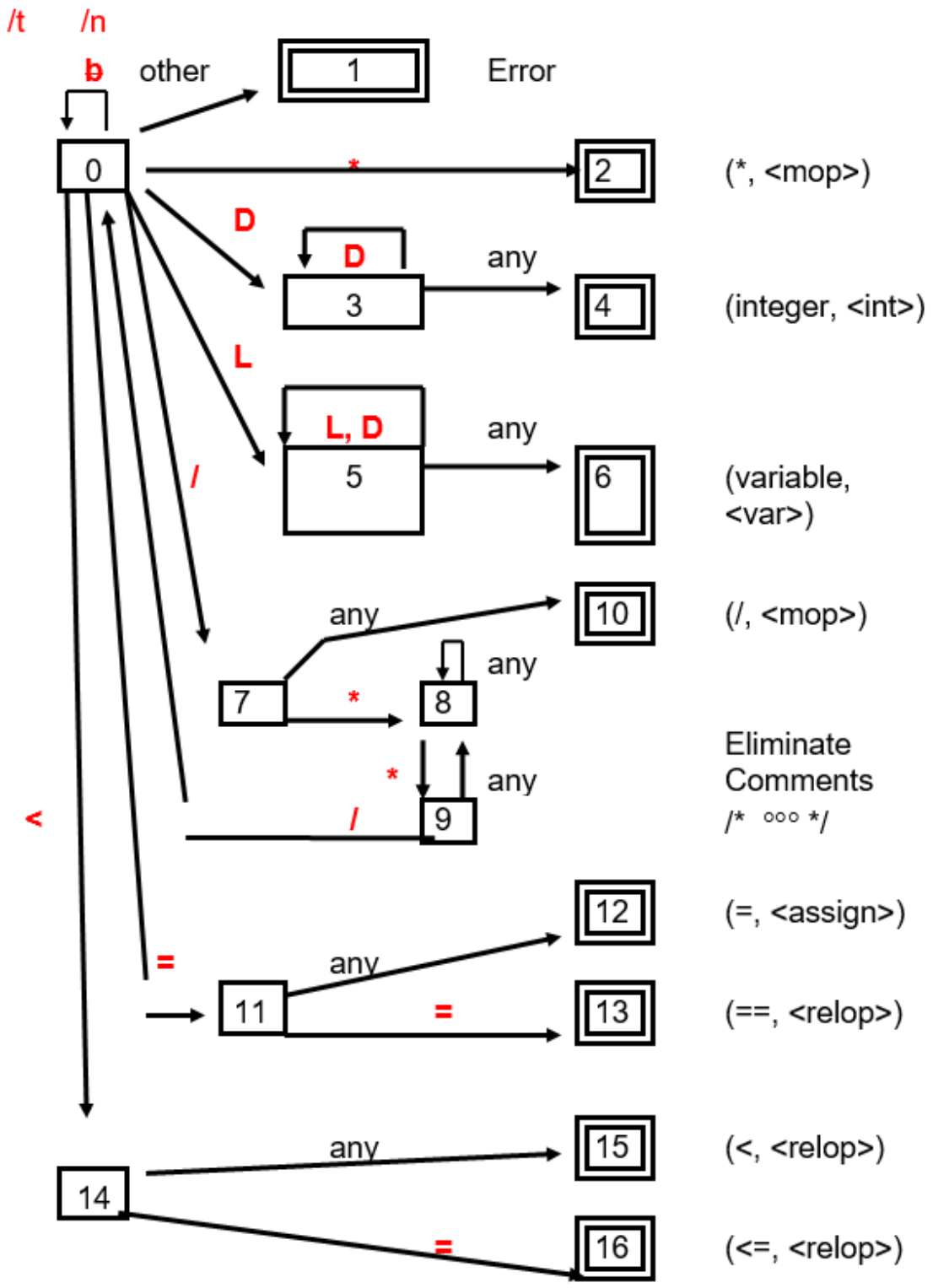
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	w	**D**	**L**	*****	**/**	**=**	**<**	**>**	**+**	**-**	**(**	**)**	**{**	**}**	**,**	**;**	**Other**
0	0	1	3	5	6	10	13	16	19	20	21	22	23	24	25	26	27
1	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	**Integer**, <int> → State 0																
3	4	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	**Variable**, <var> → State 0																
5	***, <mop> → State 0																
6	7	7	7	8	7	7	7	7	7	7	7	7	7	7	7	7	7
7	*/*, <mop> → State 0																
8	8	8	8	9	8	8	8	8	8	8	8	8	8	8	8	8	8
9	8	8	8	8	0	8	8	8	8	8	8	8	8	8	8	8	8
10	11	11	11	11	11	12	11	11	11	11	11	11	11	11	11	11	11
11	**=**, <assign> → State 0																
12	**==**, <relop> → State 0																
13	14	14	14	14	14	15	14	14	14	14	14	14	14	14	14	14	14
14	**<**, <relop> → State 0																
15	**<=**, <relop> → State 0																
16	17	17	17	17	17	18	17	17	17	17	17	17	17	17	17	17	17
17	**>**, <relop> → State 0																
18	**>=**, <relop> → State 0																
19	**+**, <addop> → State 0																
20	**-**, <addop> → State 0																
21	**(**, <parenthesisL> → State 0																
22	**)**, <parenthesisR> → State 0																
23	**{**, <braceL> → State 0																

