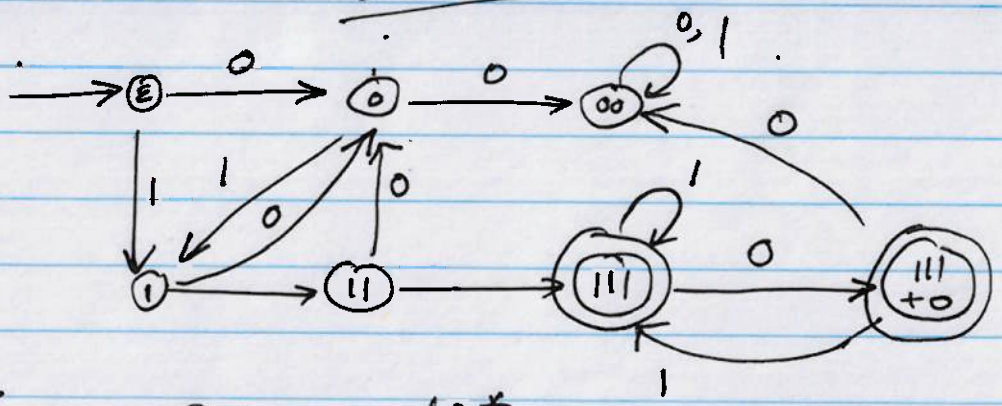
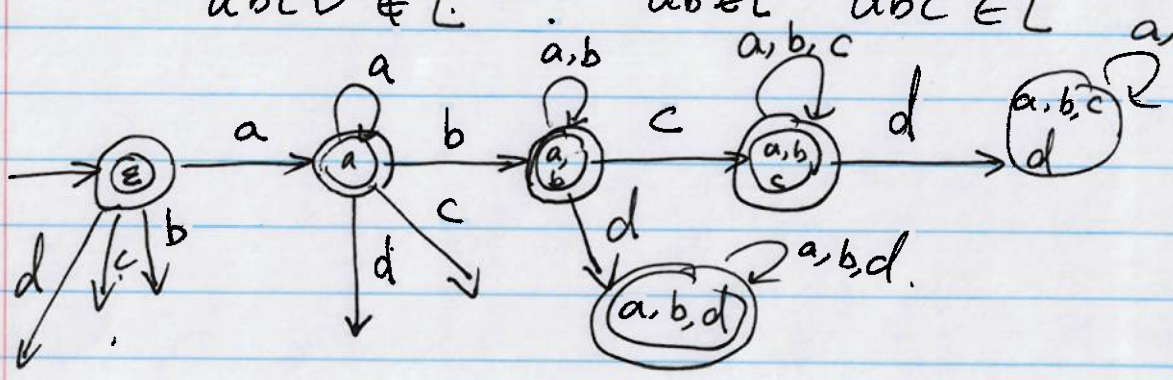
02/03/2025

0 and no 111



$L = \{ w \in \{a, b, c, d\}^* \}$. w doesn't have all alphabet

abcD $\notin L$. ab $\in L$ abc $\in L$



- ε; a, b, c, d (5)
- ab ac, ad bc.. bd. cd (6)
- abd - - - - - (4)
- abcd (1)

DFA: can be complicated.

NFA: 2.4.1. Non-deterministic Finite Automata

1. NFA: DFA $= \{ S, \Sigma, \delta, S_0, S_A \}$.

δ = function.

NFA:

$NFA = \{S, \Sigma, \Delta, S_0, S_A\}$

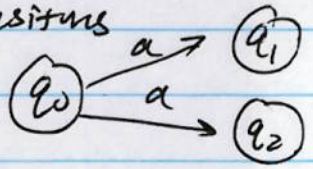
relation.

$\Delta: \quad (S \times (\Sigma \cup \{\varepsilon\})) \times S$

$y = \sqrt{x}$ $R^{\geq 0}$

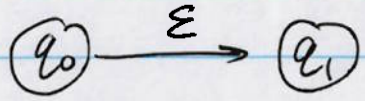
$y = \pm\sqrt{x}$

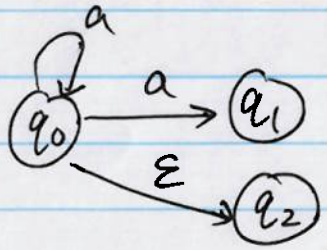
(1). a state can have multiple transitions on the same character.



