Automata Design Document

Github Repository: https://github.com/ibbi1020/i230816-i230500-B

1. Regular Expressions

The following regular expressions define the lexical patterns for the SimpleLang compiler.

1.1 Keywords

Regex: (start + finish + loop + condition + declare + output + input + function + return + break + continue + else)

1.2 Identifiers

Regex: upper (lower + digit + _)^k, 0 ≤ k ≤ 30

1.3 Integer Literals

Regex: (sign + ε) digit digit^*

1.4 Floating-Point Literals

Regex: (sign + ε) digit digit^* . digit^k (((E + e) (sign + ε) digit digit^*) + ε), 1 ≤ k ≤ 6

1.5 String Literals

Regex: " ((Σ - {" , \ , \n}) + (\ (" + \ + n + t + r)))^* "

1.6 Character Literals

Regex: ((Σ - {' , \ , \n}) + (\ (' + \ + n + t + r)))

1.7 Boolean Literals

Regex: (true + false)

1.8 Operators & Punctuators

- **Arithmetic:** (** + + + - + * + / + %)

- **Relational:** (== + != + <= + >= + < + >
- **Logical:** (&& + || + !)
- **Assignment:** (+= + -= + *= + /= + =)
- **Increment/Decrement:** (++ + --)
- **Punctuators:**((+) + { + } + [+] + , + ; + :)

1.9 Comments & Whitespace

- **Single-line:** ##[^\n]*
- **Multi-line:** #*([^*]|*+[^*#])**+#
- **Whitespace:** [\t\r\n]+ (Skipped during tokenization)

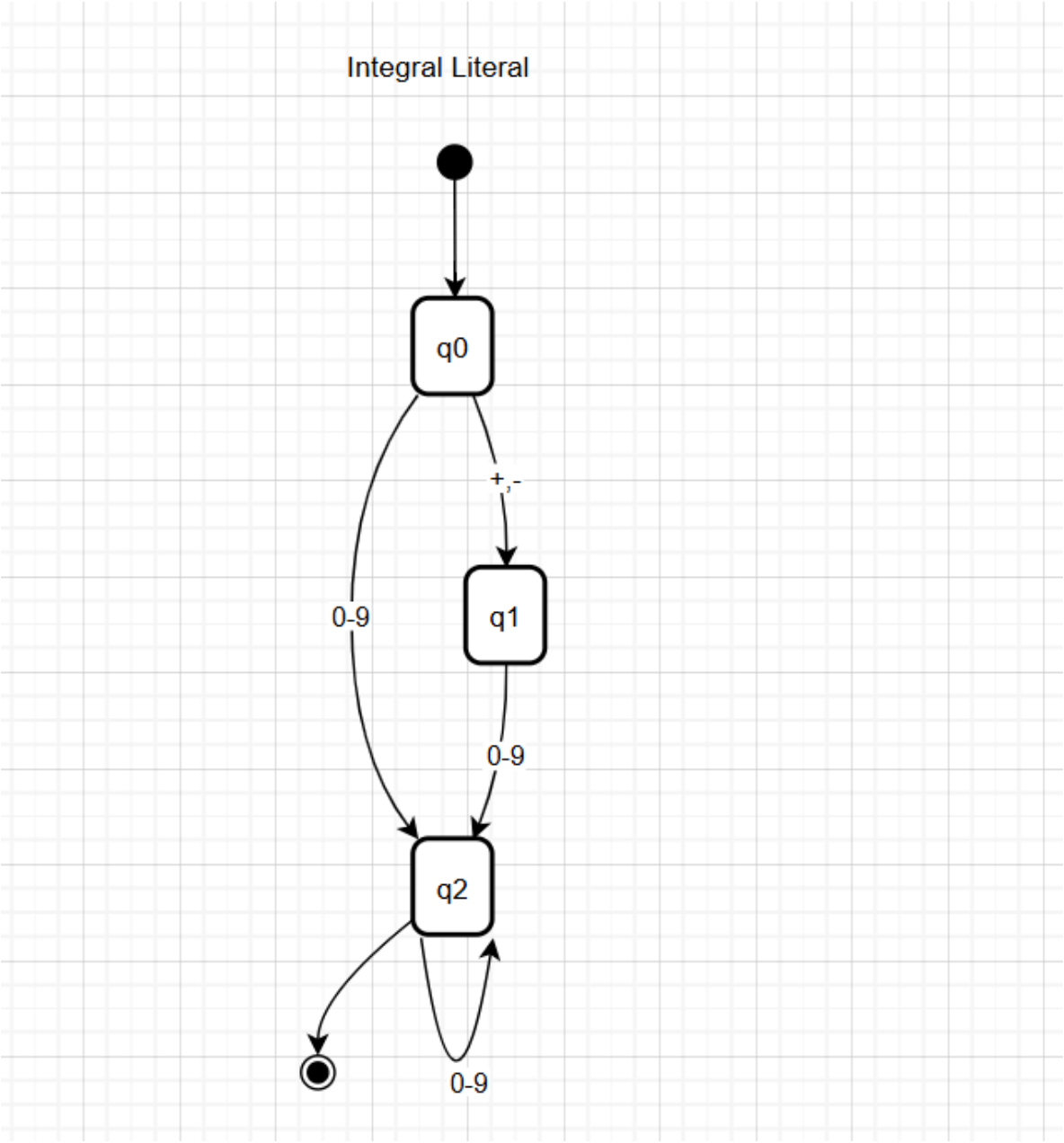
2. NFA Diagrams

This section presents Non-Deterministic Finite Automata (NFA) for the mandatory and selected additional token types.