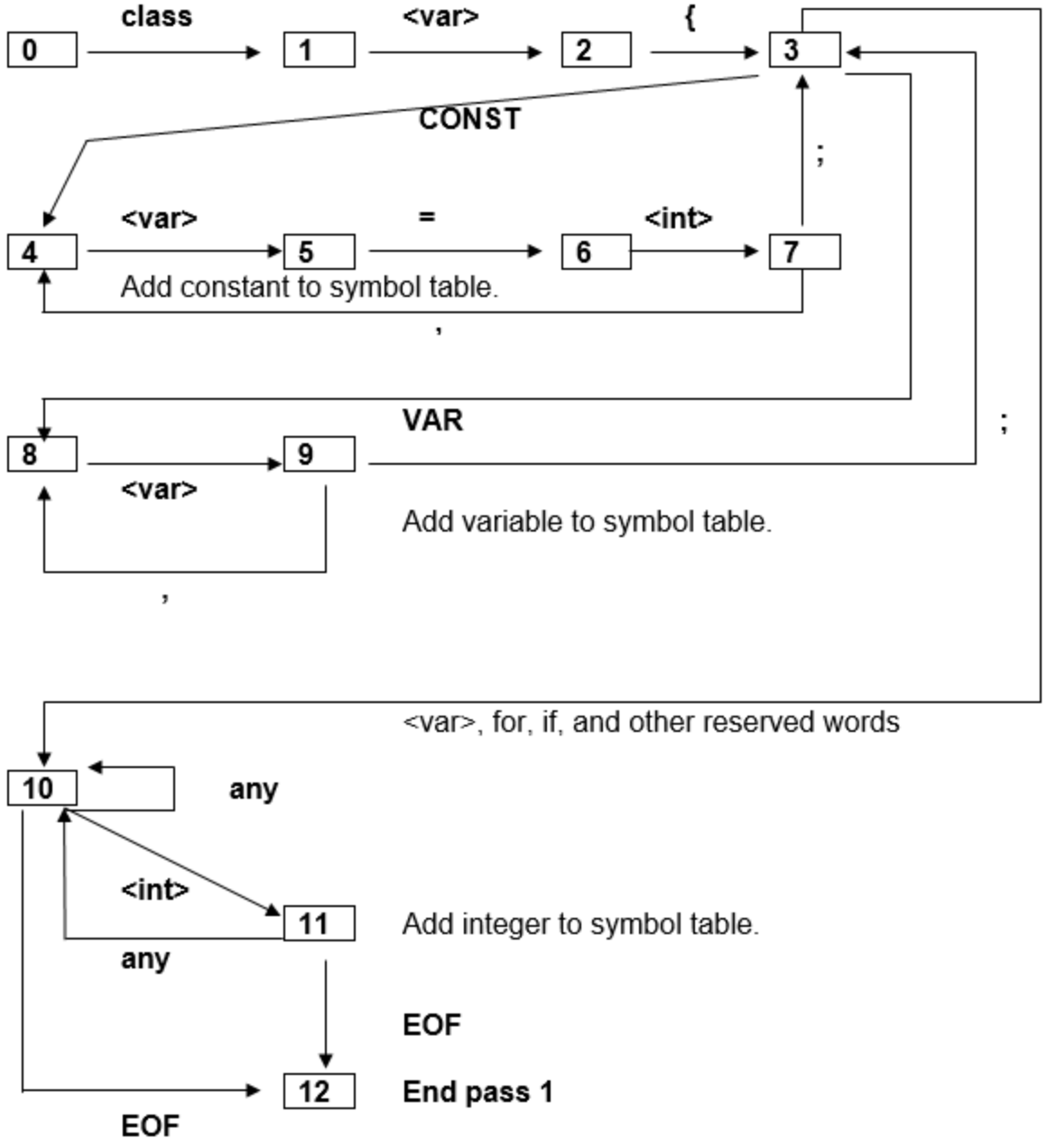
24	**}**, <braceR> → State 0
25	**,**, <comma> → State 0
26	**;**, <semicolon> → State 0
27	**Other** → State 0