(2) it can also have no transitions.

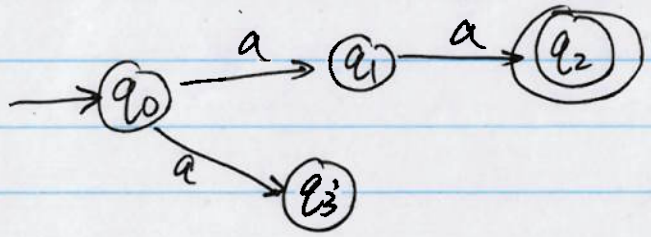
(3). a state can have ε transitions (transition consumes no input)





A string w is accepted by a machine M iff there exists a path from S_0 to an accepting state s.t. the edge labels (on the path) spell w. ignoring ε-s.

M:



aa is accepted

$L(M) := \{$ all strings accepted by $M\}$.

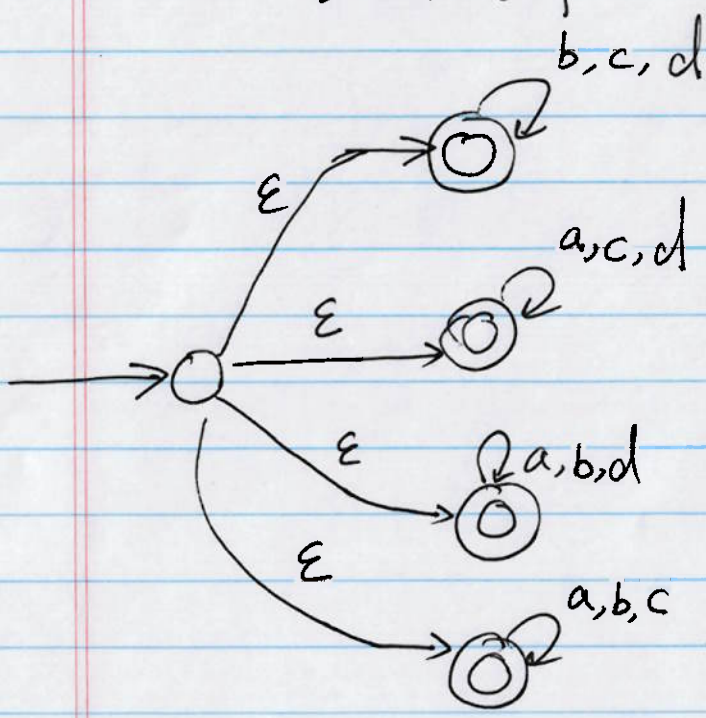
two models for the behavior of NFA.

① . The NFA always guesses the transitions correctly.

② . each time the NFA must make a choice, NFA clone itself for each possible choice.

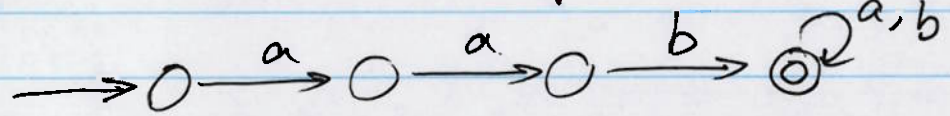
2. example.

ex: $L = \{ w \in \{a, b, c, d\}^* \mid w$ contains at most 3 letters $\}$

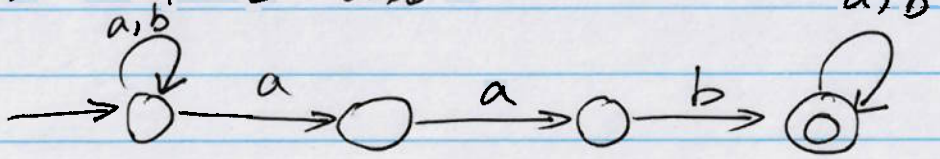


$L = \{ w \in \{a,b\}^* \mid w$ starts with aab $\}$.

