# Automata and Languages

**Prof. Mohamed Hamada**

**Software Engineering Lab.**
**The University of Aizu**
**Japan**

## Today's Topics

- DFA Minimization
- Examples
- Minimization Algorithms



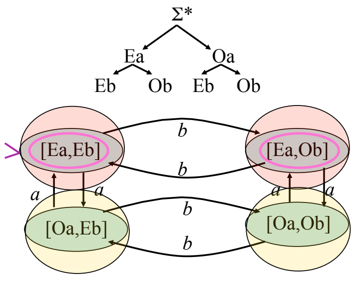
## DFA Minimization



DFA Minimization

**Example**

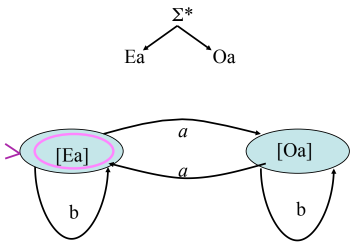
Strings over {*a*,*b*} with even number of *a*'s



## DFA Minimization

**Example**

Strings over {*a*,*b*} with even number of *a*' s




5

## DFA Minimization

**Observation**

- The states among the state sets {[Ea,Eb], [Ea,Ob]} and {[Oa,Eb], [Oa,Ob]} differ on aspect immaterial for the problem at hand.
- Why not collapse these state sets into one state each, to get a smaller DFA?

6

## DFA Minimization

**Definition**

### Equivalent or Indistinguishable States

- Recall that a DFA state summarizes the substring consumed so far (that is, the past history).
- Two states $q_i$ and $q_j$ are *equivalent* or (*indistinguishable*), if, when started in these states, every string causes the machine to either end up in a final state for both or end up in a non-accepting state for both.

**Two states $q_i$ and $q_k$ are *equivalent* (or *indistinguishable*), if for all strings $w \in \Sigma^*$**

$$\delta(q_i, w) \in F \Leftrightarrow \delta(q_k, w) \in F$$

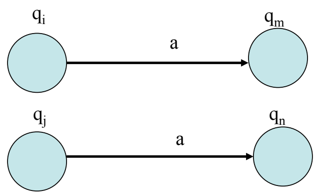
**Two states $q_i$ and $q_k$ are *distinguishable*, if for some string $w \in \Sigma^*$**

$$\delta(q_i, w) \in F \Leftrightarrow \delta(q_k, w) \notin F$$

7

## DFA Minimization

**Main Idea**



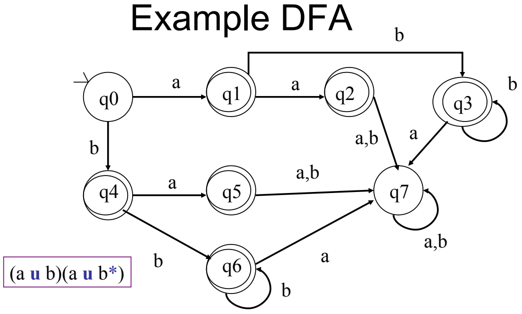

If $q_m$ and $q_n$ are distinguishable, then so are $q_i$ and $q_j$

8

DFA Minimization

**Example**

## Example DFA




9

DFA Minimization

**Example**

## Refinement of State Partitions

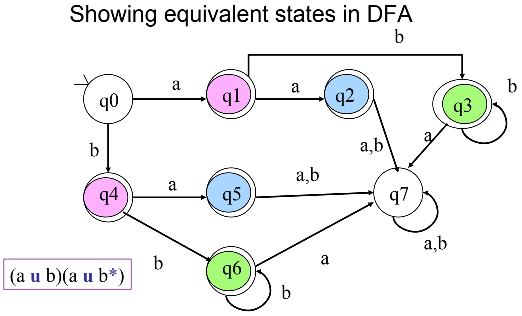
- { {q0,q7} ,  {q1,q2,q3,q4,q5,q6} }
- { {q0},{q7}, {q1,q2,q3,q4,q5,q6} }
  - On any transition
- { {q0},{q7}, {q1,q2,q3,q4,q5,q6} }
- { {q0},{q7}, {q1,q4}, {q2,q3,q5,q6} }
  - On “a” transition
- { {q0},{q7}, {q1,q4}, {q2,q5},{q3,q6} }
  - On “b” transition

