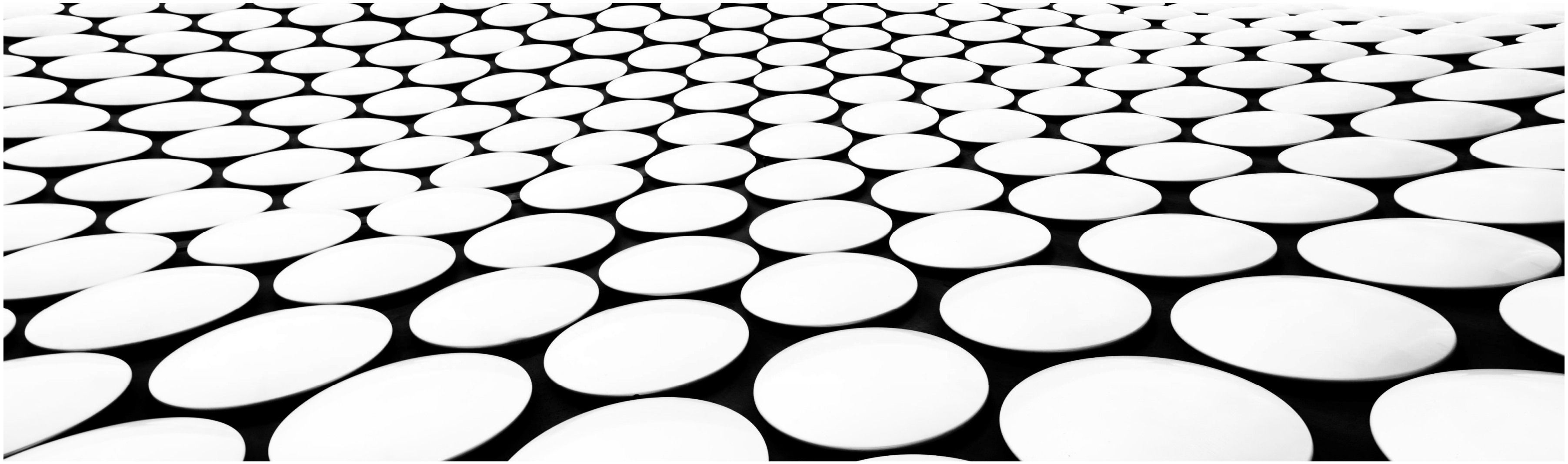
DETERMINISTIC /NON DETERMINISTIC AUTOMATA

IF 2124 TEORI BAHASA FORMAL OTOMATA

Judhi S.



Deterministic Finite Automata

A DFA is a quintuple

$$A = (Q, \Sigma, \delta, q_0, F)$$

- Q is a finite set of *states*
- Σ is a *finite alphabet* (=input symbols)
- δ is a *transition function* $(q, a) \mapsto p$
- $q_0 \in Q$ is the *start state*
- $F \subseteq Q$ is a set of *final states*

Example: An automaton A that accepts

$L = \{x01y : x, y \in \{0,1\}^*\}$

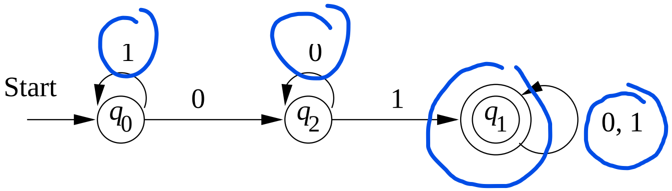
x 01 y .

The automaton $A = (\{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_1\})$
as a *transition table*:

	0	1
$\rightarrow q_0$	q_2	q_0
$\star q_1$	q_1	q_1
q_2	q_2	q_1

Sama .

