>  
THERE WILL BE NO CONTEXT CHECK.

<BOOLEAN> AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
IN THIS ORDER:

12 <BOOLEAN> ::= <IF CLAUSE> THEN <BOOLEAN> <ELSE2> <BOOLEAN>  
THERE WILL BE NO CONTEXT CHECK.

8 <IF CLAUSE> ::= IF <BOOLEAN>  
THERE WILL BE NO CONTEXT CHECK.

<PRIMARY> AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
IN THIS ORDER:

20 <TERM> ::= <TERM> \* <PRIMARY>  
THERE WILL BE NO CONTEXT CHECK.

21 <TERM> ::= <TERM> / <PRIMARY>  
THERE WILL BE NO CONTEXT CHECK.

19 <TERM> ::= <PRIMARY>  
THERE WILL BE NO CONTEXT CHECK.

<VARIABLE> AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
IN THIS ORDER:

22 <PRIMARY> ::= <VARIABLE>  
THERE WILL BE NO CONTEXT CHECK.

<STATEMENT> AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
IN THIS ORDER:

3 <STATEMENT LIST> ::= <STATEMENT LIST> <STATEMENT>

Fig. 7.1.6 (Contd.)



IN THIS ORDER:

7 <EXPR> ::= <IF CLAUSE> THEN <EXPR> ELSE <EXPR>  
THERE WILL BE NO CONTEXT CHECK.

8 <ASSIGNMENT> ::= <VARIABLE> = <EXPR>  
THERE WILL BE NO CONTEXT CHECK.

9 <PROGRAM> ::= <EXPR> <EXPRESSION>  
THERE WILL BE NO CONTEXT CHECK.

10 <TERM> AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
IN THIS ORDER:

11 <ARITH EXPR> ::= <ARITH EXPR> + <ARITH EXPR>  
THERE WILL BE NO CONTEXT CHECK.

12 <ARITH EXPR> ::= <ARITH EXPR> \* <ARITH EXPR>  
THERE WILL BE NO CONTEXT CHECK.

13 <ARITH EXPR> ::= <ARITH EXPR> / <ARITH EXPR>  
THERE WILL BE NO CONTEXT CHECK.

14 <ARITH EXPR> ::= <ARITH EXPR> ^ <ARITH EXPR>  
THERE WILL BE NO CONTEXT CHECK.

2 <STATEMENT LIST> ::= <STATEMENT>  
THERE WILL BE NO CONTEXT CHECK.

<ARITH EXPR> AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
IN THIS ORDER:

6 <EXPR> ::= <ARITH EXPR>  
THERE WILL BE NO CONTEXT CHECK.

<STATEMENT LIST> AS STACK TOP WILL CAUSE PRODUCTIONS TO BE CHECKED  
IN THIS ORDER:

1 <PROGRAM> ::= <STATEMENT LIST>  
THERE WILL BE NO CONTEXT CHECK.

ANALYSIS COMPLETE FOR ITERATION 2  
NO ERRORS WERE DETECTED.

Fig. 7.1.6 (Contd.)