10

DFA Minimization

**Example**

Showing equivalent states in DFA

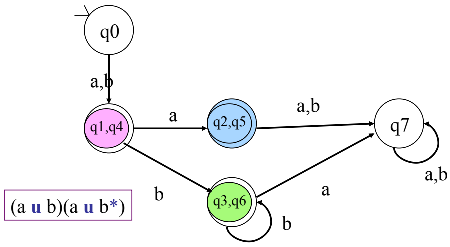



11

DFA Minimization

**Example**

## Minimum DFA




12

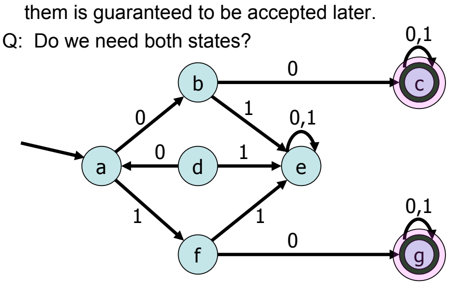
DFA Minimization

**Example**

## Equivalent States

Consider the accept states c and g. They are both sinks meaning that any string which ever reaches them is guaranteed to be accepted later.

Q: Do we need both states?




13

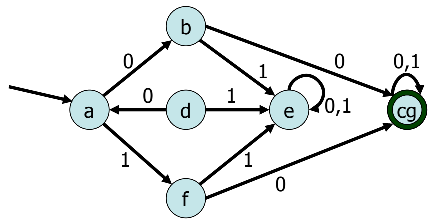
DFA Minimization

**Example**

## Joining Equivalent States

A: No, they can be unified as illustrated below.

Q: Can any other states be unified because any subsequent string suffixes produce identical results?




14

DFA Minimization

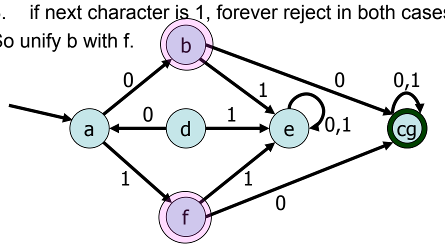
**Example**

## Equivalent States

A: Yes, b and f. Notice that if you' re in b or f then:

1. if string ends, reject in both cases
2. if next character is 0, forever accept in both cases
3. if next character is 1, forever reject in both cases

So unify b with f.




15

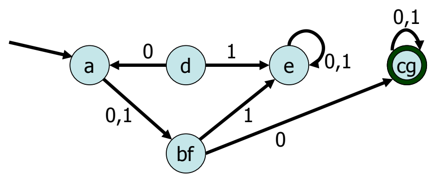
DFA Minimization

**Example**

## Joining Equivalent States

Intuitively two states are equivalent if all subsequent behavior from those states is the same.

Q: Come up with a formal characterization of state equivalence.



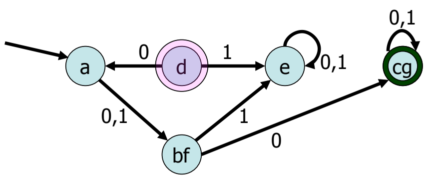

16

DFA Minimization

**Example**

Finishing the Example

Q: Any other ways to simplify the automaton?



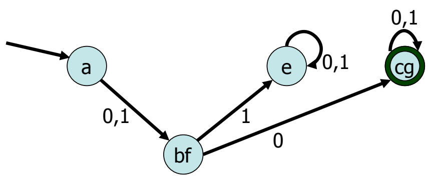

17

DFA Minimization

**Useless States**

A: Get rid of d.

Getting rid of unreachable ***useless states*** doesn't affect the accepted language.




18

DFA Minimization

**Definition**

An automaton is ***irreducible*** if

- it contains no useless states, and
- no two distinct states are equivalent.

**Goals of the Minimization Algorithm**

•The goal of minimization algorithm is to create irreducible automata from arbitrary ones.

•The algorithm actually produces smallest possible DFA for the given language, hence the name "minimization".