The automaton as a *transition diagram*:



01

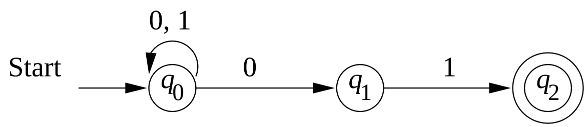
An FA *accepts* a string $w = a_1 a_2 \cdots a_n$ if there is a path in the transition diagram that

1. Begins at a start state

2. Ends at an accepting state

3. Has sequence of labels $a_1 a_2 \cdots a_n$

Example: The FA



accepts e.g. the string 01101

• The transition function δ can be extended to $\hat{\delta}$ that operates on states and strings (as opposed to states and symbols)

δ : symbol satu saja yg dibaca

$\hat{\delta}$: string.

Basis: $\hat{\delta}(q, \epsilon) = q$

Induction: $\hat{\delta}(q, xa) = \delta(\hat{\delta}(q, x), \boxed{a})$

$w : \underbrace{x}_{\text{string}} a$

a symbol.

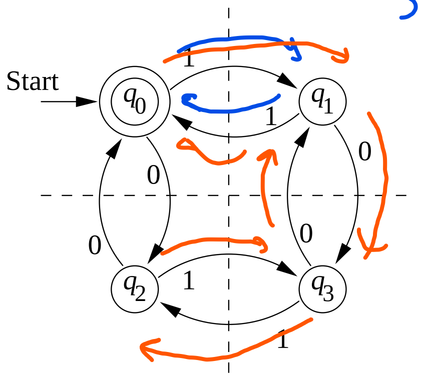
• Now, fomally, the *language accepted by* A is

$$L(A) = \{w : \hat{\delta}(q_0, w) \in F\}$$

$\hat{\delta}(q_0, w) \in F$ (sampai ke final state).

• The languages accepted by FA:s are called *regular languages*

Example: DFA accepting all and only strings with an even number of 0's and an even number of 1's

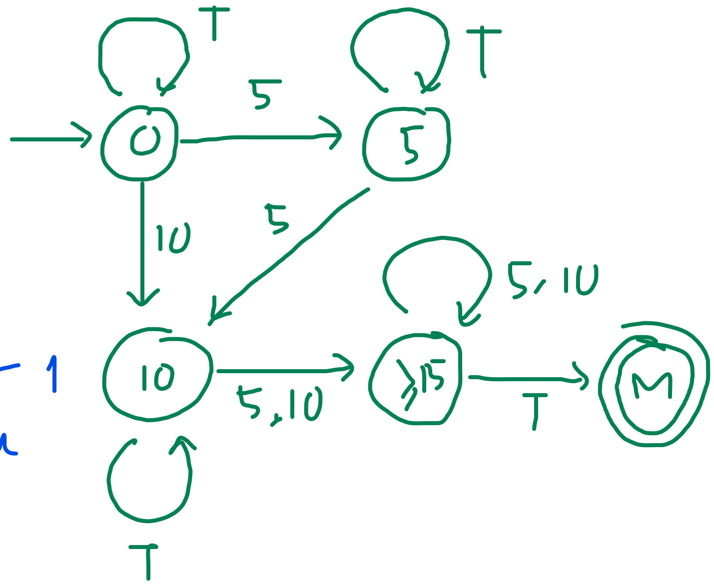


Selalu memuat jumlah 0 ~~dan~~ 1 atau genap

11

101101

dst .

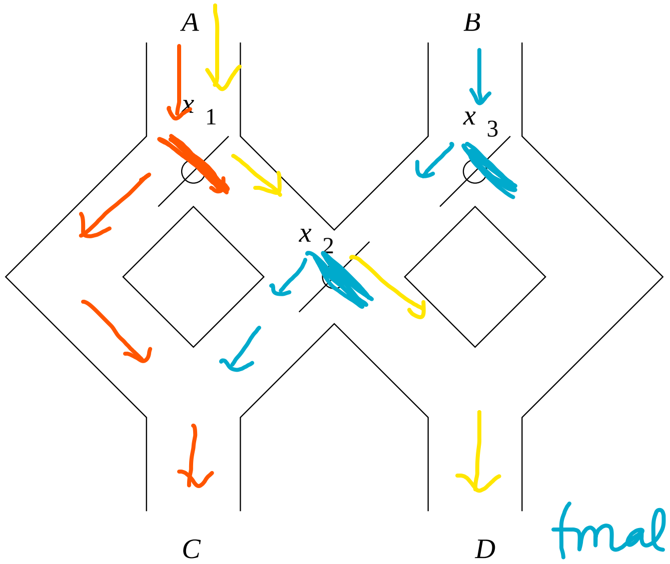


Tabular representation of the Automaton

	0	1
$\star \rightarrow q_0$	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

