WEEK 2: LECTURE NOTES

Non-deterministic Finite Automata (NFA)

A 5-tuple $(Q, \Sigma, \delta, q_0, F)$

Q : a finite set of states

Σ : a finite input alphabet

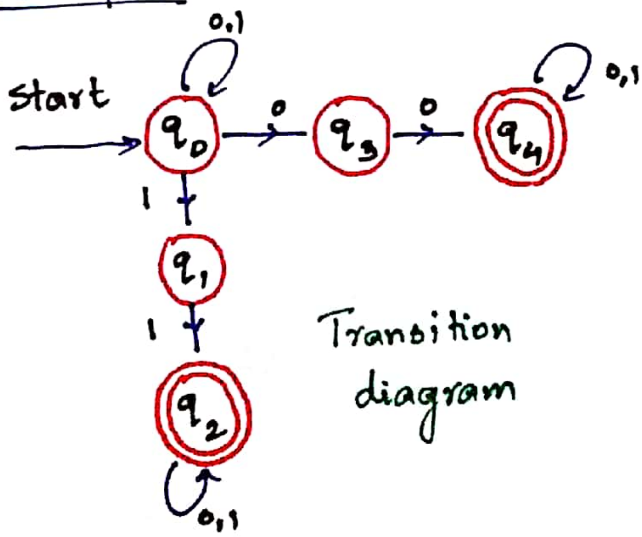
$\delta : Q \times \Sigma \longrightarrow 2^Q$; the transition function

(power set of Q, i.e. set of all subsets of Q)

$q_0 \in Q$: the initial state

$F \subseteq Q$: the set of final/accepting states

Example:



Transition diagram

Transition Table

δ	0	1
$\rightarrow q_0$	$\{q_0, q_3\}$	$\{q_0, q_1\}$
q_1	ϕ	$\{q_2\}$
$*q_2$	$\{q_2\}$	$\{q_2\}$
q_3	$\{q_4\}$	ϕ
$*q_4$	$\{q_4\}$	$\{q_4\}$

Note: Difference between DFA and NFA

→ transition function returns

↳ a single state for DFA

↳ a set of states for NFA

The extended transition function

$\delta : Q \times \Sigma \to 2^Q$: transition function for NFA

$\hat{\delta} : Q \times \Sigma^* \to 2^Q$: extended transition function for NFA, formally defined as follows:

(i) $\hat{\delta}(q, \varepsilon) = q$

(ii) Let $w = xa$, $x \in \Sigma^*$, $a \in \Sigma$

Let $\hat{\delta}(q, x) = \{p_1, p_2, \ldots, p_k\}$

Let $\bigcup_{i=1}^{k} \delta(p_i, a) = \{r_1, r_2, \ldots, r_m\}$

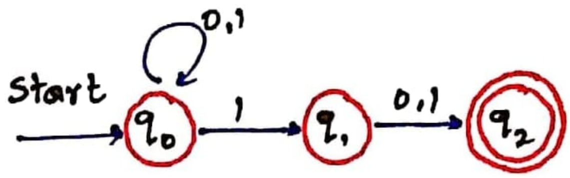
Then

$$\begin{aligned}
\hat{\delta}(q, w) &= \hat{\delta}(\hat{\delta}(q, x), a) \\
&= \hat{\delta}(\{p_1, p_2, \ldots, p_k\}, a) \\
&= \bigcup_{i=1}^{k} \delta(p_i, a) \\
&= \{r_1, r_2, \ldots, r_m\}
\end{aligned}$$

i.e. To compute $\hat{\delta}(q, w)$ where $w = xa$, we first compute $\hat{\delta}(q, x)$

Example

δ	0	1
q_0	$\{q_0\}$	$\{q_0, q_1\}$
q_1	$\{q_2\}$	$\{q_2\}$
q_2	ϕ	ϕ



- The above NFA accepts all binary strings which has second last symbol as 1
- q_2 has no transition and hence it dies
- $\hat{\delta}(01010) = ?$