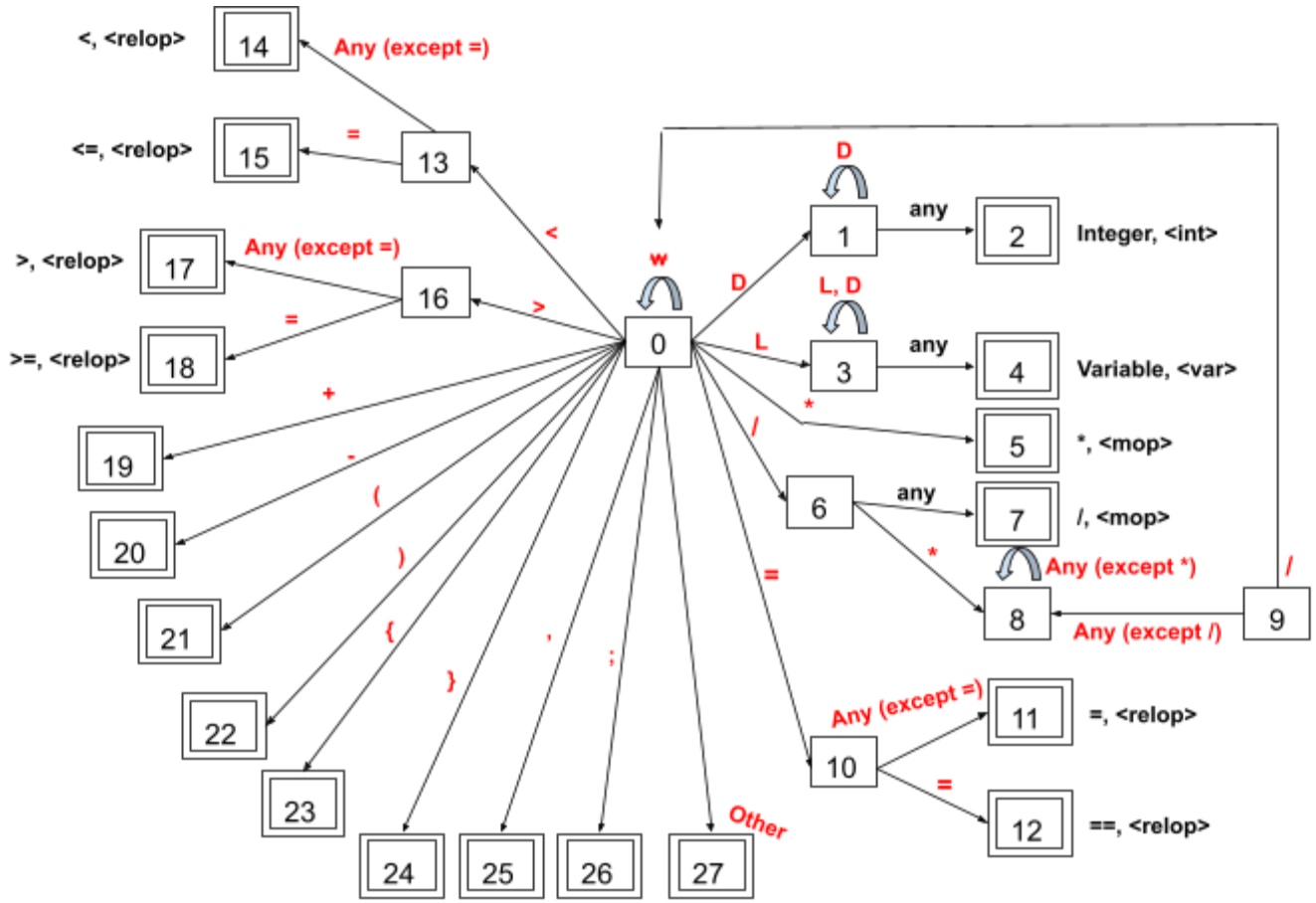
Table corresponding to Partial Heuristic Parser for C++

	L	D	*	/	=	<	ъ	other
0	5	3	1	7	11	14	0	1
1	Error, illegal character in input!							
2	*, <mop>							
3	4	3	4	4	4	4	4	
4	Integer, <int>							
5	5	5	6	6	6	6	6	
6	Variable, <var>							
7	10	10	8	10	10	10	10	
8	8	8	9	8	8	8	8	
9	8	8	8	0	8	8	8	
10	/, <mop>							
11	12	12	12	12	13	12	12	
12	=, <assignment>							
13	==, <relop>							
14	15	15	15	15	16	15	15	
15	<, <relop>							
16	<=, <relop>							

Partial heuristic parser for C++




class
<var>
{
0
1
2
3
CONST
;
<var>
=
<int>
4
5
6
7
Add constant to symbol table.
,
VAR
;
8
9
<var>
Add variable to symbol table.
,
<var>, for, if, and other reserved words
10
any
<int>
11
Add integer to symbol table.
any
EOF
12
End pass 1
EOF



		⊥	=	+	-	(	)	*	/	IF	THEN	ODD	==	!=	>	<	>=	<=	{	}	CALL
0	⊥		<																		
1	=	>		<	<	<		<	<												
2	+	>		>	>	<	>	<	<		>		>	>	>	>	>	>			
3	-	>		>	>	<	>	<	<		>		>	>	>	>	>	>			
4	(			<	<	<	=	<	<												
5	)	>		>	>		>	>	>												
6	*	>		>	>	<	>	>	>		>		>	>	>	>	>	>			
7	/	>		>	>	<	>	>	>		>		>	>	>	>	>	>			
8	IF			<	<	<		<	<		=	<	<	<	<	<	<	<			
9	THEN		<							<									<		<
	ODD			<	<	<		<	<		>										

10	**==**			<	<	<		<	<		>									
11	**!=**			<	<	<		<	<		>									
12	**>**			<	<	<		<	<		>									

Useful Links

- How to align text to an output file
-