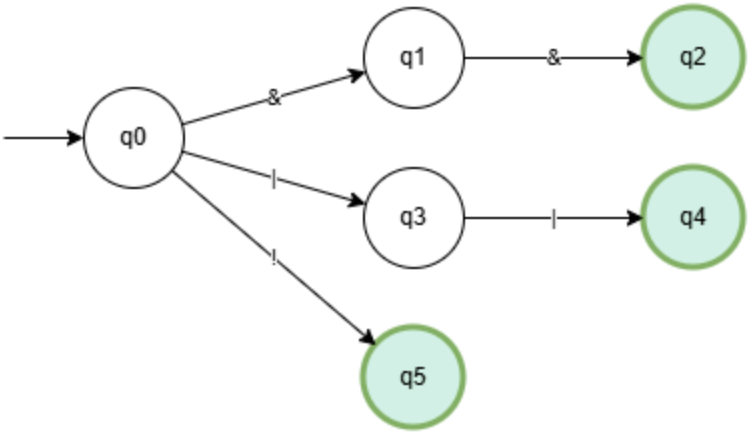
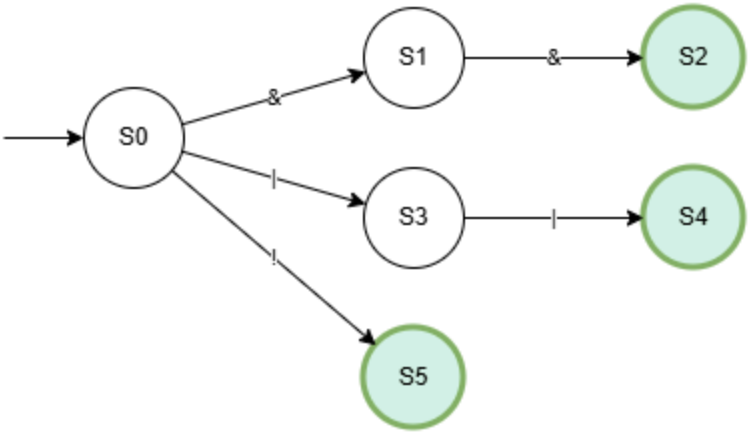
LOGICAL OPERATORS: (&&|\|\||!)

NFA (Nondeterministic Finite Automaton)



Legend: q0 = Start, q2 = Accept '&&', q4 = Accept '||', q5 = Accept '!' (Green states)

Minimized DFA (Deterministic Finite Automaton)



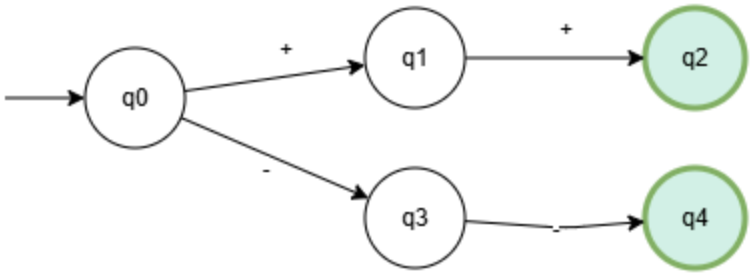
Legend: S0 = Start, S2 = Accept '&&', S4 = Accept '||', S5 = Accept '!'. DFA is same as NFA.

DFA Transition Table

State	&	|	!	Accept?
S0	S1	S3	S5	No
S1	S2	—	—	No
S2	—	—	—	**Yes (&&)**
S3	—	S4	—	No
S4	—	—	—	**Yes (||)**
S5	—	—	—	**Yes (!)**

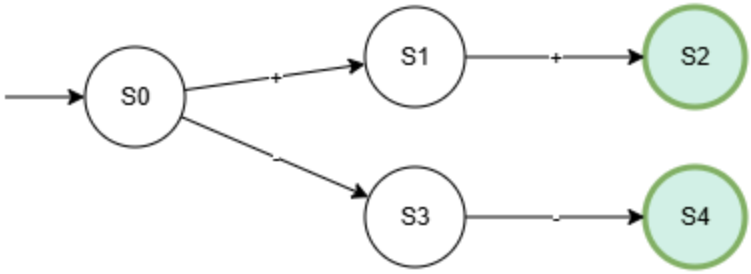
INCREMENT/DECREMENT OPERATORS: (\+\+|\-\-)

NFA (Nondeterministic Finite Automaton)



Legend: q0 = Start State, q2 = Accept '++', q4 = Accept '--' (Green states)

Minimized DFA (Deterministic Finite Automaton)



Legend: S0 = Start State, S2 = Accept '++', S4 = Accept '--'. DFA is same as NFA (already deterministic).