CARD OUTPUT:

```

DECLARE NSY LITERALLY '32', NT LITERALLY '18';
DECLARE V(NSY) CHARACTER INITIAL ( '<ERROR: TOKEN = 0>', ';', '=',
  '<', '>', '+', '-', '*', '/', '(', ')', 'IF', '_|_', 'THEN',
  'ELSE', 'TRUE', 'FALSE', '<NUMBER>', '<IDENTIFIER>', '<EXPR>',
  '<TERM>', '<PROGRAM>', '<BOOLEAN>', '<PRIMARY>', '<VARIABLE>',
  '<RELATION>', '<STATEMENT>', '<IF CLAUSE>', '<ASSIGNMENT>',
  '<ARITH EXPR>', '<STATEMENT LIST>', 'ELSE', 'ELSE');
DECLARE V_INDEX(12) BIT(8) INITIAL ( 1, 11, 12, 13, 16, 17, 17, 17,
  18, 18, 18, 18, 19);
DECLARE C1(NSY) BIT(38) INITIAL (
  "(2) 00000 00000 00000 0000",
  "(2) 00000 00000 00200 0002",
  "(2) 00000 00003 03000 0033",
  "(2) 00000 00002 02000 0022",
  "(2) 00000 00002 02000 0022",
  "(2) 00000 00001 00000 0011",
  "(2) 00000 00001 00000 0011",
  "(2) 00000 00001 00000 0011",
  "(2) 00000 00001 00000 0011",
  "(2) 00000 00001 01000 0011",
  "(2) 02222 22222 20022 0000",
  "(2) 00000 00001 01000 1111",
  "(2) 00000 00000 00000 0001",
  "(2) 00000 00001 01000 1111",
  "(2) 00000 00002 02000 2222",
  "(2) 00000 00000 00022 0000",
  "(2) 00000 00000 00022 0000",
  "(2) 02222 22220 20022 0000",
  "(2) 02222 22222 20022 0000",
  "(2) 02333 00000 30023 0000",
  "(2) 02222 22110 20022 0000",
  "(2) 00000 00000 00200 0000",
  "(2) 00000 00000 00023 0000",
  "(2) 02222 22220 20022 0000",
  "(2) 02322 22221 20022 0000",
  "(2) 00000 00001 01000 0011",
  "(2) 00000 00000 00200 0002",
  "(2) 00000 00000 00010 0000",
  "(2) 01000 00000 00000 0000",
  "(2) 02222 11000 20022 0000",
  "(2) 00000 00000 00200 0001",
  "(2) 00000 00001 01000 0011",
  "(2) 00000 00001 01000 1111");
DECLARE NC1TRIPLES LITERALLY '17';
DECLARE C1TRIPLES(NC1TRIPLES) FIXED INITIAL ( 594698, 725762, 725763,
  725764, 792578, 856834, 856835, 856836, 856846, 857614, 1573385,
  1573387, 1573393, 1573394, 1972226, 2102018, 2102019, 2102020);
DECLARE PRTB(28) FIXED INITIAL (0, 28, 0, 0, 0, 1575187, 2323, 0, 0,
  0, 0, 0, 0, 453841695, 6146, 4889, 7429, 7430, 0, 453842464, 11,
  5127, 5128, 0, 0, 30, 0, 0, 0);

```

Fig. 7.1.6 (Contd.)

```

DECLARE PRDTB(28) BIT(8) INITIAL ( 0, 4, 13, 14, 15, 26, 24, 0, 0, 9,
    10, 23, 25, 7, 5, 11, 17, 18, 16, 12, 8, 20, 21, 19, 22, 3, 2, 6,
    1);
DECLARE HDTB(28) BIT(8) INITIAL ( 0, 26, 25, 25, 25, 24, 23, 31, 32,
    22, 22, 23, 24, 19, 28, 22, 29, 29, 29, 22, 27, 20, 20, 20, 23, 30,
    30, L9, 21);
DECLARE PRLNGTH(28) BIT(8) INITIAL ( 0, 2, 1, 1, 1, 4, 3, 1, 1, 1, 1,
    1, 1, 5, 3, 3, 3, 3, 1, 5, 2, 3, 3, 1, 1, 2, 1, 1, 1);
DECLARE CONTEXT_CASE(28) BIT(8) INITIAL ( 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
DECLARE LEFT_CONTEXT(0) BIT(8) INITIAL ( 19);
DECLARE LEFT_INDEX(14) BIT(8) INITIAL ( 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 1, 1);
DECLARE CONTEXT_TRIPLE(0) FIXED INITIAL ( 0);
DECLARE TRIPLE_INDEX(14) BIT(8) INITIAL ( 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 1);
DECLARE PR_INDEX(32) BIT(8) INITIAL ( 1, 2, 3, 4, 5, 5, 5, 5, 5, 5, 7,
    7, 7, 7, 9, 10, 11, 12, 13, 16, 19, 19, 21, 24, 25, 25, 27, 27, 27,
    28, 29, 29, 29);

```

Fig. 7.1.6 (Contd.)