Example

Marble-rolling toy from p. 53 of textbook



final state

r. reject a accept

A state is represented as sequence of three bits followed by r or a (previous input *rejected* or *accepted*)

For instance, $010a$, means
left, right, left, accepted

Tabular representation of DFA for the toy

	A	B
$\to 000r$	$100r$	$011r$
$\star 000a$	$100r$	$011r$
$\star 001a$	$101r$	$000a$
$010r$	$110r$	$001a$
$\star 010a$	$110r$	$001a$
$011r$	$111r$	$010a$
$100r$	$010r$	$111r$
$\star 100a$	$010r$	$111r$
$101r$	$011r$	$100a$
$\star 101a$	$011r$	$100a$
$110r$	$000a$	$101a$
$\star 110a$	$000a$	$101a$
$111r$	$001a$	$110a$

a

C

D

$000r \to 100r$
$000r \to 011r$

*

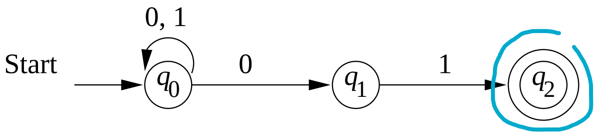
det

Nondeterministic Finite Automata

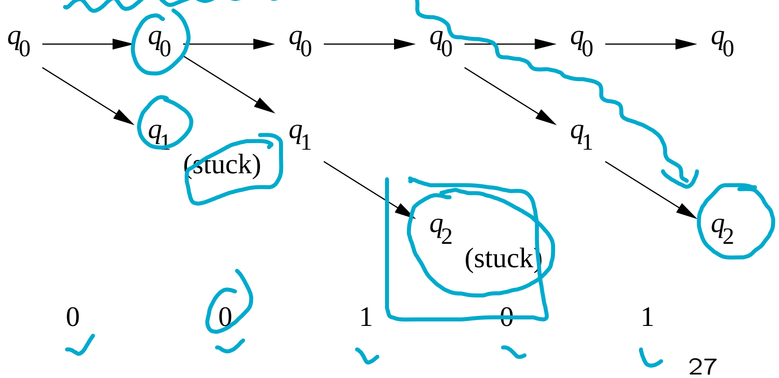
A NFA can be in several states at once, or, viewded another way, it can "guess" which state to go to next

Example: An automaton that accepts all and only strings ending in 01.

$w = x01$ x : apa saja.



Here is what happens when the NFA processes the input 00101



00101

Formally, a NFA is a quintuple

$$A = (Q, \Sigma, \delta, q_0, F)$$

- Q is a finite set of states

- Σ is a finite alphabet

$\delta(q, 0)$

- δ is a transition function from $Q \times \Sigma$ to the *powerset* of Q $2^{|Q|}$

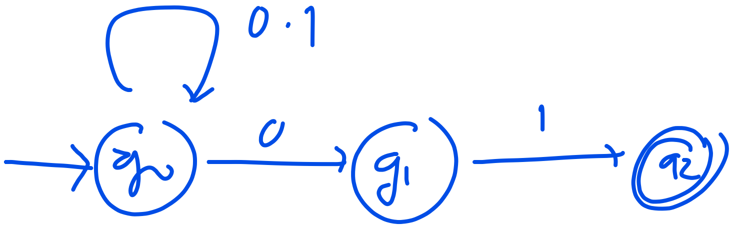
$\{q_0, q_1\}$ $\{\ \}$ $\{q_0\}$, $\{q_1\}$, $\underline{\underline{\{q_0, q_1\}}}$