2.1 Integer Literal

Regex: [+-]?[0-9]+

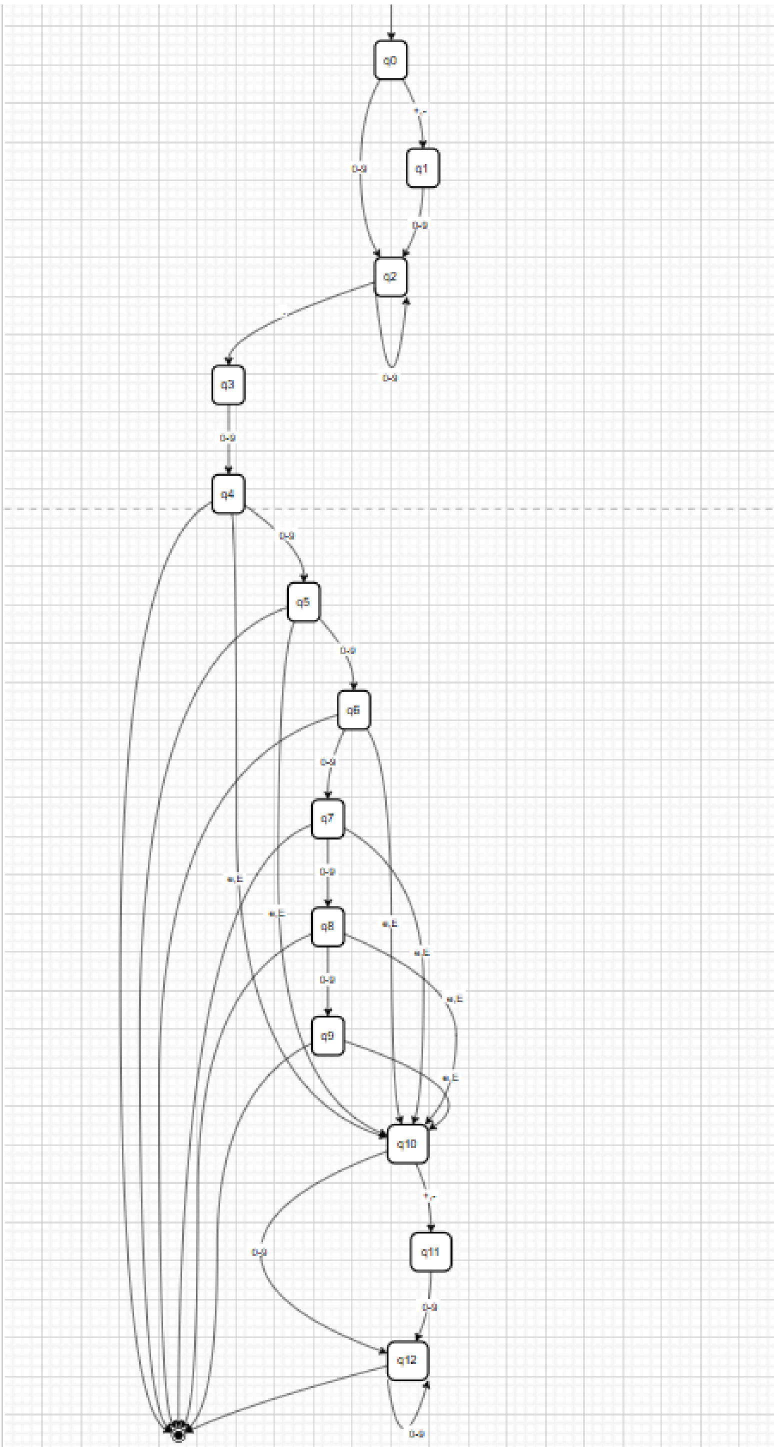
- **Start State:** q0
- **Accept State:** q2



2.2 Floating-Point Literal

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

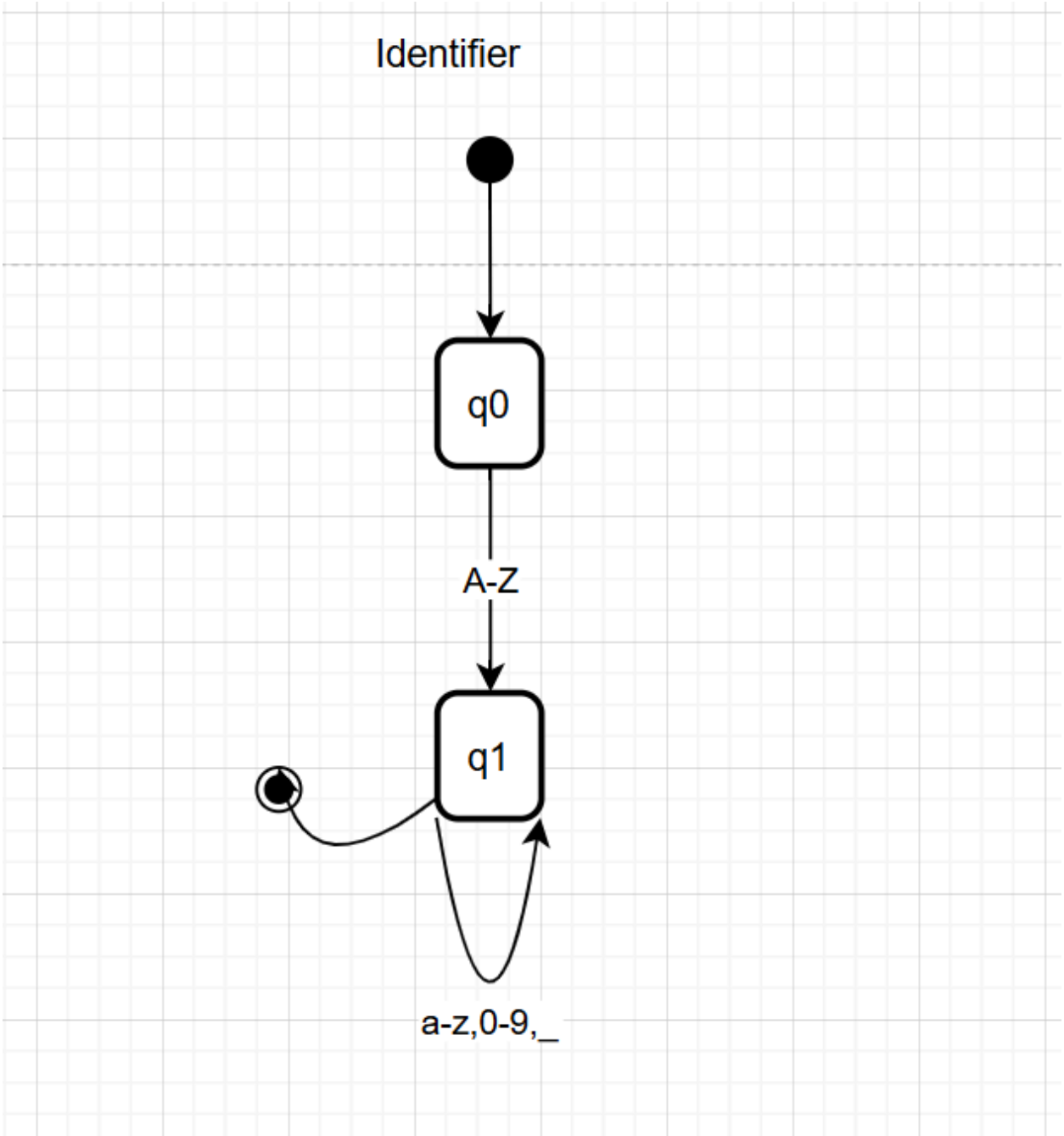
- **Start State:** q0
- **Accept States:** q4, q5, q6, q7, q8, q9, q12