19

DFA Minimization

**Minimization Algorithm**

First Part: Partition

DFA minimize(DFA ($Q$, $\Sigma$, $\delta$, $q_0$, $F$ ) )
  remove any state $q$ unreachable from $q_0$
  Partition $P$ = {$F$, $Q$ - $F$ }
  boolean Consistent = false
  while( Consistent == false )
    Consistent = true
    for(every Set $S \in P$, symbol $a \in \Sigma$, Set $T \in P$ )
      Set temp = {$q \in T$ | $\delta(q,a) \in S$ }
      if (temp != Ø && temp != $T$ )
        Consistent = false
        $P$ = ($P$ − $T$ )∪{temp, $T$−temp}
  return defineMinimizor( ($Q$, $\Sigma$, $\delta$, $q_0$, $F$ ), $P$ )

20

DFA Minimization

**Minimization Algorithm**

Second Part: Minimization

DFA defineMinimizor (DFA ($Q$, $\Sigma$, $\delta$, $q_0$, $F$ ), Partition $P$ )

Set $Q' = P$

State $q'_0$ = the set in $P$ which contains $q_0$

$F' = \{ S \in P \mid S \subseteq F \}$

for (each $S \in P$, $a \in \Sigma$)

*define* $\delta'(S,a)$ = the set $T \in P$ which contains the states $\delta'(S,a)$

return ($Q'$, $\Sigma$, $\delta'$, $q'_0$, $F'$ )