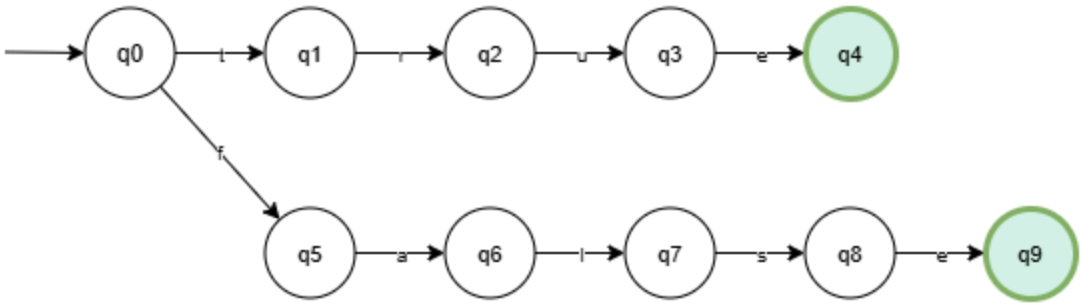
DFA Transition Table

State	+	-	Accept?
S0	S1	S3	No
S1	S2	—	No
S2	—	—	**Yes (++)**
S3	—	S4	No
S4	—	—	**Yes (--)**

Valid Examples: ++, --
Invalid Examples: +, -, +++, +-, -+

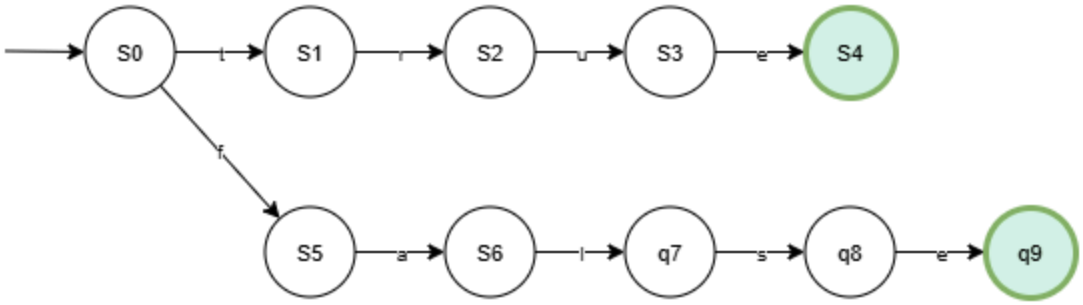
BOOLEAN LITERALS: (true|false)

NFA (Nondeterministic Finite Automaton)



Legend: q0 = Start, q4 = Accept 'true', q9 = Accept 'false' (Green states)

Minimized DFA (Deterministic Finite Automaton)



Legend: S0 = Start, S4 = Accept 'true', S8 = Accept 'false'. DFA is same as NFA (already deterministic).

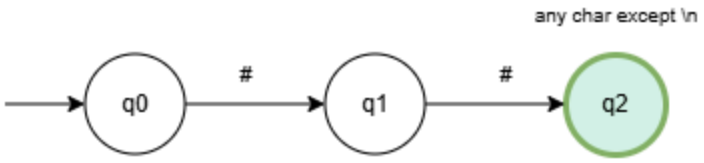
DFA Transition Table

State	t	r	u	e	f	a	l	s	Accept?
S0	S1	—	—	—	S5	—	—	—	No
S1	—	S2	—	—	—	—	—	—	No
S2	—	—	S3	—	—	—	—	—	No
S3	—	—	—	S4	—	—	—	—	No
S4	—	—	—	—	—	—	—	—	**Yes (true)**
S5	— / — / — / — / — / — / a→S6 / — / —								No
S6	— / — / — / — / — / — / — / l→S7 / —								No
S7	— / — / — / — / — / — / — / — / s→S8								No
S8	— / — / — / — / — / — / — / — / —								**Yes (false)**

Valid Examples: true, false
Invalid Examples: True, FALSE, tru, fals, truee

SINGLE-LINE COMMENT: ##[^\n]*

NFA (Nondeterministic Finite Automaton)



Legend: q0 = Start State, q2 = Accept State (Green). Accepts empty comment '##' and any content after.

