Finite Automata to Regular Expressions

From state machines to algebraic patterns

GNFA • state elimination • Arden’s theorem

Lecture focus: conversion rules • worked examples • verification

Learning outcomes

01 Explain

Relate finite automata and regular expressions.

02 Construct

Normalize an automaton as a GNFA.

03 Eliminate

Apply the state elimination rule systematically.

04 Solve

Use Arden’s theorem and verify equivalent expressions.

Two representations, one language

Finite automaton

Accepts strings by following transitions from an initial state to a final state.

Regular expression

Describes the same set of strings using union (∪), concatenation and star (*).

Kleene’s theorem

Every finite automaton has an equivalent regular expression, and conversely.

Generalized nondeterministic finite automaton

Normalize

Add a fresh start s with ε to the old start; add a fresh final f with ε from every old final.

Label transitions

Use regular expressions; merge parallel edges by union. Missing edges have label ∅.

Invariant

The new start has no incoming edges; the new final has no outgoing edges.

State elimination formula

$$R_{ij} \leftarrow R_{ij} \cup R_{ik} (R_{kk})^* R_{kj}$$

R_{ij}

Existing direct path label from i to j.

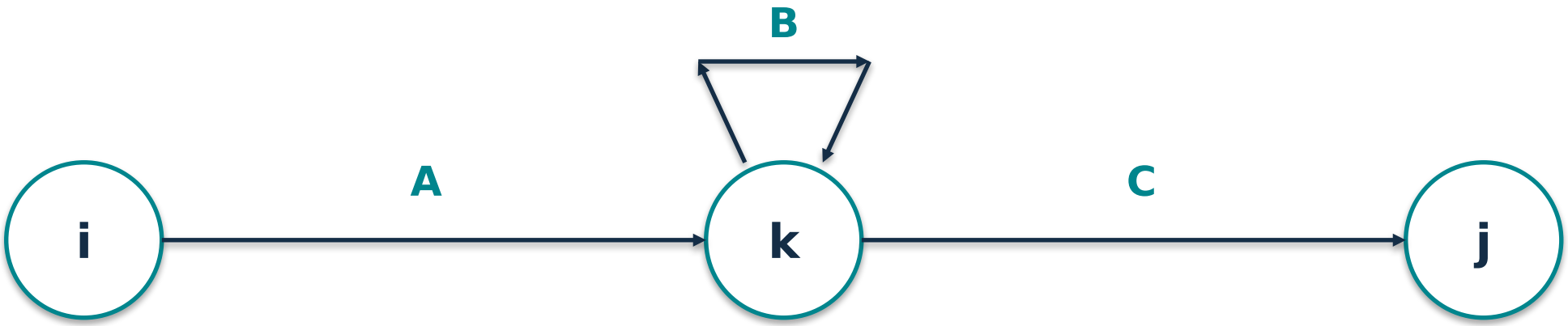
$R_{ik} (R_{kk})^* R_{kj}$

All paths through eliminated state k, including zero or more loops.

Update

Apply to all remaining ordered pairs (including i = j), then delete k.

Visualizing the elimination of k



New i → j label: D ∪ AB*C (D = existing direct label)

State elimination: step by step

Step 1

Normalize the FA to a GNFA with fresh start and final states.

Step 2

Choose an intermediate state k; update every remaining pair using the rule.

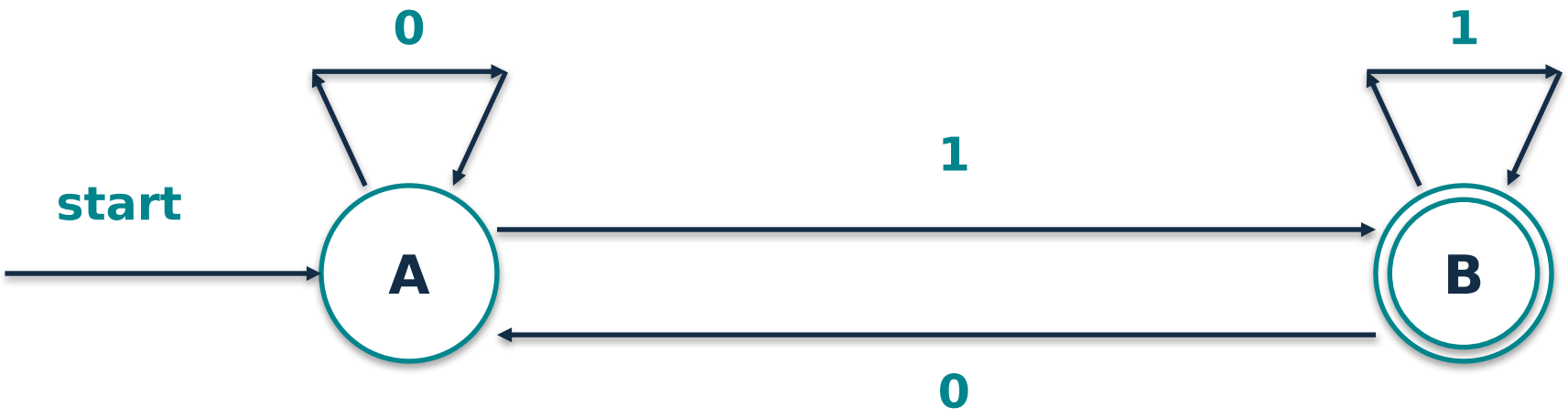
Step 3

Remove k and its incident transitions. Repeat until only s and f remain.

