THEORY OF AUTOMATA & FORMAL LANGUAGES

Conversion of NFA to DFA

Subset construction: from possible states to a single state

Collect → Move → Discover → Verify

Why convert NFA to DFA?

Both machines recognize the same class of languages.

NFA: several possible states

A symbol may have many destinations.

A symbol may have no destination.

Track a set of active NFA states.

Accept if any active state is final.

Example: strings ending in ab.

DFA: one current state

Every symbol has one destination.

A DFA state represents a set of NFA states.

Start at the singleton set $\{q_0\}$.

Accept if the set contains a final state.

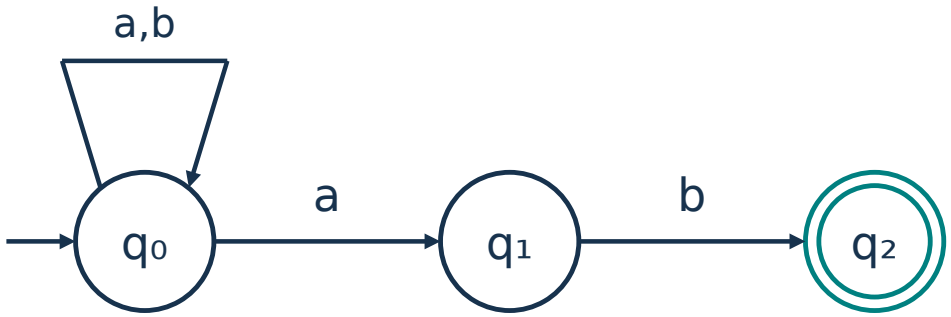
Only reachable subsets are needed.

The empty subset is a valid DFA state when reached.

Worked example: original NFA

$\Sigma = \{a,b\}$; initial q_0; $F = \{q_2\}$; language: strings ending in ab.

Original state diagram



NFA transition table

State	a	b
q_0	$\{q_0,q_1\}$	$\{q_0\}$
q_1	$\varnothing$	$\{q_2\}$
q_2	$\varnothing$	$\varnothing$

q_0 scans; q_1 guesses; q_2 accepts.

Missing NFA transitions are represented by the empty set $\varnothing$.

Step 1 — Define the subset rule

$\delta^D(S,x) = \bigcup \{\delta(q,x) : q \in S\}$.

Meaning of the rule

S is one DFA state (a set of NFA states).

x is the next input symbol.

Find δ(q,x) for each q in S.

Take the union of all destinations.

The resulting set is one DFA state.

Initial and final states

Start: $A = \{q_0\}$.

If a new subset appears, name it.

A subset is final if it intersects F.

Process each new subset for a and b.

Stop when no new subset appears.

At most 2^n subsets exist for an NFA with n states.

Step 2 — Process initial state A

A = $\{q_0\}$; calculate both outgoing transitions.

Read a from A

$\delta^D(A,a) = \delta(q_0,a)$

$= \{q_0,q_1\}$

This subset has not been seen.

Name it B = $\{q_0,q_1\}$.

Record A —a→ B.

Read b from A

$\delta^D(A,b) = \delta(q_0,b)$

$= \{q_0\}$

This is the existing state A.

Record A —b→ A.

Next process the new subset B.

A is not accepting because q_2 is not in A.

Step 3 — Discover remaining subsets

B = $\{q_0,q_1\}$; C = $\{q_0,q_2\}$.

Process B

On a: $\{q_0,q_1\} \cup \varnothing$

$= \{q_0,q_1\} = B$

On b: $\{q_0\} \cup \{q_2\}$

$= \{q_0,q_2\} = C$

C is a newly discovered subset.

Process C

On a: $\{q_0,q_1\} \cup \varnothing$

$= \{q_0,q_1\} = B$

On b: $\{q_0\} \cup \varnothing$

$= \{q_0\} = A$

No new subset remains.

C is accepting because it contains q_2. A and B are not.

Step 4 — Construct the equivalent DFA

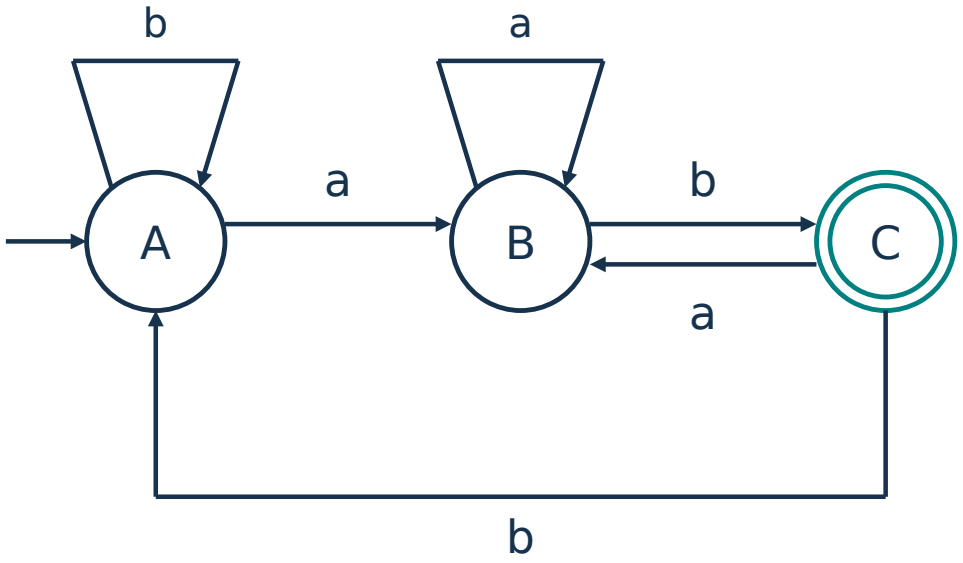
Every row now has exactly one destination per input.

Complete DFA transition table

State	a	b
→ A	B	A
B	B	C
* C	B	A

$A=\{q_0\}$; $B=\{q_0,q_1\}$; $C=\{q_0,q_2\}$

Resulting DFA



The reachable subsets are A, B and C; ∅ is not reachable here.

Step 5 — Verify language equivalence

Compare the subset trace with the DFA trace on the same input.

Accepted example: aab

Start: A = $\{q_0\}$.

Read a: B = $\{q_0,q_1\}$.

Read a: B = $\{q_0,q_1\}$.

Read b: C = $\{q_0,q_2\}$.

C contains q_2 → accept.

Rejected example: aba

Start: A = $\{q_0\}$.

Read a: B = $\{q_0,q_1\}$.

Read b: C = $\{q_0,q_2\}$.

Read a: B = $\{q_0,q_1\}$.

B does not contain q_2 → reject.

Acceptance is checked only after the entire input is consumed.

Practice and summary

Apply subset construction to any ordinary NFA.

Practice questions

1. Is every possible subset required?

2. When is a subset accepting?

3. What if the union is empty?

4. Does the example accept bab?

5. Does the example accept ε?

Explained answers

1. No, only reachable subsets.

2. If it contains an NFA final state.

3. Add ∅, with all transitions to ∅.

4. Yes: A → A → B → C.

5. No: A is not accepting.

Next: direct conversion of ε-NFA to DFA using ε-closure.