Minimized DFA (Deterministic Finite Automaton)

Legend: S0 = Start State, S2 = Accept State (Green). The DFA is identical to NFA for this simple pattern.

DFA Transition Table

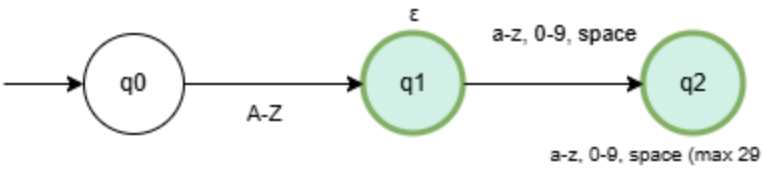
State	#	any (except \n)	Accept?
S0	S1	—	No
S1	S2	—	No
S2	S2	S2	**Yes**

Note: The comment terminates at end of line (\n), which is not consumed by the pattern.
Any character except newline can appear after ##.

Valid Examples: ##, ## this is a comment, ## 12345, ## special chars !@#$%
Invalid Examples: # (single hash), #comment (single hash), ## multi\nline (newline breaks it)

IDENTIFIER: [A-Z][a-z0-9]{0,30}

NFA (Nondeterministic Finite Automaton)



Legend: q0 = Start, q1 & q2 = Accept States (Green). Epsilon transition from q1 to itself represents 0 additional chars.

Minimized DFA (Deterministic Finite Automaton)

Legend: S0 = Start, S1 & S2 = Accept States. S1 accepts single uppercase letter, S2 accepts with suffix.

DFA Transition Table

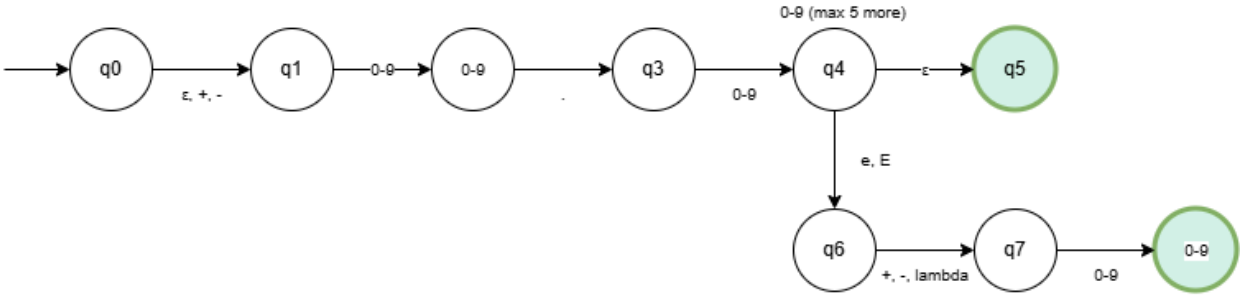
State	A-Z	a-z, 0-9, space	Accept?
S0	S1	—	No
S1	—	S2	**Yes**
S2	—	S2 (max 30)	**Yes**

Note: In practice, the DFA would need to track character count (0-30) to enforce the maximum length constraint.
This simplified version shows the structure but doesn't explicitly count characters.

Valid Examples: A, Variable, Max30characters123456789012, Test 123
Invalid Examples: abc (lowercase start), A12345678901234567890123456789012345 (too long)

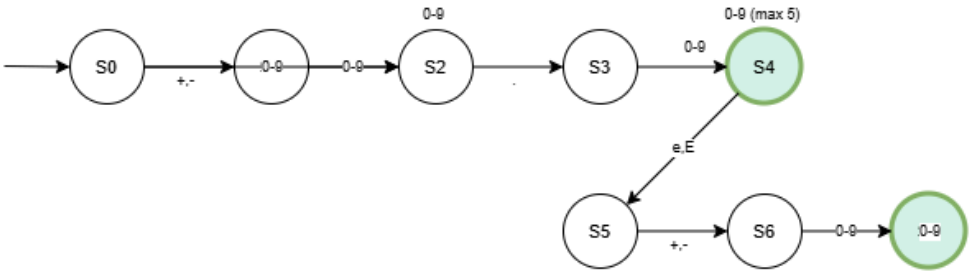
FLOATING-POINT LITERAL: [+-]?[0-9]+\.[0-9]{1,6}([eE][+-]?[0-9]+)?