$\hat{\delta}(q_0, \varepsilon) = \{q_0\}$

$\hat{\delta}(q_0, 0) = \{q_0\}$

$\hat{\delta}(q_0, 01) = \hat{\delta}(\{q_0\}, 1) = \{q_0, q_1\}$

$$\begin{aligned}\hat{\delta}(q_0, 010) = \hat{\delta}(\{q_0, q_1\}, 0) &= \delta(q_0, 0) \cup \delta(q_1, 0) \\ &= \{q_0\} \cup \{q_2\} \\ &= \{q_0, q_2\}\end{aligned}$$

$$\begin{aligned}\hat{\delta}(q_0, 0101) = \hat{\delta}(\{q_0, q_2\}, 1) &= \delta(q_0, 1) \cup \delta(q_2, 1) \\ &= \{q_0, q_1\} \cup \phi \\ &= \{q_0, q_1\}\end{aligned}$$

$\hat{\delta}(q_0, 01010) = \hat{\delta}(\{q_0, q_1\}, 0) = \{q_0, q_2\}$

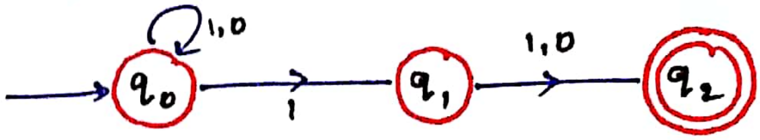
$\hat{\delta}(q_0, 010101) = \hat{\delta}(\{q_0, q_2\}, 1) = \{q_0, q_1\}$

The language of an NFA

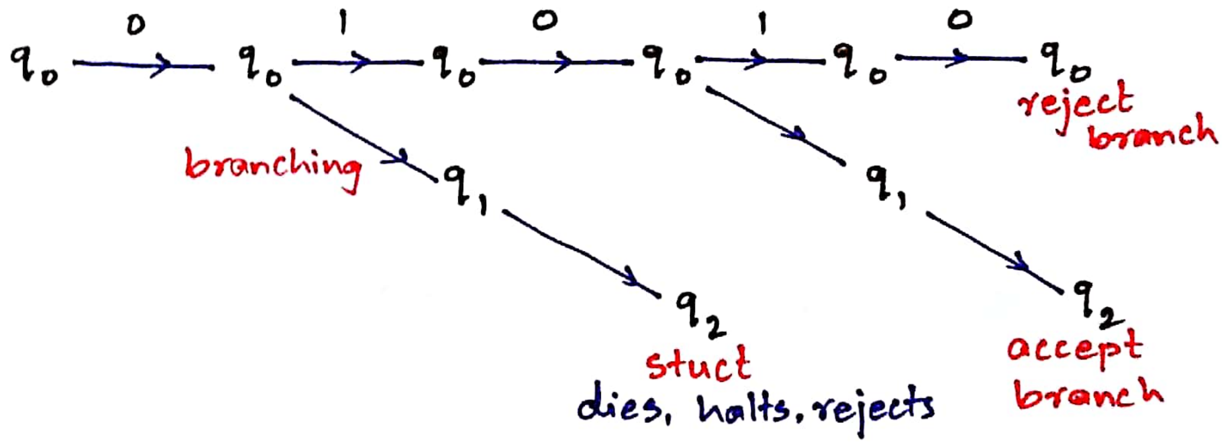
- NFA : $A = (Q, \Sigma, \delta, q_0, F)$
- $L(A) = \{ w \mid \hat{\delta}(q_0, w) \cap F \neq \phi \}$
  - language accepted by NFA A.

Computation Tree

Consider the NFA accepting all binary strings which has 1 in its second last position



