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

NFA (Nondeterministic Finite Automaton)

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

## Minimized DFA (Deterministic Finite Automaton)
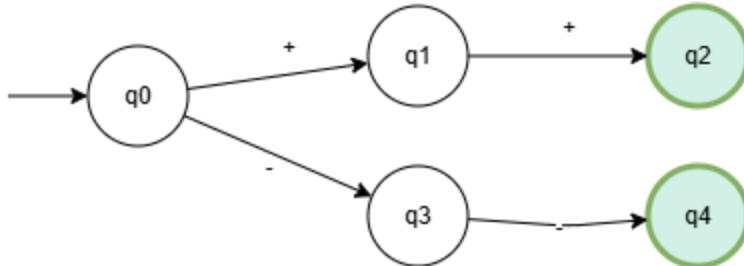
Legend: S0 = Start, S2 = Accept '&&', S4 = Accept '||', S5 = Accept '!'. DFA is same as NFA.

DFA Transition Table

| State	&	|	!	Accept?
S0	S1	S3	S5	No
S1	S2	—	—	No
S2	—	—	—	Yes (&&)
S3	—	S4	—	No
S4	—	—	—	Yes (||)
S5	—	—	—	Yes (!)
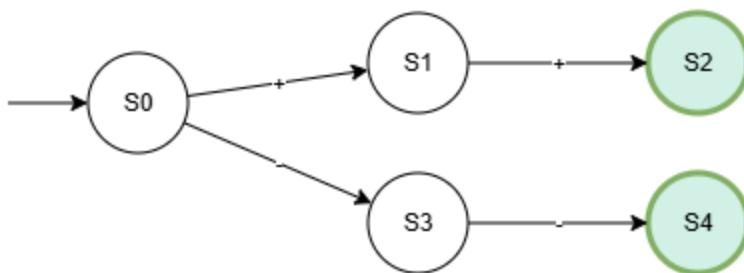
INCREMENT/DECREMENT OPERATORS: (\+\+|\-\-)

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).
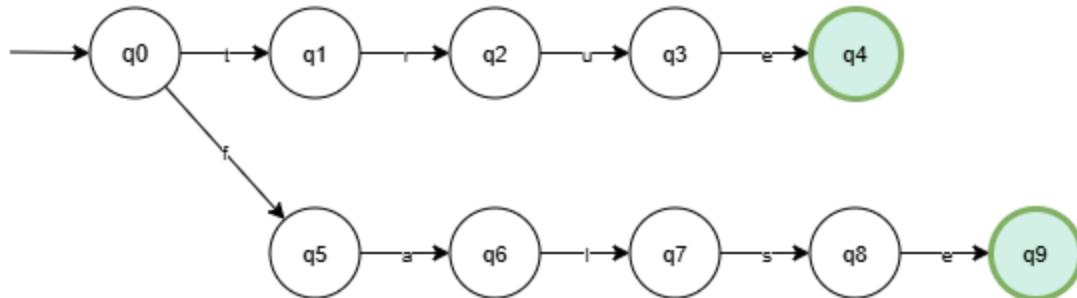
DFA Transition Table

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

Valid Examples: ++, --
Invalid Examples: +, -, +++, +-, -+
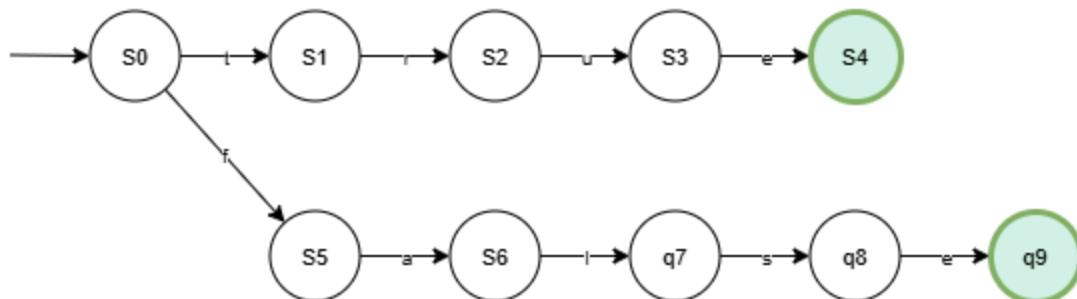
BOOLEAN LITERALS: (true|false)