- $q_0 \in Q$ is the *start state*

- $F \subseteq Q$ is a set of *final states*

Example: The NFA from the previous slide is

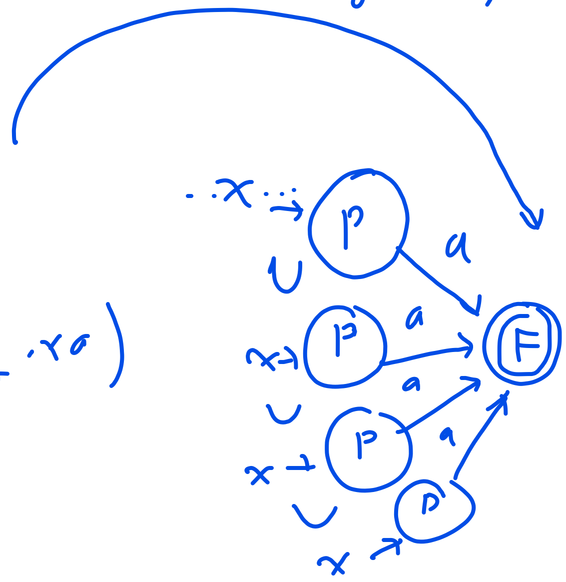
$$(\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\})$$



where δ is the transition function

	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\emptyset$	$\{q_2\}$
$\star q_2$	$\emptyset$	$\emptyset$

$\hat{\delta}(q, xa)$



Extended transition function $\widehat{\delta}$.

Basis: $\widehat{\delta}(q, \epsilon) = \{q\}$

Induction:

$w = xa$

$$\widehat{\delta}(q, xa) = \bigcup_{p \in \widehat{\delta}(q,x)} \delta(p, a)$$

$\hat{\delta}(q, xa)$

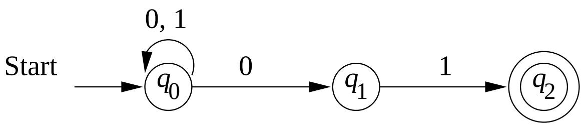
Example: Let's compute $\widehat{\delta}(q_0, 00101)$ on the blackboard

1. $\hat{\delta}(q_0, \varepsilon) = \{q_0\}$
2. $\hat{\delta}(q_0, 0) = \delta(q_0, 0) = \{q_0, q_1\}$
3. $\hat{\delta}(q_0, 00) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$
4. $\hat{\delta}(q_0, 001) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}$

- Now, fomally, the *language accepted by* A is

$$L(A) = \{w : \widehat{\delta}(q_0, w) \cap F \neq \emptyset\}$$

Let's prove formally that the NFA



accepts the language $\{x01 : x \in \Sigma^*\}$. We'll do a mutual induction on the three statements below

0. $w \in \Sigma^* \Rightarrow q_0 \in \hat{\delta}(q_0, w)$

1. $q_1 \in \hat{\delta}(q_0, w) \Leftrightarrow w = x0$

2. $q_2 \in \hat{\delta}(q_0, w) \Leftrightarrow w = x01$

5) $\hat{\delta}(q_0, 0010) = \delta(q_0, 0) \cup \delta(q_2, 0) = \{q_0, q_1\}$
$\cup\ \phi = \{q_0, q_1\}$

6) $\hat{\delta}(q_0, 00101) = \delta(q_0, 1) \cup \delta(q_1, 1)$
$= \{q_0\} \cup \{q_2\}$
$= \{q_0, q_2\}$

$\{q_0, q_2\} \cap F \neq \phi$

Basis: If $|w| = 0$ then $w = \epsilon$. Then statement (0) follows from def. For (1) and (2) both sides are false for ϵ

Induction: Assume $w = xa$, where $a \in \{0, 1\}$, $|x| = n$ and statements (0)–(2) hold for x. We will show on the blackboard in class that the statements hold for xa.

Equivalence of DFA and NFA

- NFA's are usually easier to "program" in.

- Surprisingly, for any NFA N there is a DFA D, such that $L(D) = L(N)$, and vice versa.