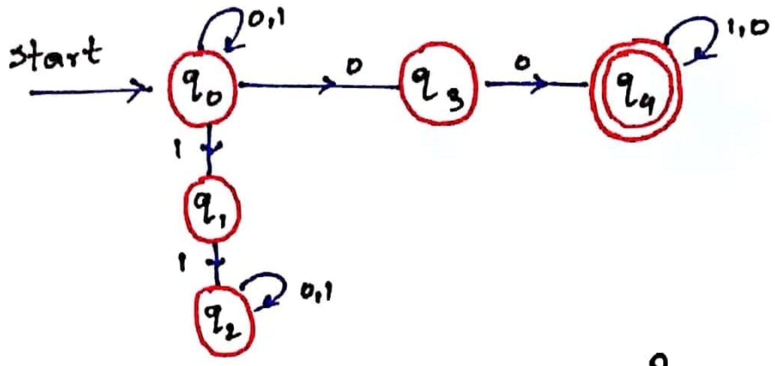
$w = 01010$



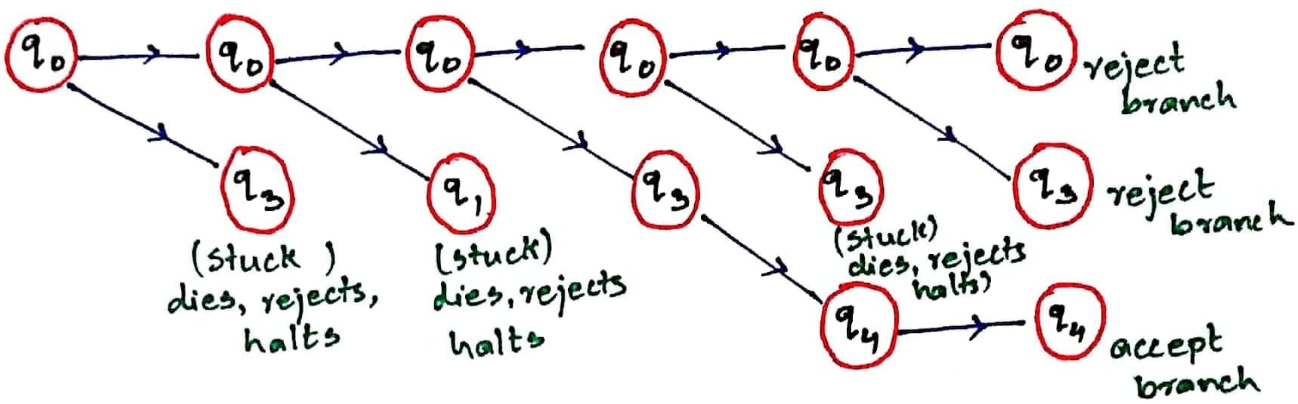
- Non-determinism : guess and verify
  - make as many guess as it likes but it must chech them
- w is accepted by NFA : computation tree has one accepting state
- w is rejected by NFA : every branch of the computatio tree must reject

Scanned by CamScanner

Example: (Non determinism as a computation)



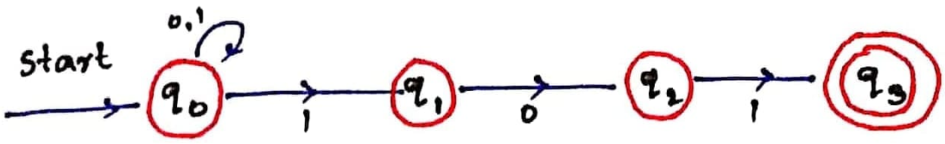
input : 01001 accepted or not ?



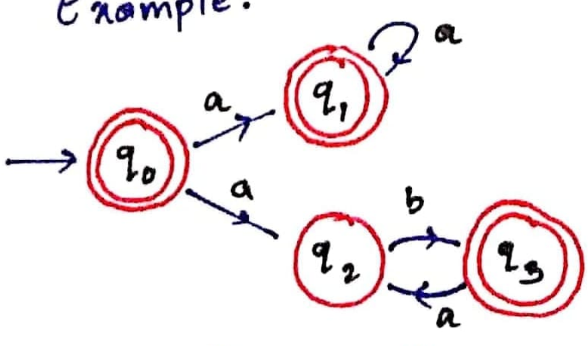
Building NFA

- It is easier compared to building a DFA

Example: NFA accepting all binary strings that end with pattern 101

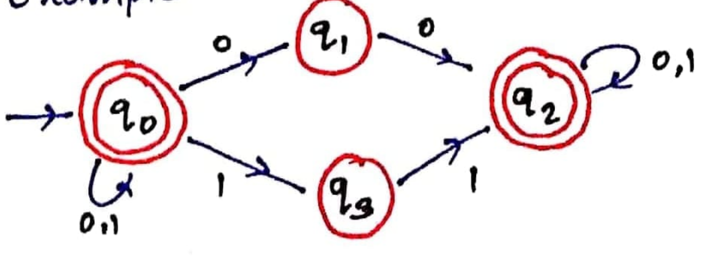


Example:



$a^* + (ab)^*$

Example



$x_1 00 y_1 + x_2 11 y_2$, $x_1, x_2, y_1, y_2 \in \{0,1\}^*$

i.e. strings that contain 00 or 11

The equivalence of DFA's and NFA's

- DFA can be treated as NFA
- language of an NFA is also a language of same DFA
  - i.e. NFA accepts only regular languages
  - i.e. for every NFA, we can construct an equivalent DFA (accepting the same language)