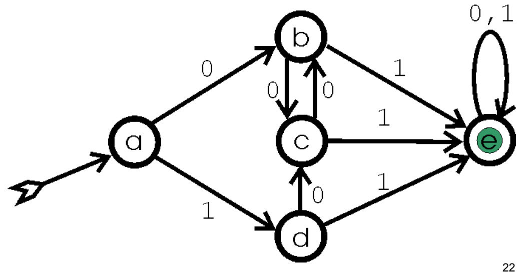
21

DFA Minimization

**Example**

Minimization Example

Start with a DFA



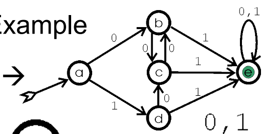

22

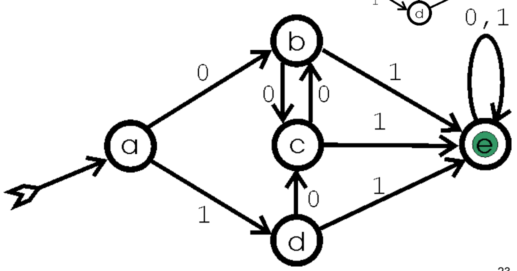
DFA Minimization

**Example**

Minimization Example

Miniature version →





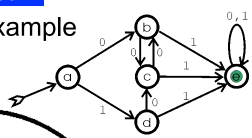

23

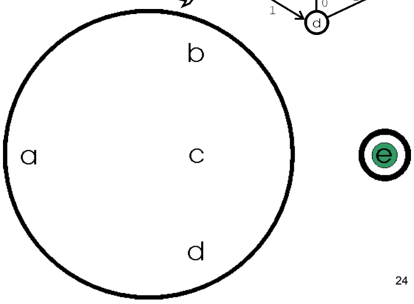
DFA Minimization

**Example**

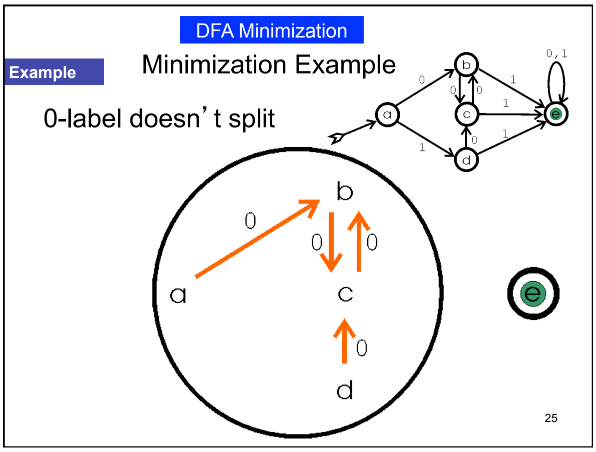
Minimization Example

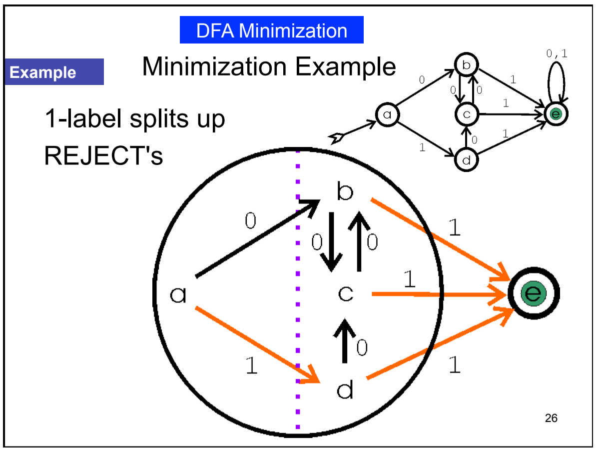
Split into two parts.
ACCEPT
vs.
REJECT

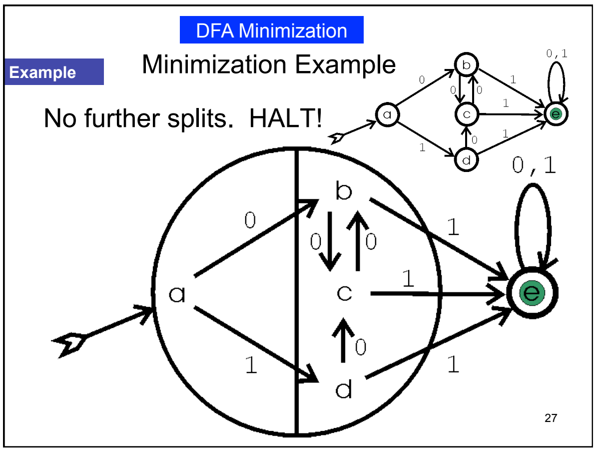


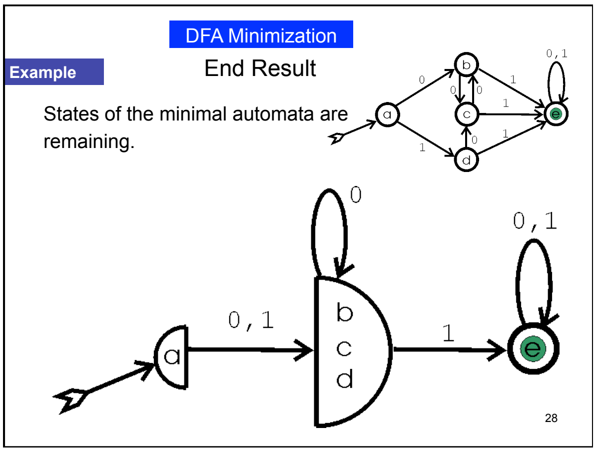



24


DFA Minimization
Example
Minimization Example
0-label doesn't split
0
0
0
0
a
b
c
d
e
25


DFA Minimization
Example
Minimization Example
1-label splits up
REJECT's
0
0
0
0
1
1
1
1
a
b
c
d
e
26

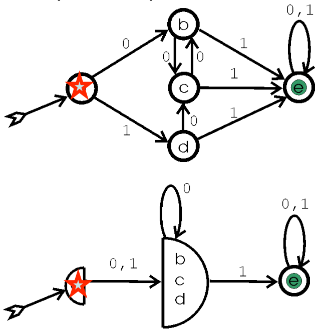

DFA Minimization
Example
Minimization Example
No further splits.  HALT!
0,1
0
0
0
0
1
1
1
1
a
b
c
d
e
27


DFA Minimization
Example
End Result
States of the minimal automata are remaining.
0
0,1
0,1
1
a
b
c
d
e
28