2.3 Identifier

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

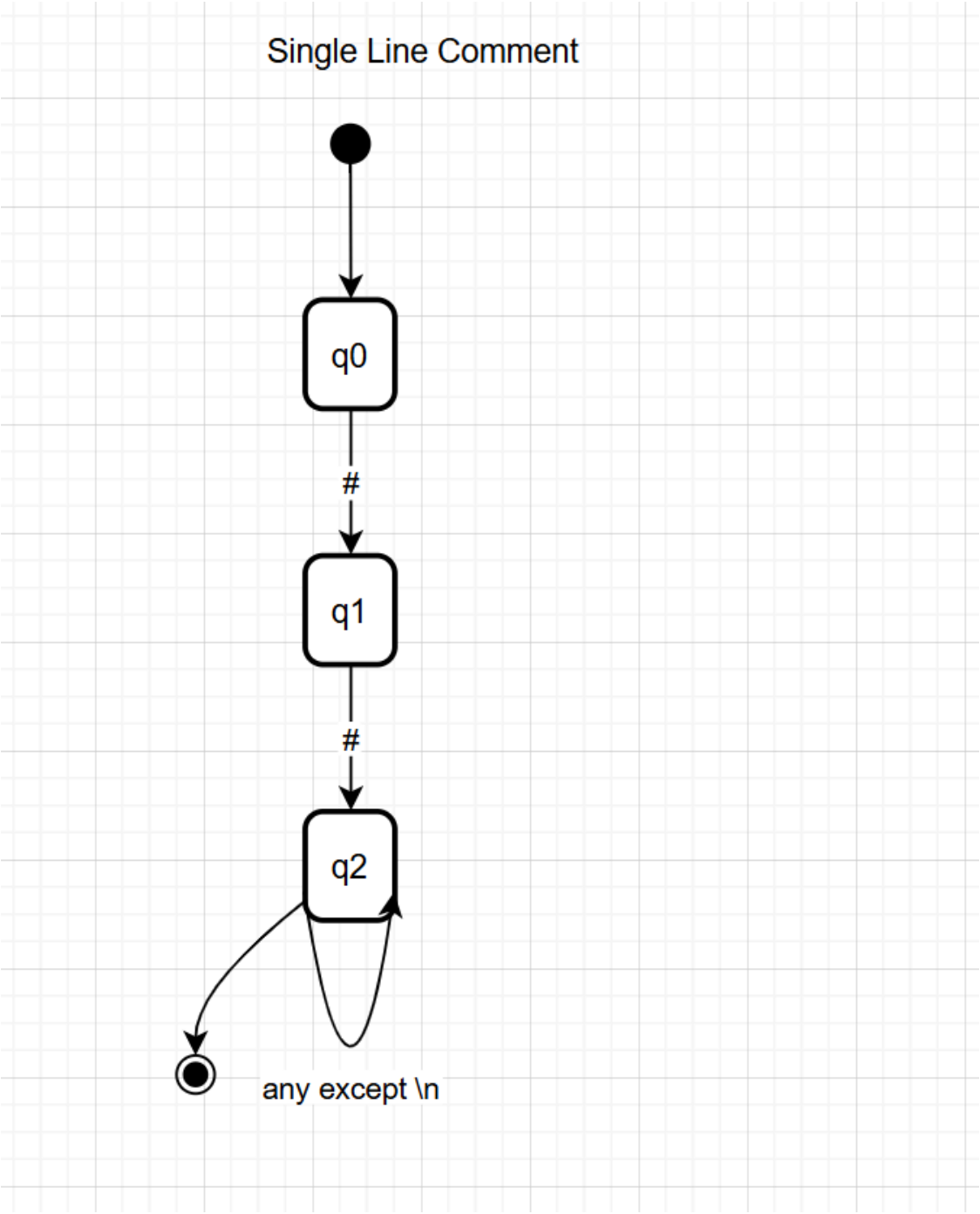
- **Start State:** q0
- **Accept State:** q1



2.4 Single-line Comment

Regex: ##[^\n]*

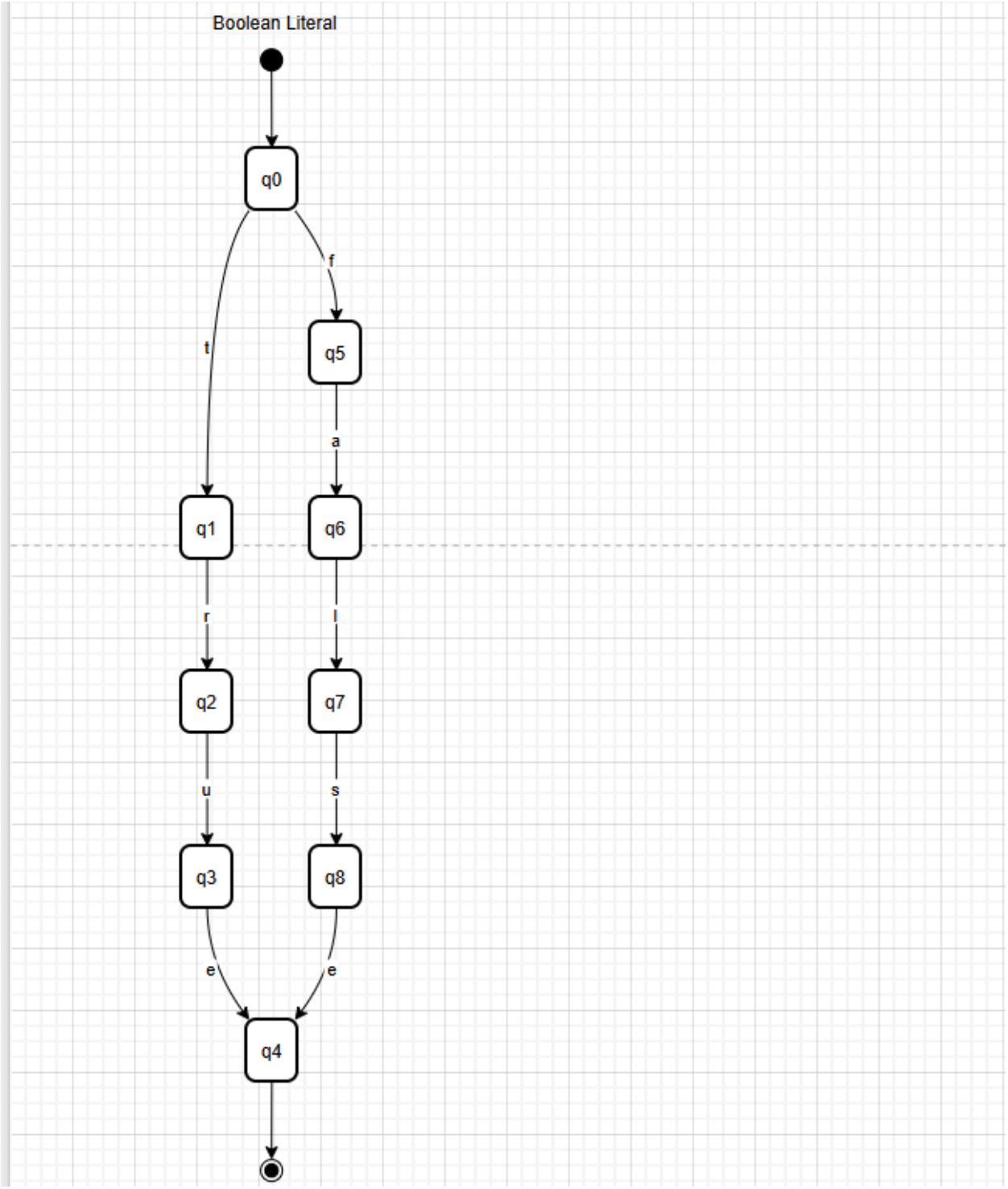
- **Start State:** q0
- **Accept State:** q2