NFA (Nondeterministic Finite Automaton)



Legend: q0 = Start, q5 & q8 = Accept States (Green). Fractional part limited to 6 digits max.

Minimized DFA (Deterministic Finite Automaton)



Legend: S0 = Start, S4 & S7 = Accept States. S4 accepts without exponent, S7 with exponent.

DFA Transition Table (Simplified)

State	+/-	0-9	.	e/E	Accept?
S0	S1	S2	—	—	No
S1	—	S2	—	—	No
S2	—	S2	S3	—	No
S3	—	S4	—	—	No
S4	—	S4	—	S5	**Yes**
S5	S6	—	—	—	No
S6	—	S7	—	—	No
S7	—	S7	—	—	**Yes**

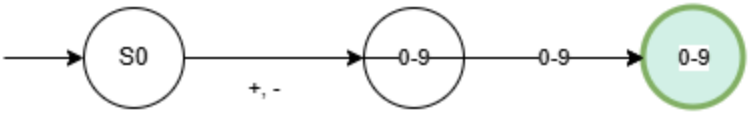
Valid Examples: 3.14, +2.71828, -0.5e-10, 123.456789 (only 6 decimal digits max)
Invalid Examples: 3, .5, 1.2345678 (too many decimals), 1.e5

INTEGER LITERAL: [+-]?[0-9]+

NFA (Nondeterministic Finite Automaton)

Legend: q0 = Start State, q2 = Accept State (Green)

Minimized DFA (Deterministic Finite Automaton)



Legend: S0 = Start State, S2 = Accept State (Green)

DFA Transition Table

State	+	-	0-9	Accept?
S0	S1	S1	S2	No
S1	—	—	S2	No
S2	—	—	S2	**Yes**

Valid Examples: 123, +456, -789, 0
Invalid Examples: ++1, 1.5, abc

COMBINED MINIMIZED DFA — TRANSITION TABLE

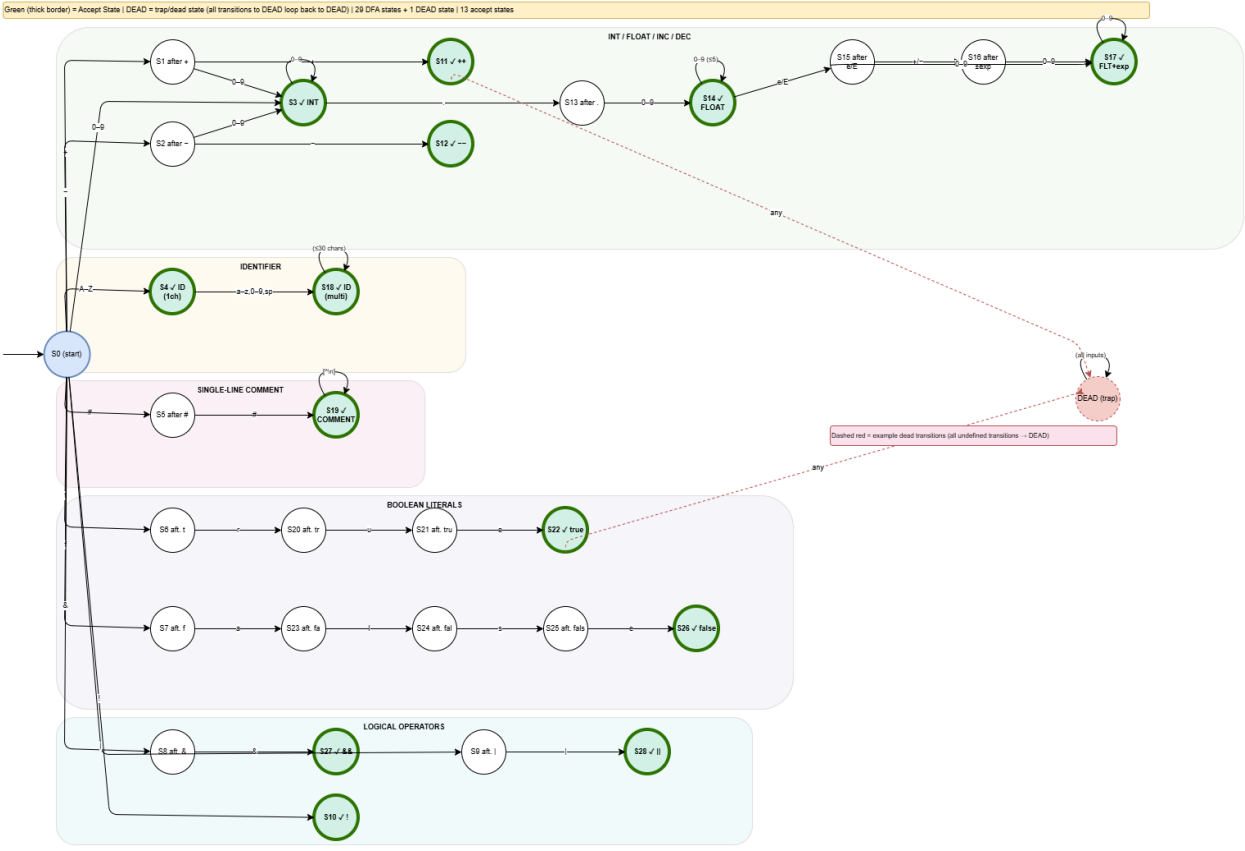
— = DEAD/trap state | ✓ = accept state | All accept states have no outgoing transitions except where shown