Subset Construction

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

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

such that

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

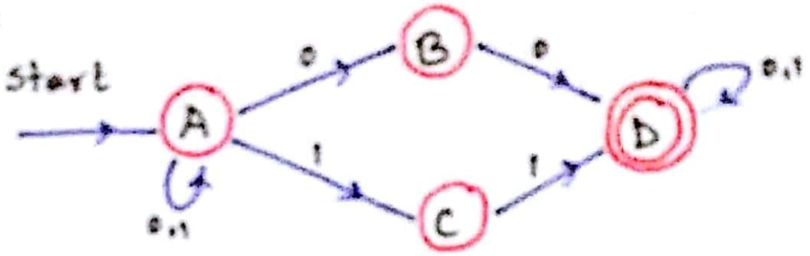
- input alphabet of N, D are the same (Σ)
- start state of D is the singleton set consisting of start state of N
- $Q_D = 2^{Q_N}$ i.e. if N has n states, D has 2^n states
  - We may throw away states that are not accessible from the initial state $\{q_0\}$ of D : $Q_D \subseteq 2^{Q_N}$
- $F_D = \{ S \in Q_D \mid S \cap F_N \neq \phi \}$

  i.e. all sets of N's states that include at least one accepting state of N

$$\delta_D(S, a) = \bigcup_{p \in S} \delta_N(p, a) \quad \text{i.e.} \quad S \subseteq Q_N, \; S \in Q_D$$

Example (Conversion from NFA to DFA)

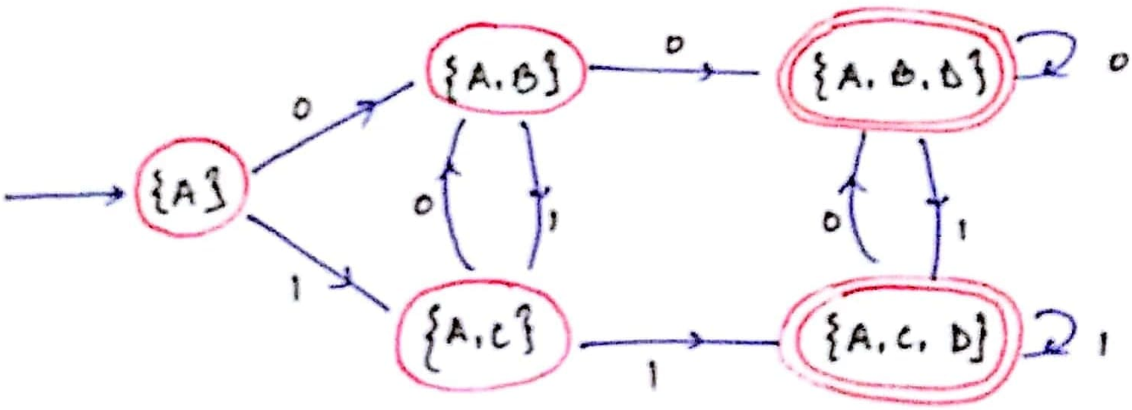
NFA:



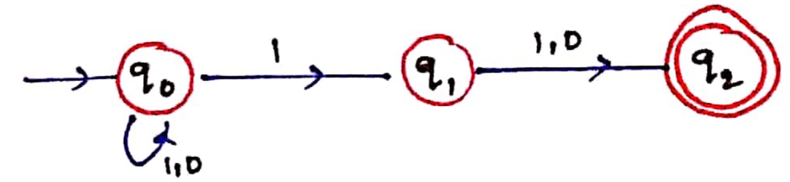
Equivalent DFA

- consider states that are reachable from initial state i.e. subsets of 2^{Q_N} containing A

	0	1
→ {A}	{A, B}	{A, C}
{A, B}	{A, B, D}	{A, C}
{A, C}	{A, B}	{A, C, D}
* {A, B, D}	{A, B, D}	{A, C, D}
* {A, C, D}	{A, B, D}	{A, C, D}



Example (NFA to DFA conversion)