2.5 Boolean Literal (Additional Type 1)

Regex: (true|false)

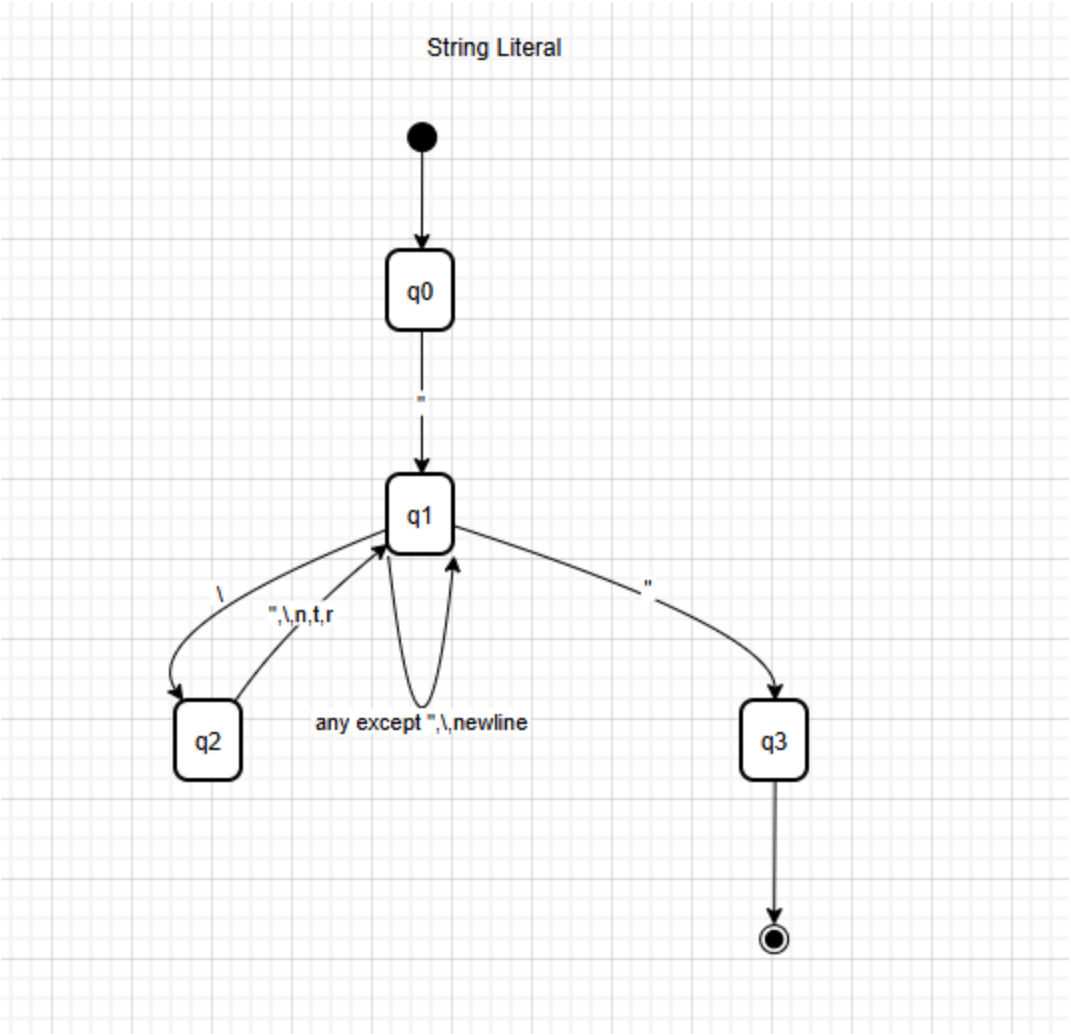
- **Start State:** q0
- **Accept State:** q4



2.6 String Literal (Additional Type 2)

Regex: "([^"\\\n]|\\["\\ntr])*"

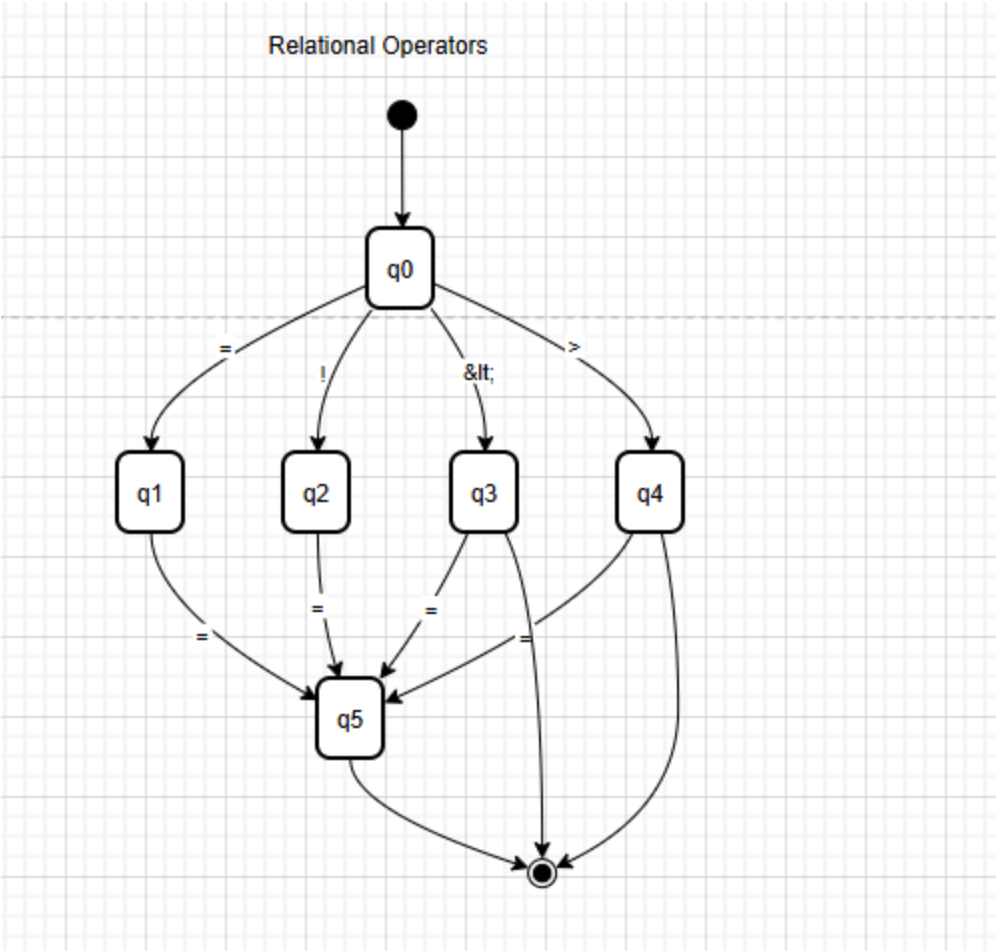
- **Start State:** q0
- **Accept State:** q3