State	Recognizes	+	−	0–9	.	e/E	A–Z	a–z,sp	#	t	r	u	f	a	l	s	e	&	|	!	Accept?
S0	Start	S1	S2	S3	—	—	S4	—	S5	S6	—	—	S7	—	—	—	—	S8	S9	S10	—
S1	after +	S11	—	S3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
S2	after −	—	S12	S3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
S3✓	INT	—	—	S3	S13	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	**INT**
S4✓	ID (1-char)	—	—	S18	—	—	—	S18	—	—	—	—	—	—	—	—	—	—	—	—	**ID**
S5	after #	—	—	—	—	—	—	—	S19	—	—	—	—	—	—	—	—	—	—	—	—
S6	after t	—	—	—	—	—	—	—	—	—	S20	—	—	—	—	—	—	—	—	—	—
S7	after f	—	—	—	—	—	—	—	—	—	—	—	—	S23	—	—	—	—	—	—	—
S8	after &	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	S27	—	—
S9	after |	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	S28	—
S10✓	LOGICAL !	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	**!**
S11✓	INC ++	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	**++**
S12✓	DEC −−	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	**−−**
S13	after .	—	—	S14	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
S14✓	FLOAT	—	—	S14	—	S15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	**FLOAT**
S15	after e/E	S16	S16	S17	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
S16	after exps	—	—	S17	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
S17✓	FLOAT+exp	—	—	S17	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	**FLOAT+exp**
S18✓	ID (multi)	—	—	S18	—	—	—	S18	—	—	—	—	—	—	—	—	—	—	—	—	**ID**
S19✓	COMMENT	S19	S19	S19	S19	S19	S19	S19	S19	S19	S19	S19	S19	S19	S19	S19	S19	S19	S19	S19	**S19**
S20	after tr	—	—	—	—	—	—	—	—	—	—	—	S21	—	—	—	—	—	—	—	—
S21	after tru	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	S22	—	—	—
S22✓	true	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	**true**
S23	after fa	—	—	—	—	—	—	—	—	—	—	—	—	—	—	S24	—	—	—	—	—
S24	after fal	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	S25	—	—	—	—
S25	after fals	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	S26	—	—	—
S26✓	false	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	**false**
S27✓	LOGIC &&	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	**&&**
S28✓	LOGIC ||	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	**||**
DEAD	(trap)	DEAD	DEAD	DEAD	DEAD	DEAD	DEAD	DEAD	DEAD	DEAD	DEAD	DEAD	DEAD	DEAD	DEAD	DEAD	DEAD	DEAD	DEAD	DEAD	DEAD

