THEORY OF AUTOMATA & FORMAL LANGUAGES

Equivalence of Finite Automata

Do different machines recognize the same language?

Compare → Pair states → Search → Prove or refute

What does equivalence mean?

Two machines may look different but recognize exactly the same strings.

Formal definition

M_1 and M_2 are equivalent if $L(M_1)=L(M_2)$.

For every $w \in \Sigma^*$, both accept or both reject.

Same number of states is not required.

Different state names do not matter.

Convert an NFA to DFA first if necessary.

How to test equivalence

Use the same alphabet for both machines.

Complete missing DFA transitions with a sink.

Start from the pair of initial states.

Explore all reachable pairs of states.

A pair with opposite acceptance is a mismatch.

Testing a few strings alone cannot prove equivalence.

Example 1 — Two different DFAs

Alphabet $\Sigma = \{a,b\}$; both machines are complete.

Machine M_1 (two states)

State	a	b
→ A	B	A
* B	B	A

Initial state: A; $F_1 = \{B\}$.

Meaning: the last symbol is a.

Machine M_2 (three states)

State	a	b
→ P	R	Q
Q	R	P
* R	R	Q

Initial state: P; $F_2 = \{R\}$.

Goal: prove equivalence without testing infinitely many strings.

Product-state method

Track both machines simultaneously for every input.

Mathematical rule

A product state is a pair (p,q).

Initial pair: (A,P).

$\delta((p,q),x) = (\delta_1(p,x), \delta_2(q,x))$.

Mismatch: exactly one member is final.

Search only pairs reachable from the start.

First calculations

(A,P) on a → (B,R)

Because $\delta_1(A,a)=B$ and $\delta_2(P,a)=R$.

(A,P) on b → (A,Q)

Because $\delta_1(A,b)=A$ and $\delta_2(P,b)=Q$.

Add both new pairs to the search.

A reachable mismatch is an accepting state of the symmetric-difference machine.

Complete reachable-pair table

Continue until every discovered pair has been processed.

Product transitions

Pair	a	b
→ (A,P)	(B,R)	(A,Q)
(B,R)	(B,R)	(A,Q)
(A,Q)	(B,R)	(A,P)

No further pairs are discovered.

Acceptance comparison

(A,P): reject / reject

(B,R): accept / accept

(A,Q): reject / reject

No reachable mismatch exists.

Therefore $L(M_1) = L(M_2)$.

All possible input strings reach one of these three pairs; equivalence is proved.

Verify with sample strings

Manual traces illustrate the proof; they are not the proof themselves.

Accepted input: ba

Start pair (A,P).

Read b → (A,Q).

Read a → (B,R).

Both B and R are accepting.

Both machines accept ba.

Rejected input: ab

Start pair (A,P).

Read a → (B,R).

Read b → (A,Q).

Neither A nor Q is accepting.

Both machines reject ab.

The empty string is rejected by both initial states.

Example 2 — Non-equivalent machines

Compare M_1 against M_3, which accepts an odd number of a symbols.

Machine M_3

State	a	b
→ E	O	E
* O	E	O

Initial state: E; $F_3 = \{O\}$.

E = even count; O = odd count.

Search reachable pairs

Start (A,E): reject / reject.

Read a → (B,O): accept / accept.

Read another a → (B,E).

B is accepting; E is rejecting.

Counterexample: aa.

One distinguishing string is sufficient to prove non-equivalence.

Alternative method — Minimization

Use the minimization algorithm from the preceding lecture.

Procedure

Remove unreachable states from each DFA.

Minimize both complete DFAs.

Match their initial states.

Compare acceptance and labeled transitions.

Check whether a state renaming matches them.

What the result means

Same minimal structure → equivalent.

Different minimal structure → not equivalent.

For M_2, merge P and Q into one state.

R remains separate; it maps to B.

The minimized M_2 matches M_1.

Minimal complete DFAs for the same language and alphabet are unique up to renaming.

Practice and summary

Use reachable pairs to prove equivalence or find a counterexample.

Practice questions

1. Must equivalent DFAs have equal size?

2. What if initial acceptance differs?

3. Is a mismatch in an unreachable pair useful?

4. What distinguishes M_1 and M_3?

5. When does the search stop?

Explained answers

1. No; their languages must agree.

2. ε is a counterexample.

3. No; only reachable pairs matter.

4. aa is accepted only by M_1.

5. At a mismatch or when no new pair exists.

Breadth-first search can find a shortest distinguishing string.