DFA Minimization

**Example** Minimization Example. Compare

100100101

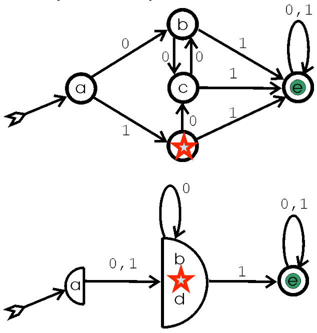



29

DFA Minimization

**Example** Minimization Example. Compare

100100101

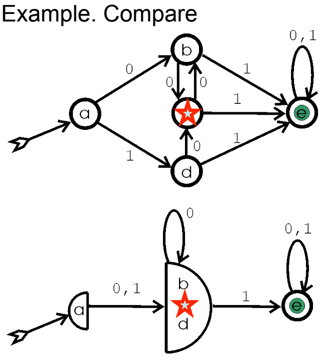



30

DFA Minimization

**Example** Minimization Example. Compare

100100101

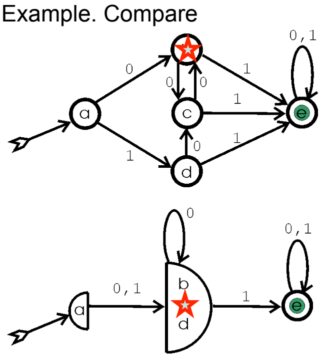



31

DFA Minimization

**Example** Minimization Example. Compare

100100101

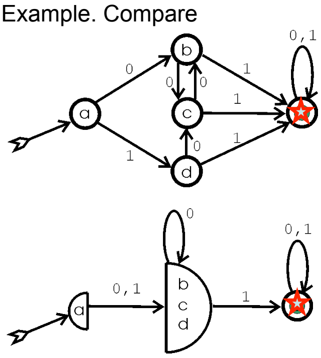



32

## DFA Minimization

**Example** Minimization Example. Compare

100100101

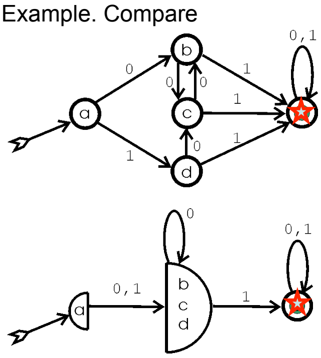



33

## DFA Minimization

**Example** Minimization Example. Compare

100100101

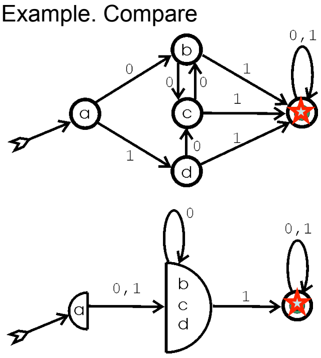



34

## DFA Minimization

**Example** Minimization Example. Compare

100100101

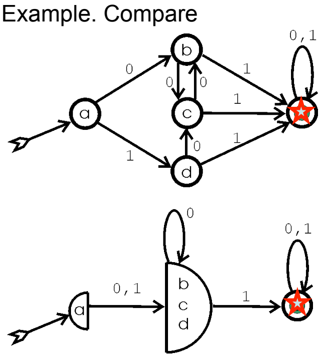



35

## DFA Minimization

**Example** Minimization Example. Compare

100100101

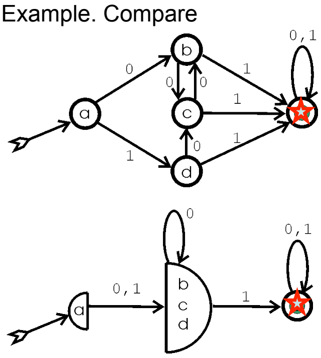



36

DFA Minimization

**Example** Minimization Example. Compare

100100101



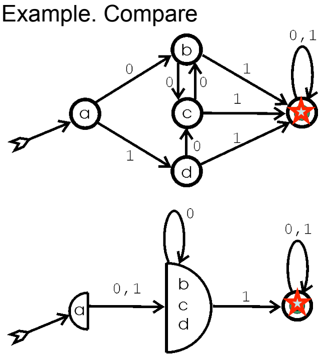

37

DFA Minimization

**Example** Minimization Example. Compare

100100101

ACCEPTED.