NFA (Nondeterministic Finite Automaton)

```
q0 --t--> q1 --r--> q2 --u--> q3 --e--> q4
 |
 f
 v
q5 --a--> q6 --l--> q7 --s--> q8 --e--> q9
```

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

Minimized DFA (Deterministic Finite Automaton)

```
S0 --t--> S1 --r--> S2 --u--> S3 --e--> S4
 |
 f
 v
S5 --a--> S6 --l--> q7 --s--> q8 --e--> q9
```

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

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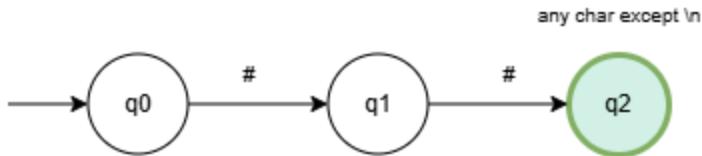
State	t	r	u	e	f	a	l	s	Accept?
S0	S1	—	—	—	S5	—	—	—	No
S1	—	S2	—	—	—	—	—	—	No
S2	—	—	S3	—	—	—	—	—	No
S3	—	—	—	S4	—	—	—	—	No
S4	—	—	—	—	—	—	—	—	Yes (true)
S5	— / — / — / — / — / a→S6 / — / —								No
S6	— / — / — / — / — / — / l→S7 / —								No
S7	— / — / — / — / — / — / — / s→S8								No
S8	— / — /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

Valid Examples: true, false
Invalid Examples: True, FALSE, tru, fals, truee
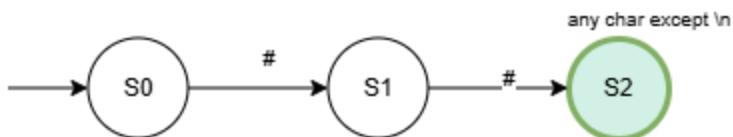
SINGLE-LINE COMMENT: ##[^\n]*

NFA (Nondeterministic Finite Automaton)

any char except \n

q0 → # → q1 → # → q2

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

Minimized DFA (Deterministic Finite Automaton)

any char except \n

S0 → # → S1 → # → S2

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

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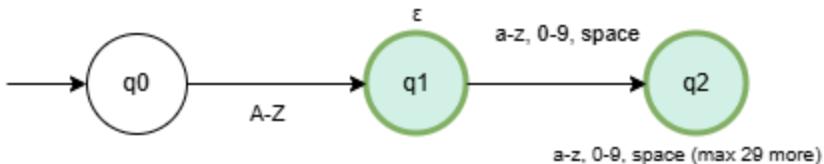
State	#	any (except \n)	Accept?
S0	S1	—	No
S1	S2	—	No
S2	S2	S2	Yes

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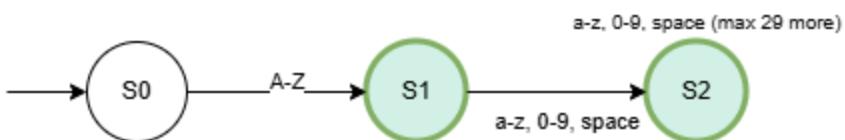
IDENTIFIER: [A-Z][a-z0-9]{0,30}