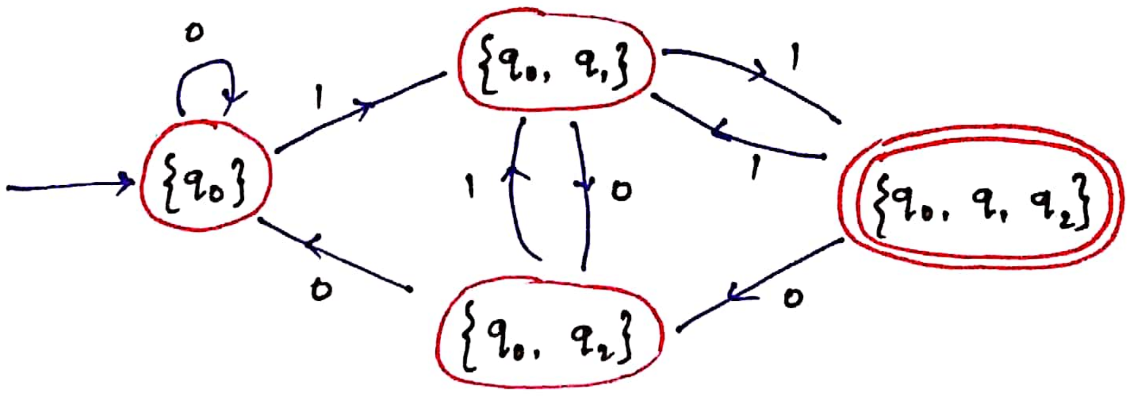


NFA: $N \Rightarrow (Q_N, \Sigma, \delta_N, q_0, F_N = \{q_2\})$

DFA: $D \Rightarrow (Q_D \subseteq 2^{Q_N}, \Sigma, \delta_D, \{q_0\}, F_D \subseteq Q_D)$

contains q_2

δ_D	0	1
$\{q_0\}$	$\{q_0\}$	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$
$\{q_0, q_2\}$	$\{q_0\}$	$\{q_0, q_1\}$
$\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$	$\{q_0, q_1\}$



Theorem:

If $D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ is the DFA constructed from the NFA $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ by the subset construction, then $L(D) = L(N)$

Proof:

We prove by induction on $|w|$ that

claim : $\hat{\delta}_D(\{q_0\}, w) = \hat{\delta}_N(q_0, w)$ for $w \in \Sigma^*$

Note that subset construction gives

$$F_D = \{ S \subseteq Q_D \mid S \cap F_N \neq \phi \}$$

when $Q_D \in 2^{Q_N}$

Also $\hat{\delta}$ returns a set of states from Q_N

$\hat{\delta}_D$ — a single state of Q_D

$\hat{\delta}_N$ — a set of states from Q_N

- D accepts w iff $\hat{\delta}_D(\{q_0\}, w) \in F_D$

  $\hat{\delta}_D(\{q_0\}, w) \cap F_N \neq \phi$

- N accepts w iff $\hat{\delta}_N(q_0, w) \cap F_N \neq \phi$

$\Rightarrow L(D) = L(N)$

Proof of claim (by induction on $|w|$)

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

Base: $|w| = 0$ i.e. $w = \varepsilon$

$$\hat{\delta}_D(\{q_0\}, \varepsilon) = \{q_0\}$$

$$\hat{\delta}_N(q_0, \varepsilon) = q_0$$

Induction:

$w = xa$, $|w| = n+1$, $|x| = n$

$w, x \in \Sigma^*$, $a \in \Sigma$

By induction hypothesis

$$\hat{\delta}_D(\{q_0\}, x) = \hat{\delta}_N(q_0, x) = \{p_1, p_2 \dots p_k\} \text{ (say)}$$

Then, $$\hat{\delta}_N(q_0, w) = \hat{\delta}_N(q_0, xa) = \bigcup_{i=1}^{N} \delta_N(p_i, a) \qquad \text{—— (1)}$$

Also, $$\hat{\delta}_D(\{q_0\}, w) = \hat{\delta}_D(\{q_0\}, xa)$$

$$= \delta_D(\hat{\delta}_D(\{q_0\}, x), a)$$

(by defition of $\hat{\delta}_D$)

$$= \delta_D(\{p_1, p_2 \dots p_k\}, a)$$

(by induction hypothesis)

$$= \bigcup_{i=1}^{k} \delta_N(p_i, a)$$ (by subset construction)

—— (2)

(1) and (2) establishes that our claim is true for w where $|w| = n+1$ whenever it is true for x with $|x| = n$

It also holds for $|w| = 0$

Hence by induction, the claim follows.