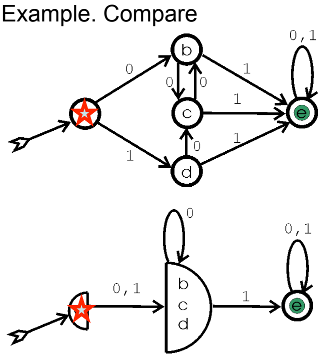



38

DFA Minimization

**Example** Minimization Example. Compare

10000

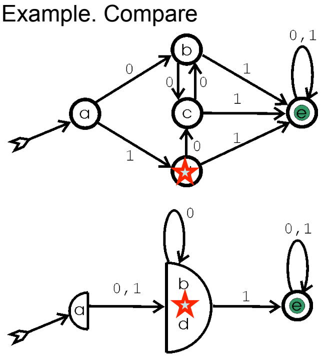



39

DFA Minimization

**Example** Minimization Example. Compare

10000

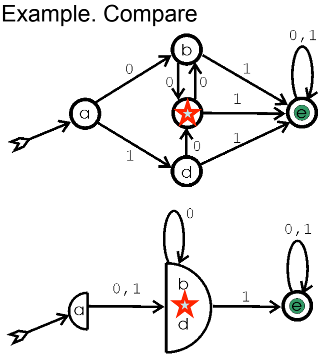



40

DFA Minimization

**Example** Minimization Example. Compare

10000

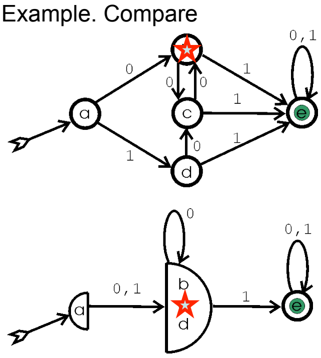



41

DFA Minimization

**Example** Minimization Example. Compare

10000

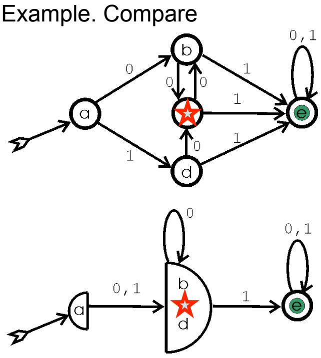



42

DFA Minimization

**Example** Minimization Example. Compare

10000



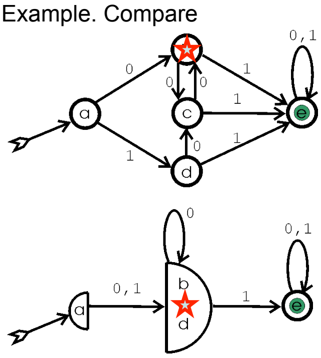

43

DFA Minimization

**Example** Minimization Example. Compare

10000

REJECT.




44

DFA Minimization

**Minimal Automaton**

Previous algorithm guaranteed to produce an irreducible FA. Why should that FA be the smallest possible FA for its accepted language?

45

DFA Minimization

## Minimal Automaton

**Theorem** (Myhill-Nerode): The minimization algorithm produces the smallest possible automaton for its accepted language.

**Proof** Show that any irreducible automaton is the smallest for its accepted language *L*:

We say that two strings $u,v \in \Sigma^*$ are ***indistinguishable*** if for all suffixes *x, ux* is in *L* exactly when *vx* is.

Notice that if *u* and *v* *are* distinguishable, the path from their paths from the start state must have different endpoints.

46

DFA Minimization

**Proof (cont.)**

Consequently, the number of states in any DFA for *L* must be as great as the number of mutually distinguishable strings for *L*.

But an irreducible DFA has the property that every state gives rise to another mutually distinguishable string!