- This involves the *subset construction*, an important example how an automaton B can be generically constructed from another automaton A.

- Given an NFA

$$N = (Q_N, \Sigma, \delta_N, q_0, F_N)$$

we will construct a DFA

$$D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$$

such that

$$L(D) = L(N)$$

.

The details of the subset construction:

- $Q_D = \{S : S \subseteq Q_N\}$.

Note: $|Q_D| = 2^{|Q_N|}$, although most states in Q_D are likely to be garbage.

- $F_D = \{S \subseteq Q_N : S \cap F_N \neq \emptyset\}$

- For every $S \subseteq Q_N$ and $a \in \Sigma$,

$$\delta_D(S, a) = \bigcup_{p \in S} \delta_N(p, a)$$

Let's construct δ_D from the NFA on slide 27

	0	1
$\emptyset$	$\emptyset$	$\emptyset$
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	$\emptyset$	$\{q_2\}$
$\star\{q_2\}$	$\emptyset$	$\emptyset$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$\star\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\star\{q_1, q_2\}$	$\emptyset$	$\{q_2\}$
$\star\{q_0, q_1, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$

Note: The states of D correspond to subsets of states of N, but we could have denoted the states of D by, say, $A - F$ just as well.

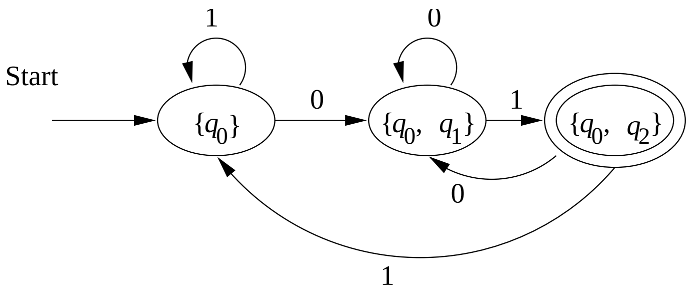
	0	1
A	A	A
$\rightarrow B$	E	B
C	A	D
$\star D$	A	A
E	E	F
$\star F$	E	B
$\star G$	A	D
$\star H$	E	F

We can often avoid the exponential blow-up by constructing the transition table for D only for accessible states S as follows:

Basis: $S = \{q_0\}$ is accessible in D

Induction: If state S is accessible, so are the states in $\bigcup_{a \in \Sigma} \delta_D(S, a)$.

Example: The "subset" DFA with accessible states only.



Theorem 2.11: Let D be the "subset" DFA of an NFA N. Then $L(D) = L(N)$.

Proof: First we show on an induction on $|w|$ that

$$\widehat{\delta}_D(\{q_0\}, w) = \widehat{\delta}_N(q_0, w)$$

Basis: $w = \epsilon$. The claim follows from def.

Induction:

$$\begin{aligned}
\widehat{\delta}_D(\{q_0\}, xa) &\stackrel{\text{def}}{=} \delta_D(\widehat{\delta}_D(\{q_0\}, x), a) \\
&\stackrel{\text{i.h.}}{=} \delta_D(\widehat{\delta}_N(q_0, x), a) \\
&\stackrel{\text{cst}}{=} \bigcup_{p \in \widehat{\delta}_N(q_0, x)} \delta_N(p, a) \\
&\stackrel{\text{def}}{=} \widehat{\delta}_N(q_0, xa)
\end{aligned}$$

Now (**why?**) it follows that $L(D) = L(N)$.

Theorem 2.12: A language L is accepted by some DFA if and only if L is accepted by some NFA.

Proof: The "if" part is Theorem 2.11.

For the "only if" part we note that any DFA can be converted to an equivalent NFA by modifying the δ_D to δ_N by the rule

- If $\delta_D(q, a) = p$, then $\delta_N(q, a) = \{p\}$.

By induction on $|w|$ it will be shown in the tutorial that if $\widehat{\delta}_D(q_0, w) = p$, then $\widehat{\delta}_N(q_0, w) = \{p\}$.

The claim of the theorem follows.