Theorem:

A language L is accepted by DFA iff L is accepted by some NFA.

Proof:

L accepted by NFA

$\Rightarrow$ L is accepted by a DFA using subset construction

(by using previous theorem)

L accepted by a DFA $D = (Q, \Sigma, \delta_D, q_0, F)$

can be interpreted as an NFA

$N = (Q, \Sigma, \delta_N, q_0, F)$

where δ_N is defined by

$\delta_N(q, a) = \{ \{p\} \text{ if } \delta_D(q, a) = \phi \}$

This NFA accepts L.

Note:

NFA $\xrightarrow[\text{construction}]{\text{subset}}$ DFA

n states — 2^n states

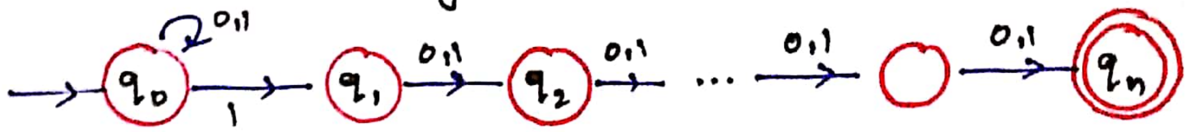
throw away states that are not reachable from the initial state

$\approx n$ states.

Example:

$L(N) = \{ w \mid w \in \{0,1\}^*$ with n-th symbol from the end is $(0+1)^* 1 (0+1)^{n-1} \}$

The NFA realizing $L(N)$ is:



- having $n+1$ states
- equivalent DFA using subset construction has at least 2^n states
- smallest DFA D realizing $L(N)$ cannot have $< 2^n$ states

Otherwise, D can be in state q after reading different sequence of n-bits, say $a_1 a_2 .. a_n$ and $b_1, b_2, ... b_n$. This follows from the pigeon hole principle: " If you have more pegions than pigeon hole and each pegion flies into some pegion hole, then there must be at least one pegion hole that has more than one pegion."

Assumption: # of pegion hole is finite.

- $a_1 a_2 ... a_n \neq b_1 b_2 .. b_n \Rightarrow a_i \neq b_i$ for some i
  let $a_i = 1$, $b_i = 0$

- if $i = 1$ then $q \to$ accepts state as $\overset{1}{a_1} a_2 .. a_n$ is accepted by D

  simultaneously

  absurd! $\to$ rejecting state as $\overset{0}{b_1} b_2 .. b_n$ is rejected by D

if $i > 1$

Scanned by CamScanner

if $i > 1$ then consider state p that D enters after reading $i-1$ 0's

p — accepting state as $a_i \, a_{i+1} \cdots a_n \, 00 \ldots 0$ is accepted by D ($\neq$)

simultaneously

absurd !

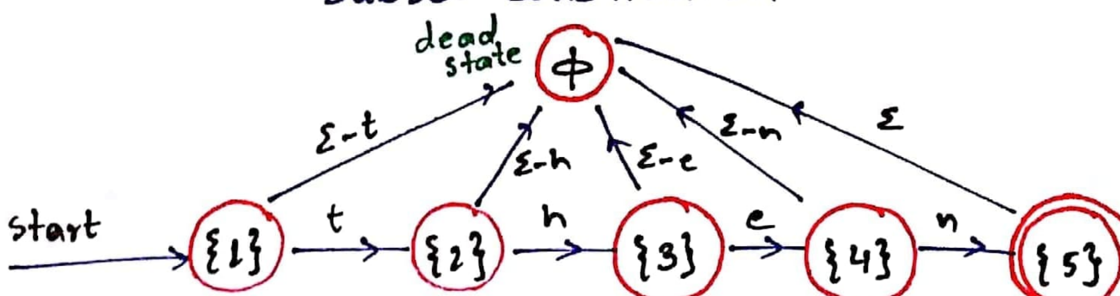
p — rejecting state as $b_i \, b_{i+1} \cdots b_n \, 0 \ldots 0$ is rejected by D

Dead state (DFA)

A non-accepting state that goes to itself on every possible input symbol

Example:

subset construction



Non determinism added to FA

- doesnot expand the class of languages that can accepted by FA
- easier to design than NFA
- can always convert NFA to DFA

  (DFA may have exponentially more states than NFA, fortunately such cases are rare)