2.7 Relational Operators (Additional Type 3)

Regex: (==|!=|<=|>=|<|>)

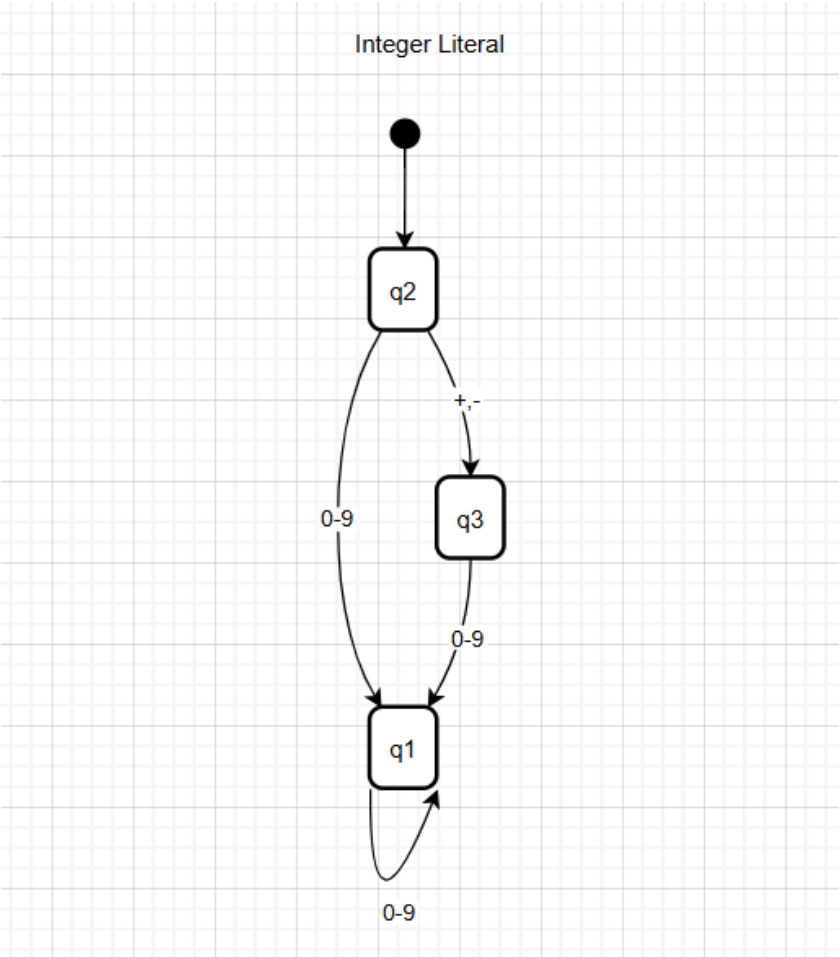
- **Start State:** q0
- **Accept States:** q3, q4, q5



3. Minimized DFA & Transition Tables

This section presents the minimized Deterministic Finite Automata (DFA) derived from the NFAs above, along with their state transition tables.

