Regular Expressions to Finite Automata

From algebraic patterns to state machines

Thompson construction • ε-NFA • subset construction

Lecture focus: construction rules • worked examples • verification

Learning outcomes

01 Explain

Relate regular expressions, ε-NFAs and DFAs.

02 Construct

Build automata for symbols, union, concatenation and star.

03 Convert

Apply ε-closure and subset construction.

04 Verify

Trace accepted and rejected strings.

The equivalence theorem

Regular expression R

Describes a regular language L(R).

ε-NFA N

Thompson’s construction produces N with L(N) = L(R).

DFA D

Subset construction produces D with L(D) = L(N).

Base cases



Literal a: accepts only a.

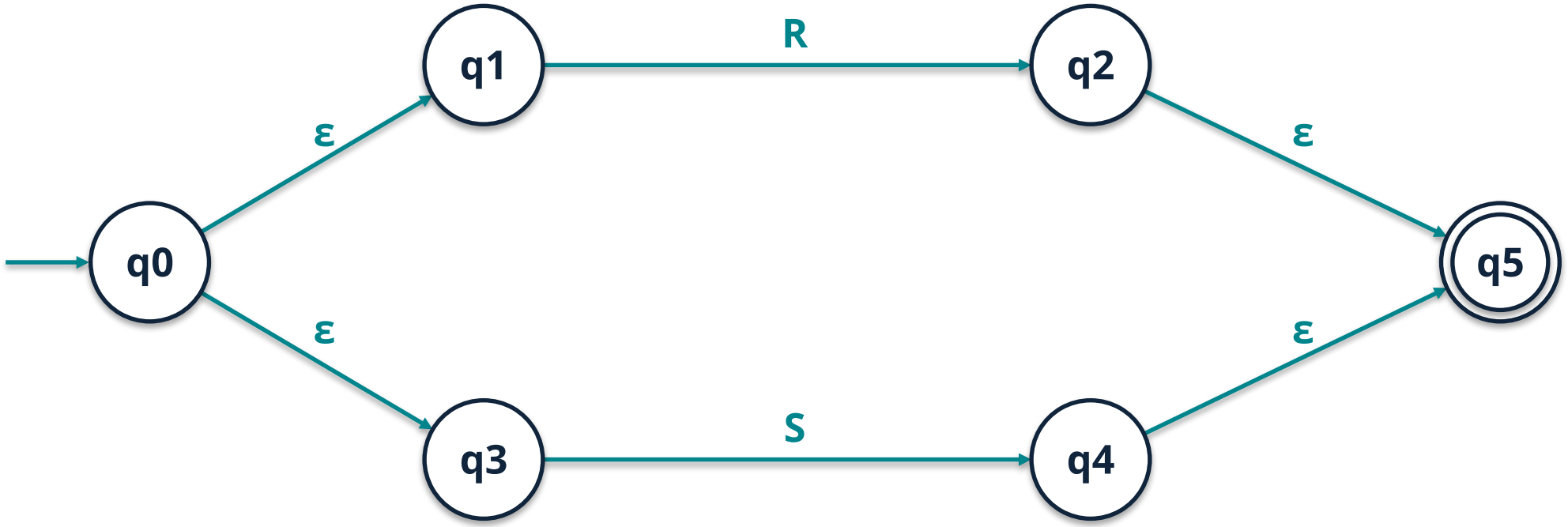


Empty string: accepts only ε.



Empty language: no connecting transition.

Union: R | S



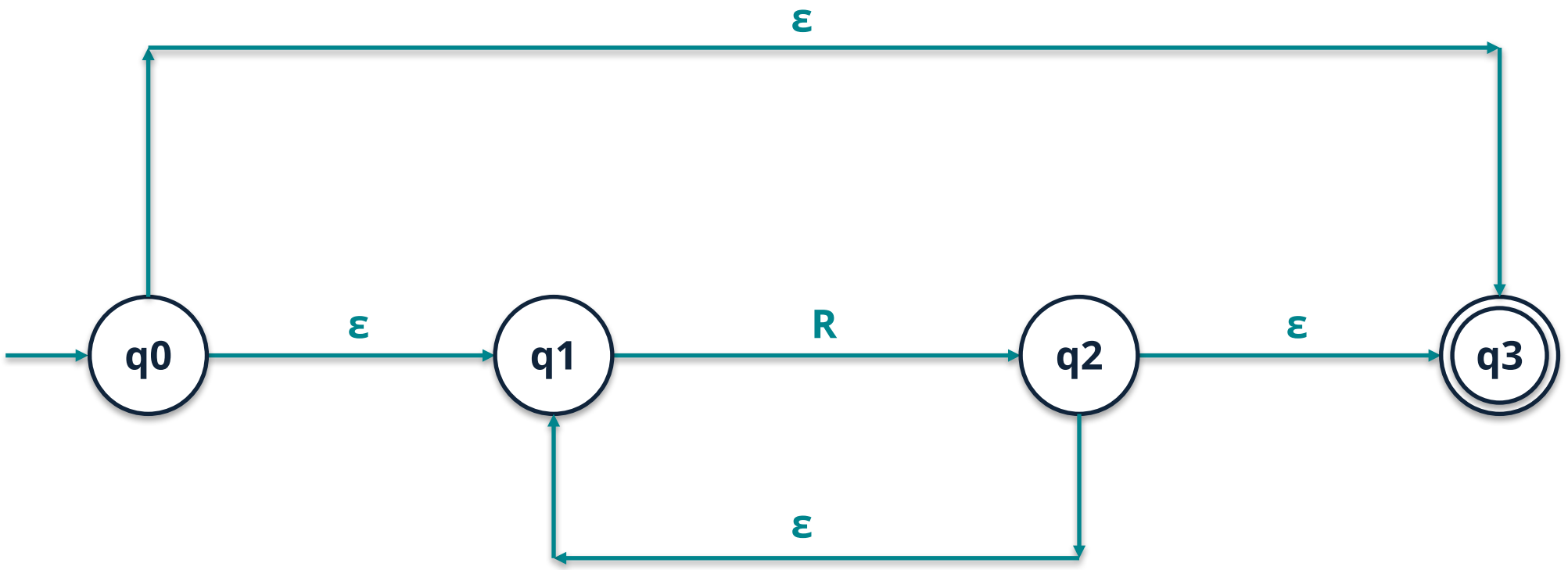
R and S denote entire subautomata; choose either branch.