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.
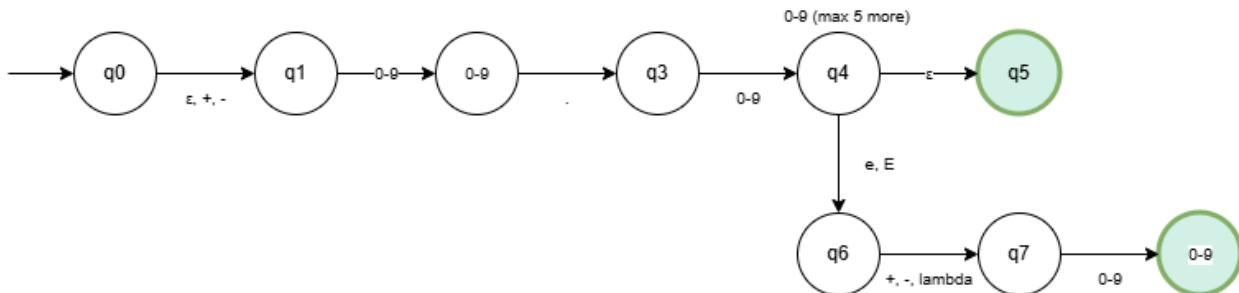
Valid Examples: A, Variable, Max30characters123456789012, Test 123
Invalid Examples: abc (lowercase start), A1234567890123456789012345678901234 (too long)
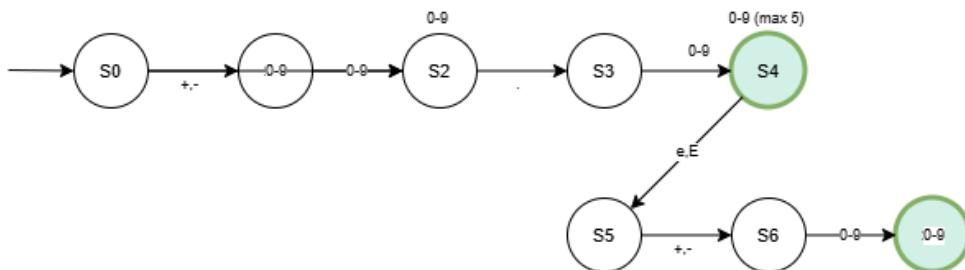
FLOATING-POINT LITERAL: [+-]?[0-9]+\.[0-9]{1,6}([eE][+-]?[0-9]+)?

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.
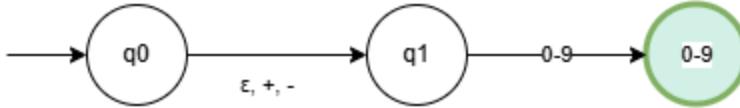
DFA Transition Table (Simplified)

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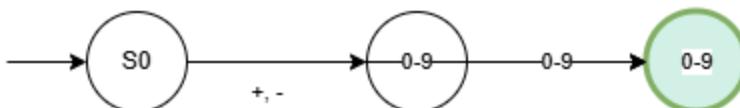
INTEGER LITERAL: [+-]?[0-9]+

NFA (Nondeterministic Finite Automaton)

q0 —ε, +, -→ q1 —0-9→ 0-9

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

Minimized DFA (Deterministic Finite Automaton)