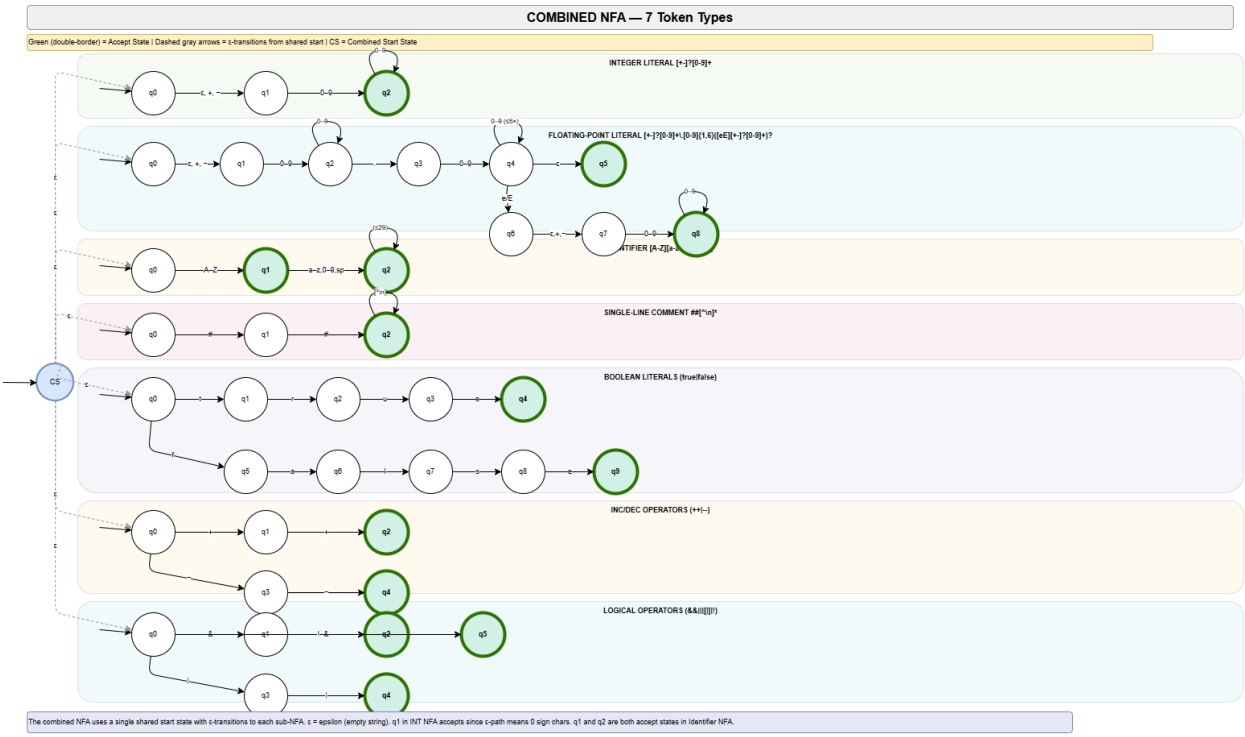
Total: 29 live states (S0–S28) + DEAD state = 30 states | 13 accept states | Input alphabet: {(+, −, 0–9, ., e/E, A–Z, a–z, space, #, t, r, u, f, a, l, s, e, &, |, !)}

COMBINED MINIMIZED DFA — 7 Token Types

Green (thick border) = Accept State | DEAD = trap/dead state (all transitions to DEAD loop back to DEAD) | 29 DFA states + 1 DEAD state | 13 accept states



Minimality: No two states are equivalent (distinguishable by unique continuations). 29 live states + 1 dead/trap state. Accept states: S3(INT), S4/S18(ID), S10(!), S11(++), S12(−−), S14(FLOAT), S17(FLOAT+exp), S19(CMT), S22(true), S26(false), S27(&&), S28(||).



DFA MINIMIZATION ALGORITHM

DFA minimization was performed using the Table-Filling (Marking) Algorithm. The process begins by constructing a triangular table for all state pairs (qi, qj). In the base case, any pair where exactly one state is an accept state is immediately marked as distinguishable. The algorithm then iterates over all unmarked pairs: for each input symbol a, if δ(qi, a) and δ(qj, a) lead to an already-distinguished pair, then (qi, qj) is also marked as distinguishable. This continues until no new pairs can be marked. Any pair that remains unmarked at termination represents two equivalent states that can be safely merged without changing the language accepted.

Applying this algorithm to the combined 7-token DFA (30 states: S0–S28 + DEAD), the base case immediately distinguished all 13 accept states from the 17 non-accept states. After full iteration, only one equivalent pair was found: S4 (identifier accepting a single uppercase letter) and S18 (identifier accepting multiple characters) are indistinguishable, as both accept the same token type and have identical transition functions going forward. These two states were merged into a single state S_ID, reducing the DFA from 30 to 29 states. All remaining state pairs were confirmed distinguishable, proving the resulting 29-state DFA is the unique minimal DFA for this language.