Concatenation: RS



Connect the final state of R to the start of S by ε; only the final of S accepts.

Kleene star: R*



The bypass accepts ε; the return path allows repeated copies of R.

Thompson’s construction algorithm

Parse the expression

Use precedence: star, concatenation, then union.

Build base fragments

Each fragment has one entry and one exit.

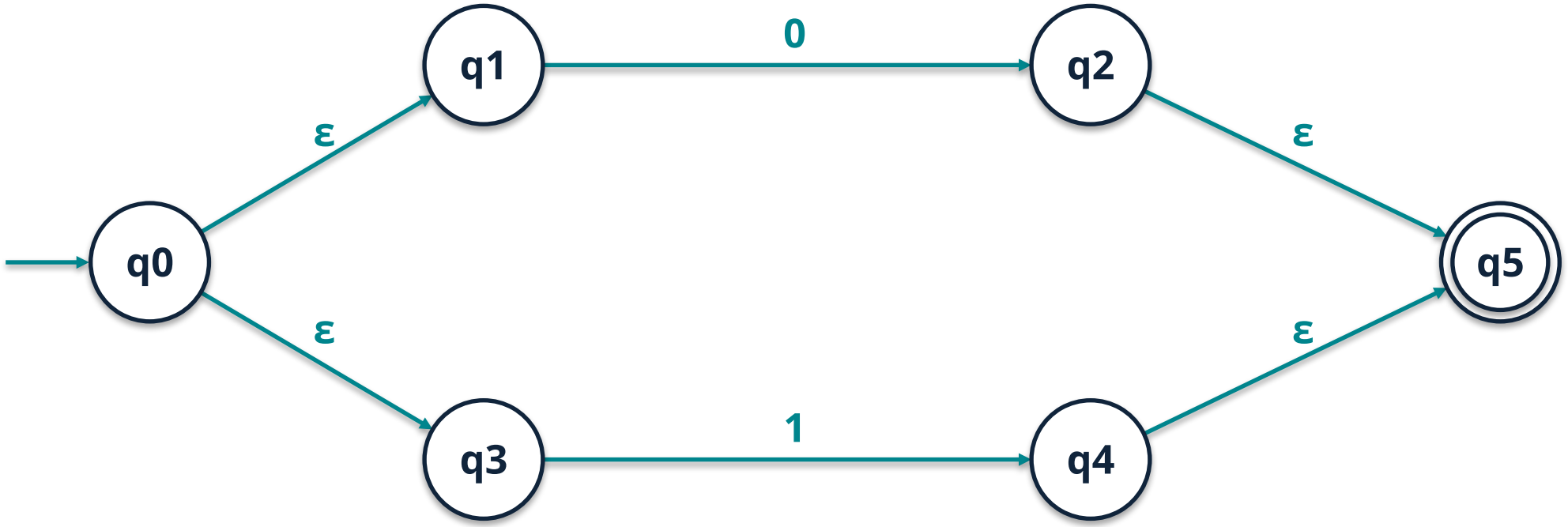
Combine recursively

Apply the rule for each operator; introduce fresh states.

Finish

Mark only the outer entry as start and outer exit as accepting.

Construct an ε-NFA for 0 | 1



Accepts 0 or 1, but not ε or any string of length two.

Construct an ε-NFA for 01



Exactly one 0 followed by one 1; q3 is the sole accepting state.

Construct an NFA for (0 | 1)*01

A compact equivalent NFA (after simplifying Thompson fragments)



q0 reads any prefix; nondeterministically guess the final 01.

Trace the input 101

Input read	Possible states	Reason
ε	{q0}	Initial state
1	{q0}	Prefix loop
10	{q0, q1}	Loop or start suffix
101	{q0, q2}	Prefix or completed suffix

Accepted: at least one possible state (q2) is accepting.

From ε-NFA to DFA

ε-closure(S)

All states reachable from S using zero or more ε transitions.

Initial DFA state

ε-closure({start state}).

Transition on a

ε-closure(move(S, a)), where move collects all a-successors.

Accepting DFA states

Any reachable subset containing an NFA accepting state.

Subset construction: suffix 01

DFA state	NFA subset	0	1
→ A	{q0}	B	A
B	{q0, q1}	B	C
* C	{q0, q2}	B	A

No ε edges in this compact NFA. C accepts; all three subsets are reachable.

Classroom exercises

Q1 Construct

Draw a Thompson ε-NFA for 1*.

Q2 Construct

Draw a Thompson ε-NFA for (0 | 1)0.

Q3 Verify

For (0 | 1)*01, decide whether ε, 01, 110 and 1001 are accepted.

Q4 Explain

Why does a DFA subset accept if it contains an NFA final state?

Solutions and key takeaways

Q1

Use the star fragment on slide 7 with R = 1.

Q2

Use the union fragment on slide 9; join its exit to a new 0 fragment by ε.

Q3

Accept 01 and 1001; reject ε and 110.

Q4

An NFA accepts if at least one complete computation reaches a final state.