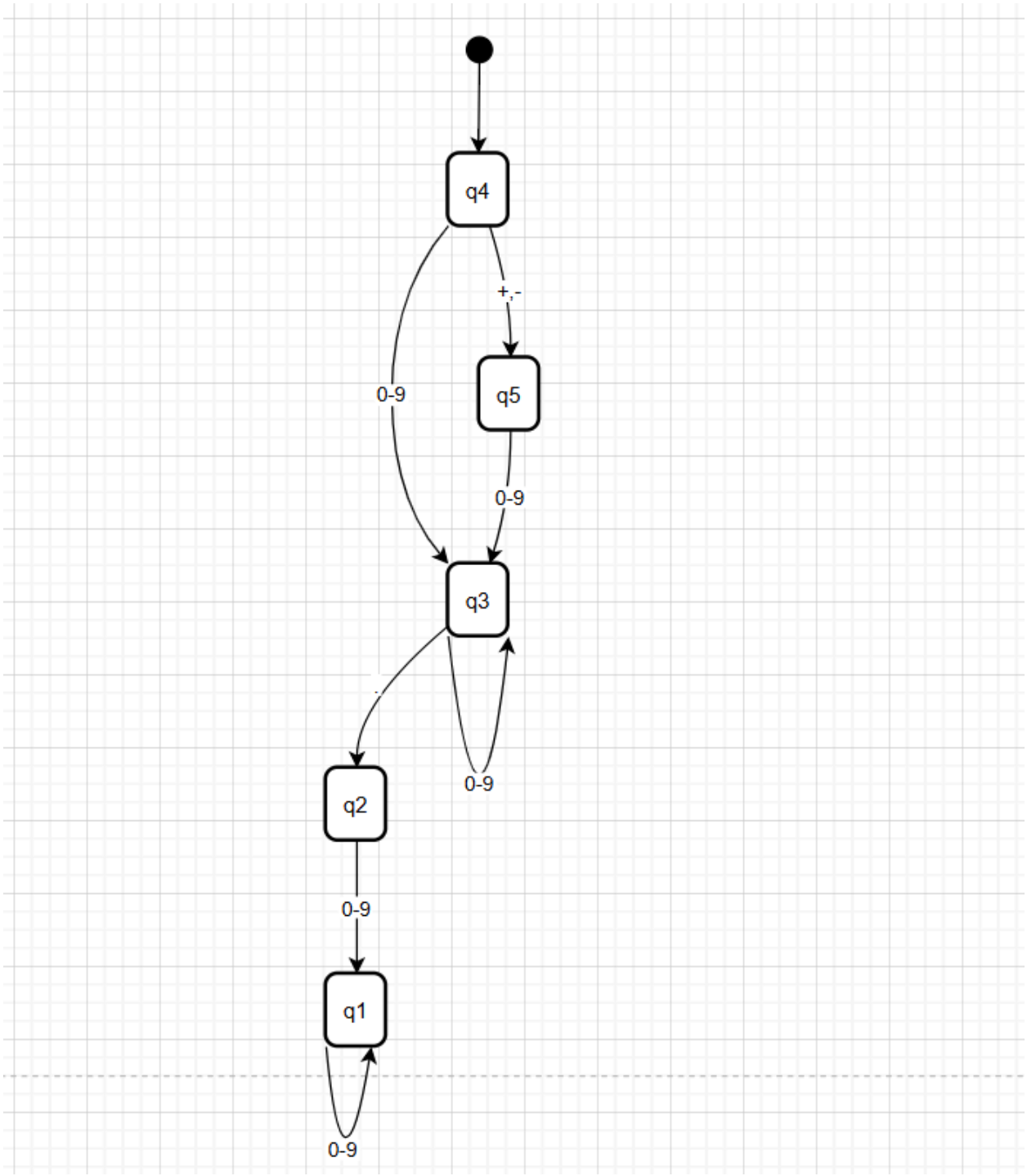
3.1 Integer Literal DFA



Transition Table:

State	+	-	0	1	2	3	4	5	6	7	8	9
1	NULL	NULL	1	1	1	1	1	1	1	1	1	1
2	3	3	1	1	1	1	1	1	1	1	1	1
3	NULL	NULL	1	1	1	1	1	1	1	1	1	1

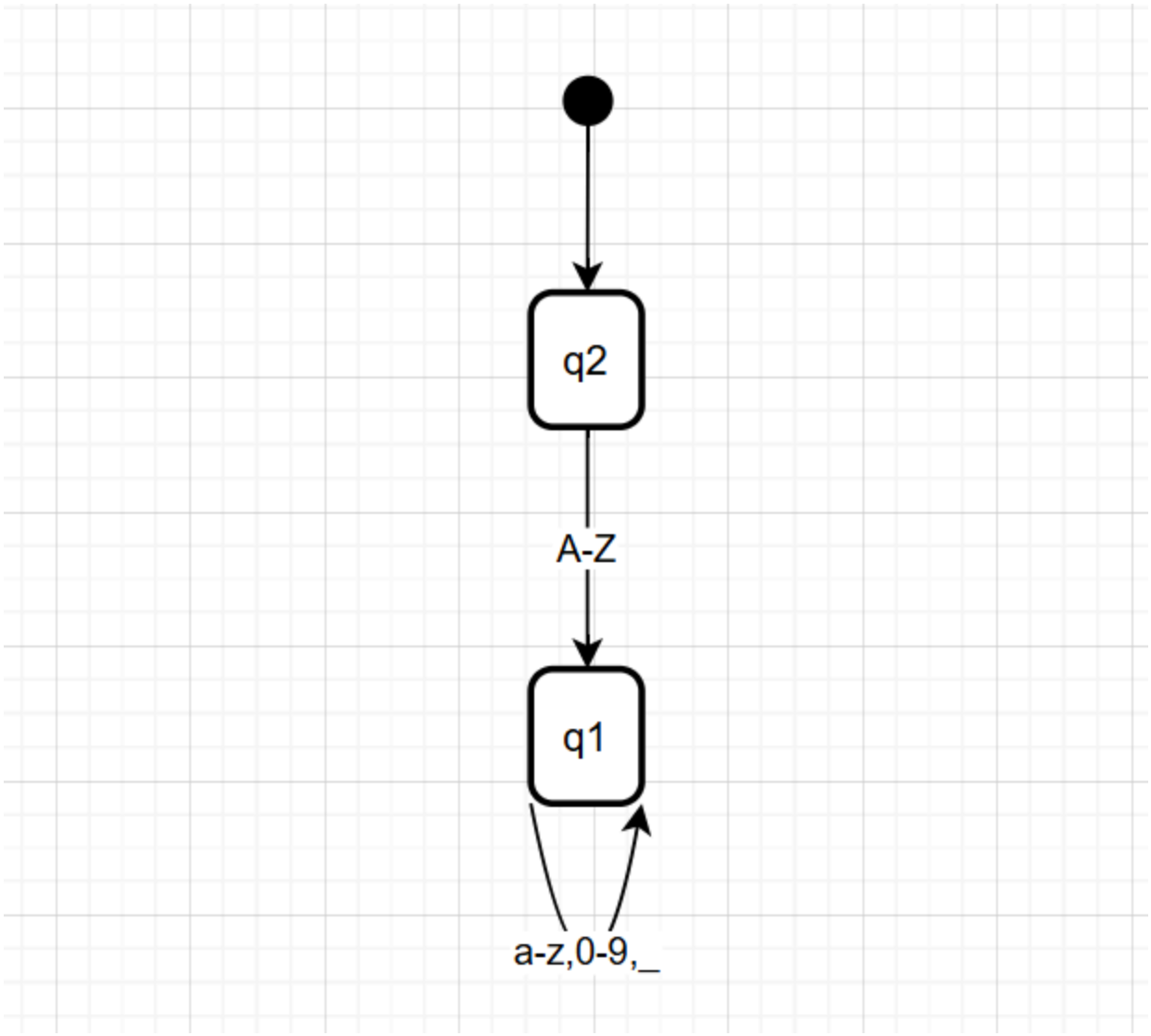
3.2 Floating-Point Literal DFA



Transition Table:

State	+	-	.	0	1	2	3	4	5	6	7	8	9
1	NULL	NULL	NULL	1	1	1	1	1	1	1	1	1	1
2	NULL	NULL	NULL	1	1	1	1	1	1	1	1	1	1
3	NULL	NULL	2	3	3	3	3	3	3	3	3	3	3
4	5	5	NULL	3	3	3	3	3	3	3	3	3	3
5	NULL	NULL	NULL	3	3	3	3	3	3	3	3	3	3