Step 4

Read the expression on s → f. Different elimination orders may yield equivalent expressions.

Binary strings ending in 1



A: start / last symbol not 1 **B: accepting / last symbol 1**

Normalize and eliminate A

Initial GNFA labels

s→A: ε; A→A: 0; A→B: 1; B→A: 0; B→B: 1; B→f: ε.

New s→B

∅ ∪ ε(0)*1 = 0*1.

New B→B

1 ∪ 0(0)*1 = 1 ∪ 00*1.

Remaining states

s, B, f. The B→f label remains ε; s→f is ∅.

Eliminate B and simplify

$$R = 0^*1(1 \cup 00^*1)^*$$

Eliminate B

$s \to f = \emptyset \cup 0^*1(1 \cup 00^*1)^*\varepsilon$.

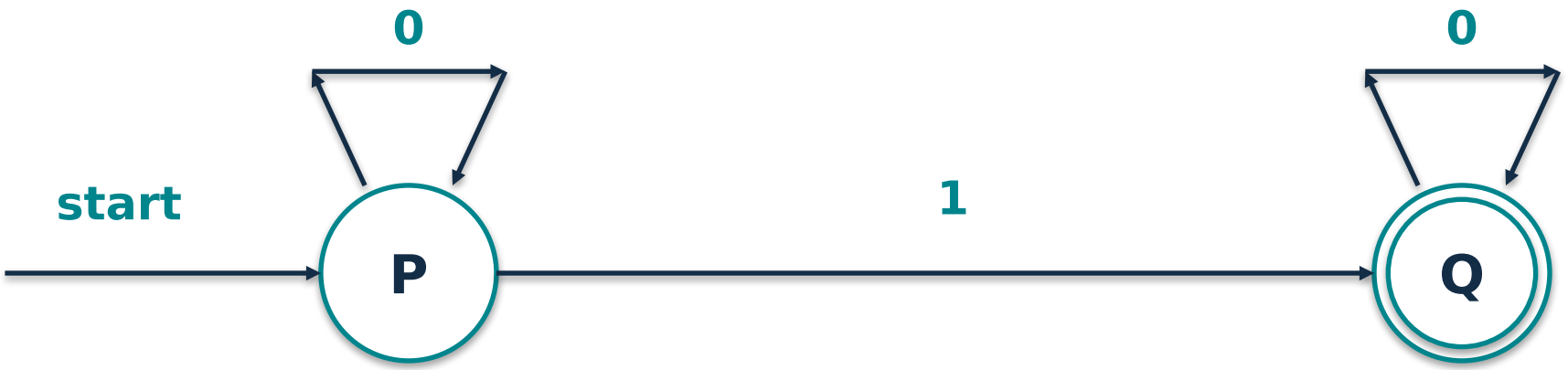
Equivalent expression

$(0 \cup 1)^*1$. Every block ends in 1, with zero or more preceding zeros.

Check

101 and 1 are accepted; ε, 10 and 000 are rejected.

A second example: exactly one 1



Partial DFA: a missing transition rejects (or goes to an implicit dead state).

Eliminate P, then Q

Normalize

$s \to P = \varepsilon$; $P \to P = 0$; $P \to Q = 1$; $Q \to Q = 0$; $Q \to f = \varepsilon$.

Remove P

$s \to Q = \varepsilon 0^* 1 = 0^* 1$.

Remove Q

$s \to f = (0^* 1) 0^* \varepsilon = 0^* 1 0^*$.

Verify

00100 is accepted; 000 and 101 are rejected.

Arden’s theorem

$$X = AX \cup B \quad \Rightarrow \quad X = A^*B$$

Condition

The solution is unique if $\varepsilon \notin L(A)$. A^*B is always the least solution.

Right-sided version

$X = XA \cup B \quad \Rightarrow \quad X = BA^*$ (same uniqueness condition).

Use

Define one equation per state; solve by substitution, then apply Arden’s theorem.

Solve example 2 with state equations

Define X_P, X_φ

Languages of strings that lead from P or Q to an accepting state.

Equation for Q

$X_\varphi = 0X_\varphi \cup \varepsilon$, so $X_\varphi = 0^*$.

Equation for P

$X_P = 0X_P \cup 1X_\varphi = 0X_P \cup 10^*$.

Apply Arden

$X_P = 0^*10^*$. P is the start, so this is the required expression.

Classroom exercises

Q1

One accepting start state loops on 0 and 1. Find its regular expression.

Q2

A nonfinal start A loops on 1 and goes to final B on 0; B loops on 1 only. Find R.

Q3

Solve $X = 01X \cup 1$ using Arden’s theorem.

Q4

Why do we add a fresh start and final before state elimination?

Solutions and key takeaways

Q1

(0 ∪ 1)*, including ε.

Q2

1*01*: exactly one 0.

Q3

(01)*1; ε is not in L(01), so the solution is unique.

Q4

They protect the endpoints while every original state can be eliminated. Both methods preserve the accepted language.