S0 —+, -→ 0-9 —0-9→ 0-9

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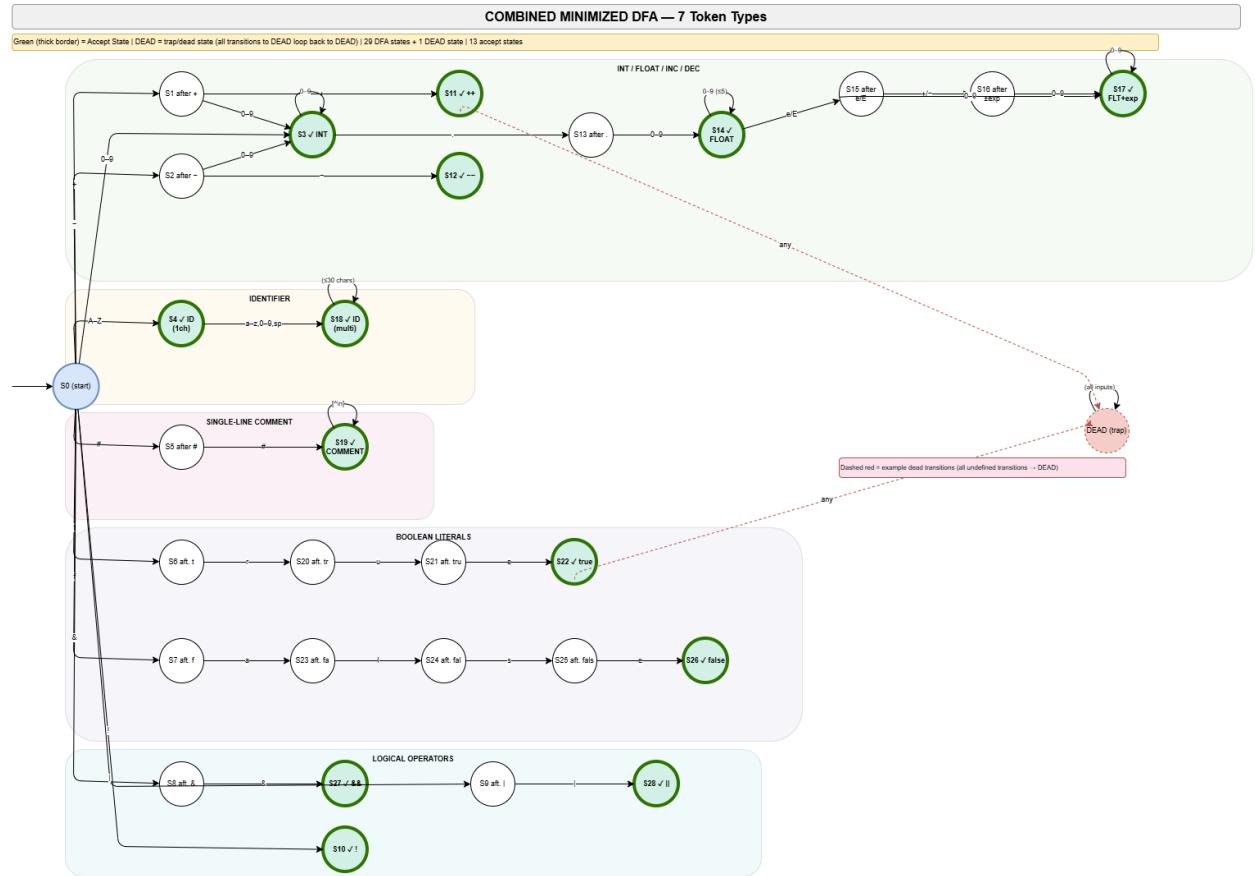
DFA Transition Table

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	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	S22	—	—	—	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

INT / FLOAT / INC / DEC

- S1 after +
- S2 after −
- S3 ✓ INT
- S11 ✓ ++
- S12 ✓ −−
- S13 after .
- S14 ✓ FLOAT
- S15 after e/E
- S16 after exps
- S17 ✓ FLT+exp

IDENTIFIER

- S4 ✓ ID (1ch)
- S18 ✓ ID (multi)

SINGLE-LINE COMMENT

- S5 after #
- S19 ✓ COMMENT

BOOLEAN LITERALS

- S6 aft. t
- S20 aft. tr
- S21 aft. tru
- S22 ✓ true
- S7 aft. f
- S23 aft. fa
- S24 aft. fal
- S25 aft. fals
- S26 ✓ false

LOGICAL OPERATORS

- S8 aft. &
- S27 ✓ &&
- S9 aft. |
- S28 ✓ ||
- S10 ✓ !

S0 (start)

DEAD (trap)

Dashed red = example dead transitions (all undefined transitions → DEAD)

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(||).

COMBINED NFA — 7 Token Types

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

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

IDENTIFIER [A-Z][a-z

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

BOOLEAN LITERALS (true|false)

INC/DEC OPERATORS (++|--)

LOGICAL OPERATORS (&&|||[]|!)

The combined NFA uses a single shared start state with ε-transitions to each sub-NFA. ε = epsilon (empty string). q1 in INT NFA accepts since ε-path means 0 sign chars. q1 and q2 are both accept states in Identifier NFA.