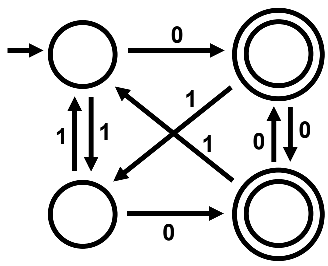
Therefore, any other DFA must have at least as many states as the irreducible DFA

47

DFA Minimization

**Quiz 1**

IS THIS DFA **MINIMAL?**



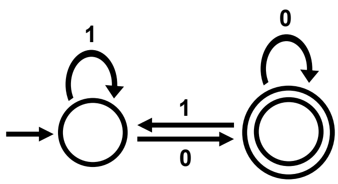

**NO**

48

DFA Minimization

Quiz 1

ITS **MINIMAL DFA IS**

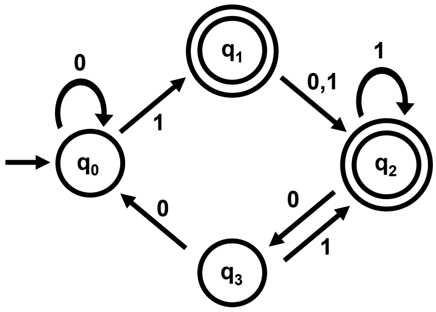



49

DFA Minimization

Quiz 2

IS THIS DFA **MINIMAL?**




50

DFA Minimization

Definition

For a DFA M = (Q, Σ, δ, $q_0$, F), let p, q, r ∈ Q

Definition:

p ~ q iff p is indistinguishable (equivalent) from q

p ≁ q iff p is distinguishable from q

Proposition: ~ is an equivalence relation

p ~ p (reflexive)

p ~ q ⇒ q ~ p (symmetric)

p ~ q and q ~ r ⇒ p ~ r (transitive)

51

DFA Minimization

Definition

For a DFA M = (Q, Σ, δ, $q_0$, F), let p, q, r ∈ Q

Definition:

p ~ q iff p is indistinguishable from q

p ≁ q iff p is distinguishable from q

Proposition: ~ is an equivalence relation

p ~ p (reflexive)

p ~ q ⇒ q ~ p (symmetric)

p ~ q and q ~ r ⇒ p ~ r (transitive)

52

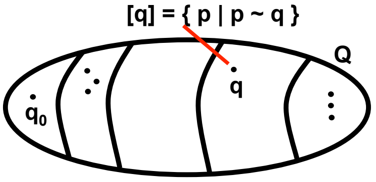
## DFA Minimization

**Definition**

**Because ~ is an equivalence relation**

**so ~ partitions the set of states of M into disjoint equivalence classes**

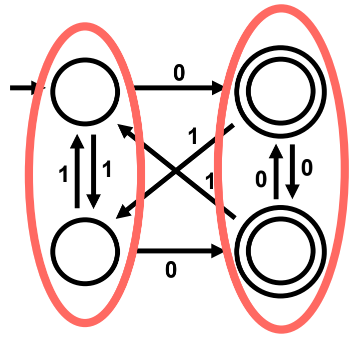
**[q] = { p | p ~ q }**




53

## DFA Minimization

**Example**




54

## DFA Minimization

**Algorithm**

**Algorithm MINIMIZE**
**Input: DFA M**
**Output: DFA $M_{MIN}$ such that:**

$M \equiv M_{MIN}$

**$M_{MIN}$ has no inaccessible states**

**$M_{MIN}$ is irreducible**

||

**states of $M_{MIN}$ are pairwise distinguishable**

**Theorem: $M_{MIN}$ is the unique minimum**

**Idea: States of $M_{MIN}$ will be blocks of equivalent states of M**

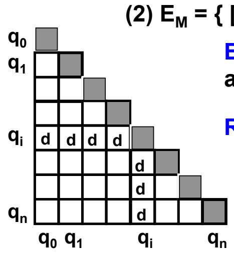
55

## DFA Minimization

**Algorithm**

### TABLE-FILLING ALGORITHM

**Input: DFA M = (Q, Σ, δ, $q_0$, F)**
**Output: (1) $D_M$ = { (p,q) | p,q ∈ Q and p $\not\sim$ q }**
**(2) $E_M$ = { [q] | q ∈ Q }**