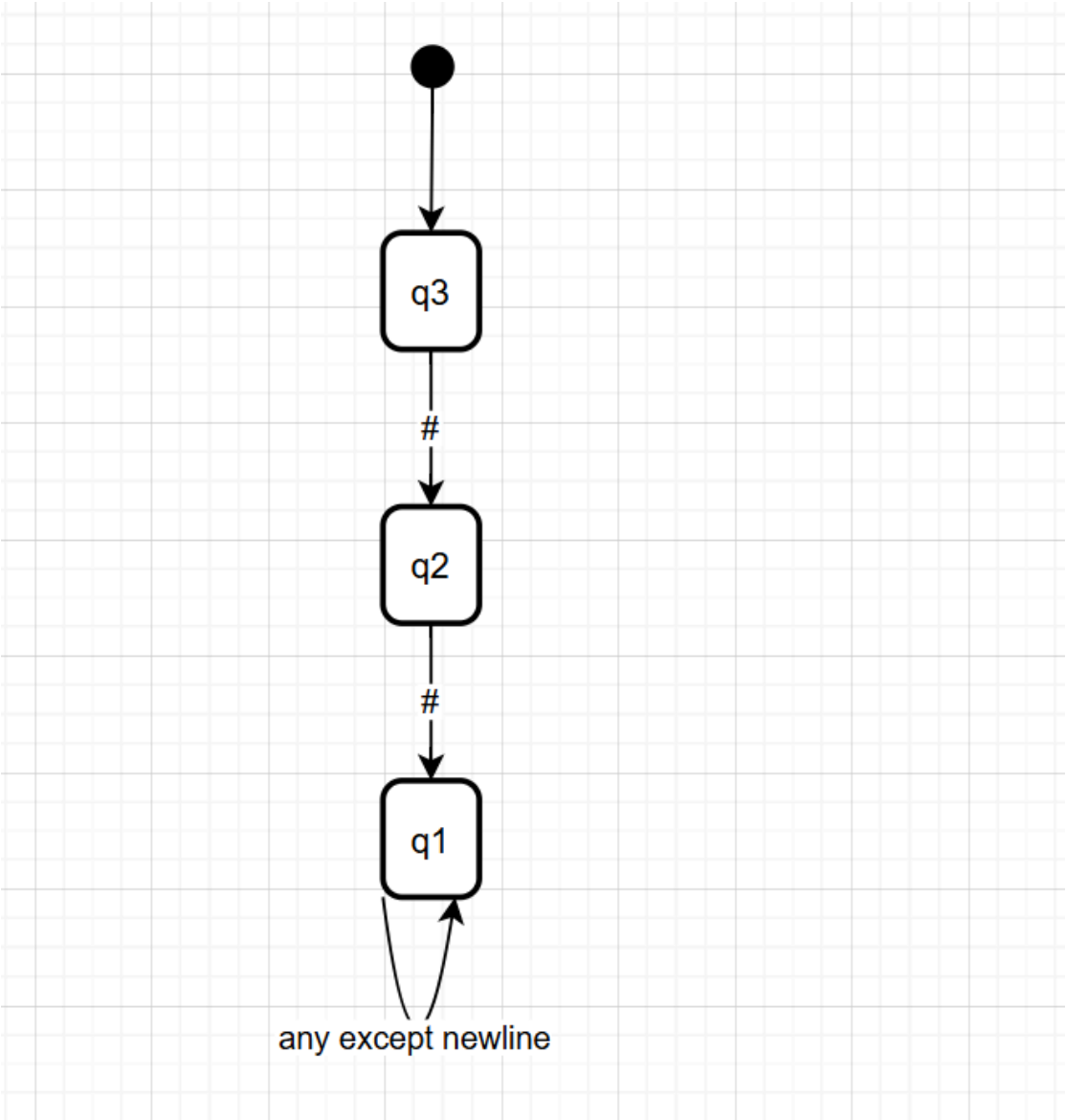
3.3 Identifier DFA



Transition Table:

State	0-9	A-Z	_	a-z
1	1	NULL	1	1
2	NULL	1	NULL	NULL

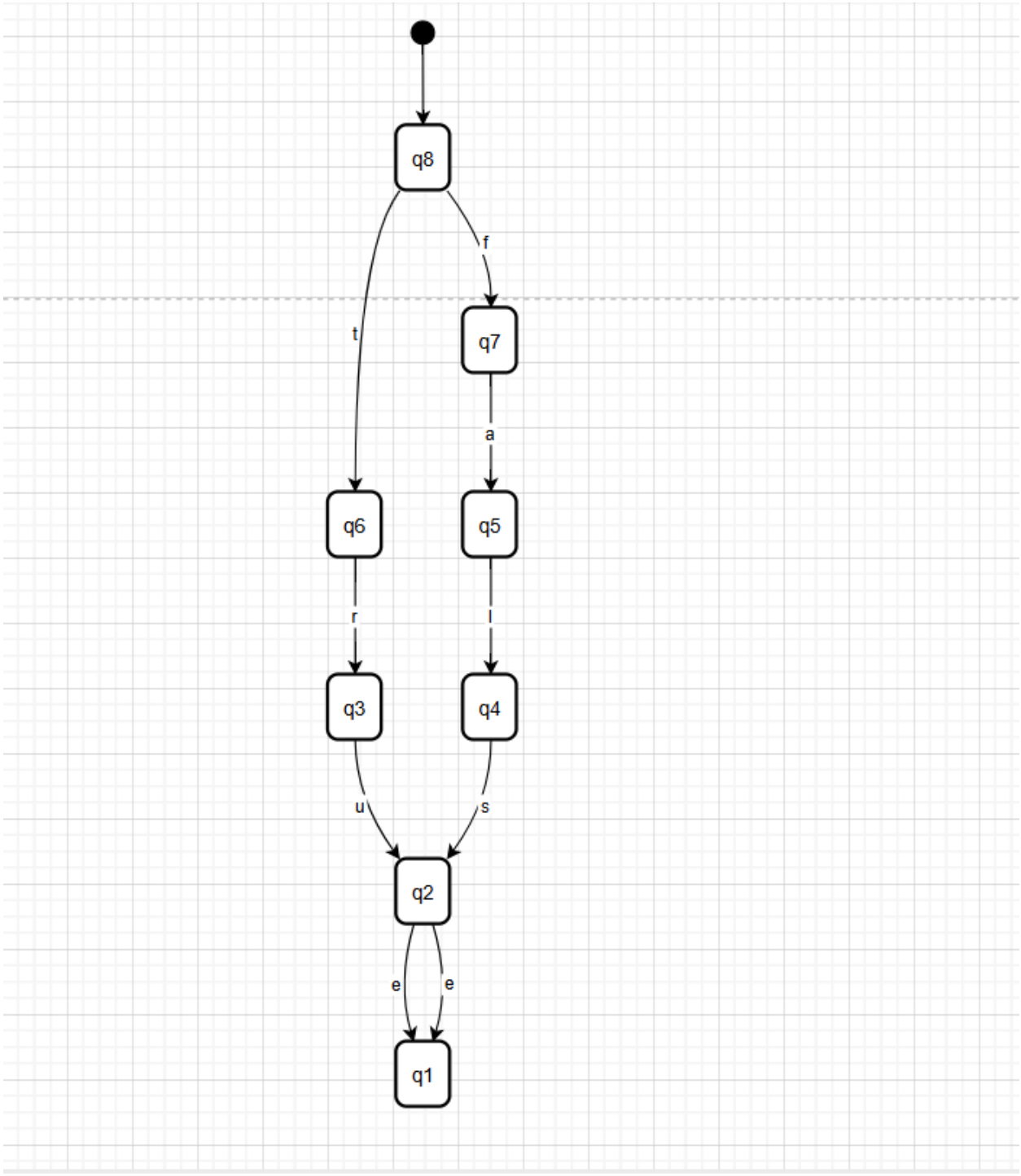
3.4 Single-line Comment DFA



Transition Table:

