input\state	else	"0"	"number"	"+"	"-"	"*"	"/"	"("	")"	" \0"
NUM	SYNTAX	VALID_NUM	VALID_NUM	VALID_UNARY	VALID_UNARY	SYNTAX	SYNTAX	LEFT_B_NUM	SYNTAX	SYNTAX
OP	SYNTAX	SYNTAX	SYNTAX	VALID_OPERATOR	VALID_OPERATOR	VALID_OPERATOR	Go2NonZ	VALID_LEFT_B	EVALUATE_B	EVALUATE_E
UNARY	SYNTAX	VALID_NUM	VALID_NUM	SYNTAX	SYNTAX	SYNTAX	SYNTAX	VALID_OPERATOR	SYNTAX	SYNTAX
NON_Z	SYNTAX	MATH	VALID_NUM	VALID_UNARY	VALID_UNARY	SYNTAX	SYNTAX	LEFT_B_NON_Z	SYNTAX	SYNTAX
TRAP										

#	event name	end state	actions
1	Go2NonZ	NON_Z	push to OP_STACK
2	VALID_OPERATOR	NUM	push to OP_STACK by presedency
3	VALID_NUM	OP	push to NUM_STACK
4	EVALUATE_E	OP	evaluate all stacks
5	EVALUATE_B	OP	evaluate brackets in stacks
6	VALID_LEFT_B	OP	push '(' to OP_STACK if * implicit make explicit
7	MATH	TRAP	end parsing, status = MATH_ERROR
8	SYNTAX	TRAP	end parsing, status = SYNTAX_ERROR
9	VALID_UNARY	UNARY	push '-' / '+' (with high presedency) to OP_STACK and '0' to NUM_STACK
10	LEFT_B_NUM	NUM	push "("to OP_STACK
11	LEFT_B_NON_Z	NUM	

SYNTAX