**Base Case: p accepts and q rejects ⇒ p $\not\sim$ q**

**Recursion:**

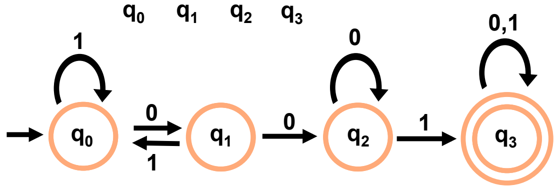
$p \xrightarrow{a} p'$
$\not\sim$ ⇒ p $\not\sim$ q
$q \xrightarrow{a} q'$

56

Example

## DFA Minimization

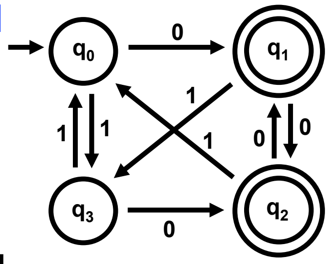
| | | | | |
|---|---|---|---|---|
| $q_0$ | | | | |
| $q_1$ | d | | | |
| $q_2$ | d | d | | |
| $q_3$ | d | d | d | |
| | $q_0$ | $q_1$ | $q_2$ | $q_3$ |




57

## DFA Minimization

Example




| | | | | |
|---|---|---|---|---|
| $q_0$ | | | | |
| $q_1$ | d | | | |
| $q_2$ | d | | | |
| $q_3$ | | d | d | |
| | $q_0$ | $q_1$ | $q_2$ | $q_3$ |

58

Algorithm

## DFA Minimization

**Algorithm MINIMIZE**

**Input: DFA M**

**Output: DFA $M_{MIN}$**

**(1) Remove all inaccessible states from M**

**(2) Apply Table-Filling algorithm to get**
**$E_M$ = { [q] | q is an accessible state of M }**

$$M_{MIN} = (Q_{MIN}, \Sigma, \delta_{MIN}, q_{0\ MIN}, F_{MIN})$$

$$Q_{MIN} = E_M,\ \ q_{0\ MIN} = [q_0],\ \ F_{MIN} = \{ [q] \mid q \in F \}$$

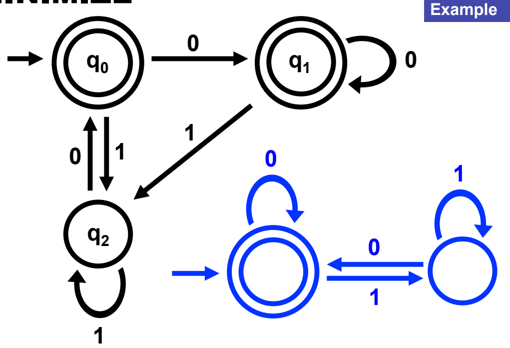
$$\delta_{MIN}( [q],\sigma ) = [\ \delta( q,\sigma )\ ]$$

59

**MINIMIZE**

## DFA Minimization

Example

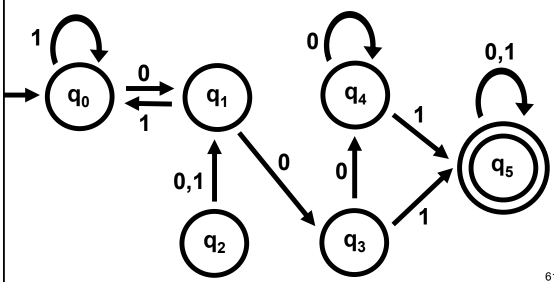



60

## DFA Minimization

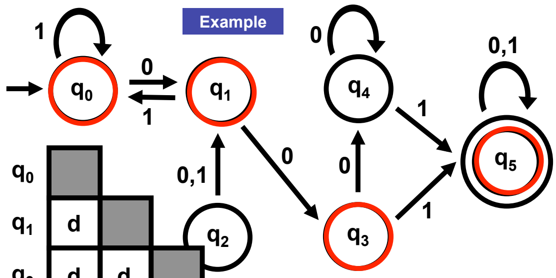
**Exercise**

**Minimize the following DFA?**






**Example**




| | $q_0$ | $q_1$ | $q_3$ | $q_4$ | $q_5$ |
|---|---|---|---|---|---|
| $q_0$ | | | | | |
| $q_1$ | d | | | | |
| $q_3$ | d | d | | | |
| $q_4$ | d | d | | | |
| $q_5$ | d | d | d | d | |