State	#	other
1	1	1
2	1	NULL
3	2	NULL

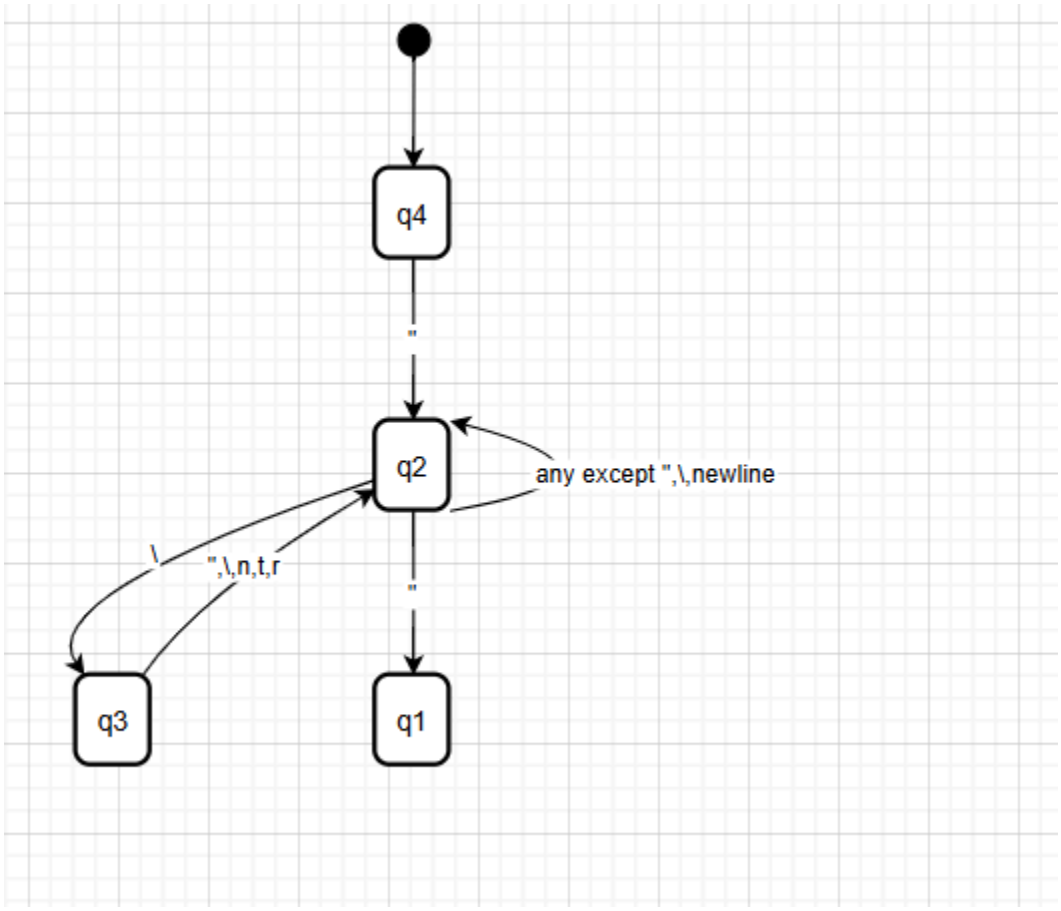
3.5 Boolean Literal DFA



Transition Table:

State	t	r	u	e	f	a	l	s
1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
2	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL
3	NULL	NULL	NULL	NULL	NULL	NULL	2	NULL
4	NULL	NULL	NULL	NULL	NULL	3	NULL	NULL
5	NULL	NULL	NULL	4	NULL	NULL	NULL	NULL
6	NULL	2	NULL	NULL	NULL	NULL	NULL	NULL
7	NULL	NULL	NULL	NULL	5	NULL	NULL	NULL
8	6	NULL	NULL	NULL	7	NULL	NULL	NULL

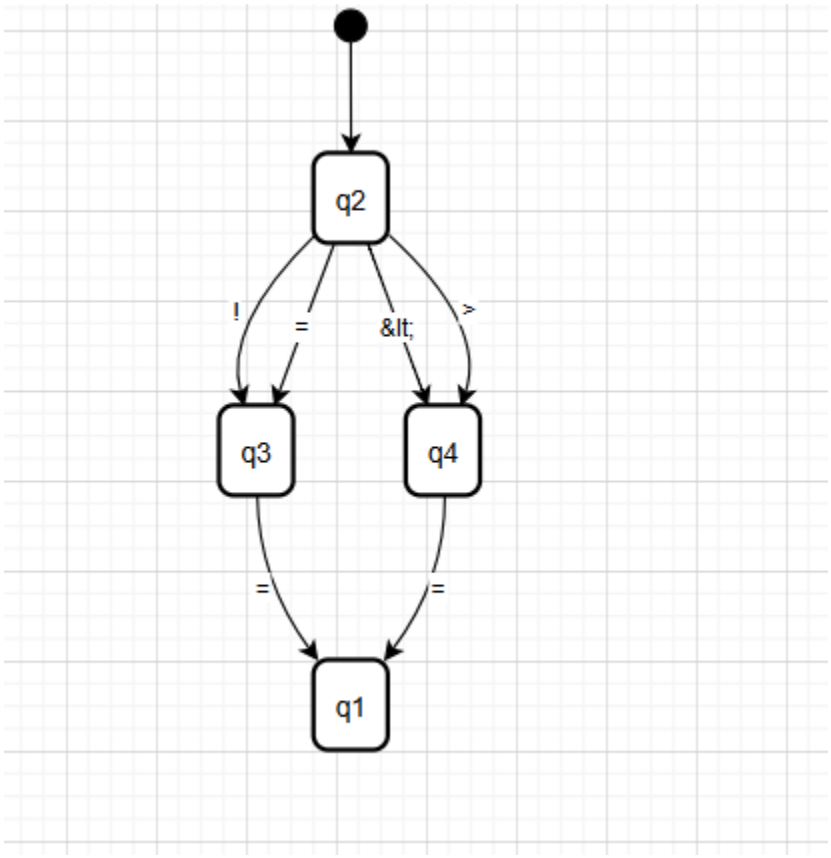
3.6 String Literal DFA



Transition Table:

State	"	\	normal	n	t	r
1	NULL	NULL	NULL	NULL	NULL	NULL
2	1	3	2	NULL	NULL	NULL
3	2	2	NULL	2	2	2
4	2	NULL	NULL	NULL	NULL	NULL

3.7 Relational Operators DFA



Transition Table:

State	!	<	=	>
1	NULL	NULL	NULL	NULL
2	3	4	3	4
3	NULL	NULL	1	NULL
4	NULL	NULL	1	NULL