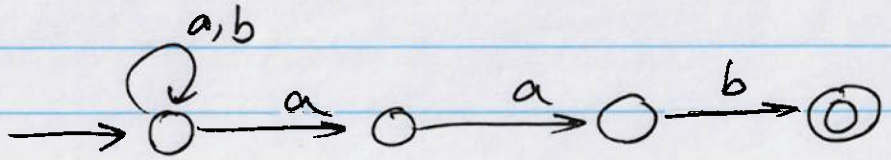
NFA: DFA is a specical case of ~~NDFA~~ NFA.

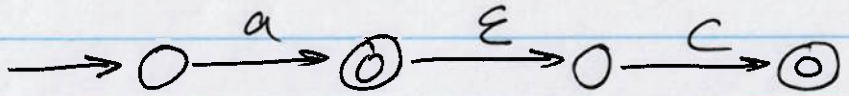


L: w contains aab

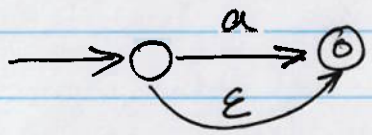


w. ends with aab.



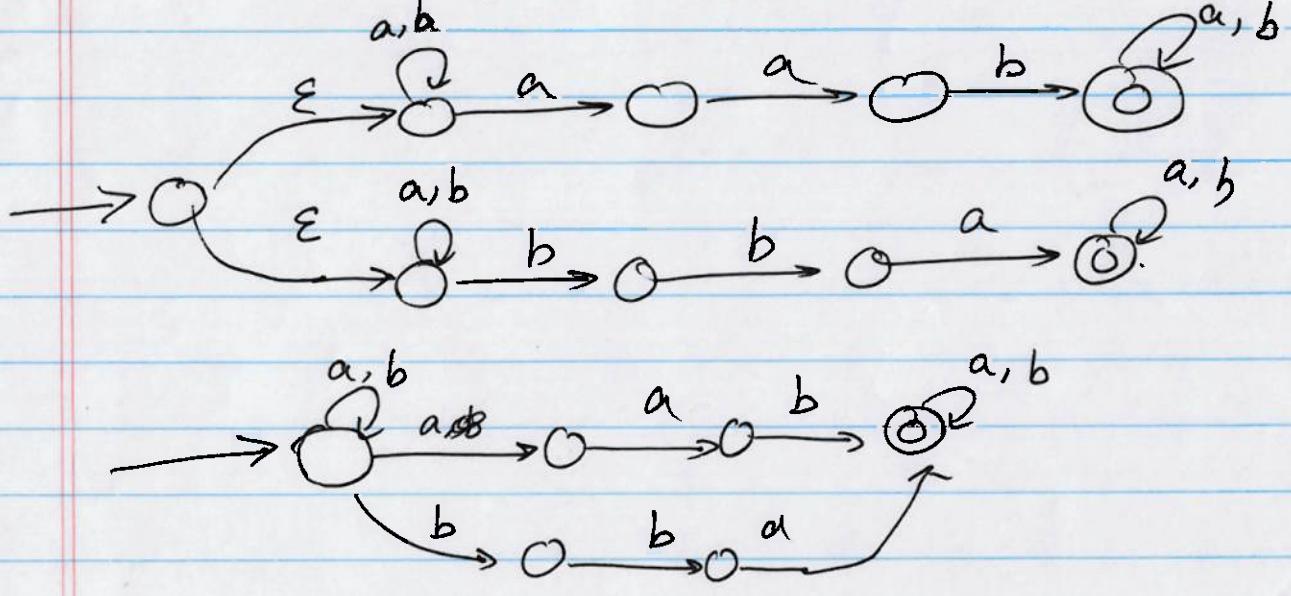


{a, ac}.



{ε, a}.

L: $w \in \{a,b\}^*$. w contains aab or bba



3. NFA vs DFA.

* NFA is more intuitive.

0, 1, 2.... transition.

ε transition.

* NFA ≡ DFA: a language is accepted by recognized.

a DFA iff it is accepted by an NFA

* Why study NFA:

key to automate the RE → DFA. scanner

construction.

4. Automating scanner construction. 2.4.2.

automating

① write down RE for the lexical structure of a language

② Build a big simple NFA.

③ systematically construct a minal DFA.

④ turn it into code.

RE → NFA $((a|b)c)^*$

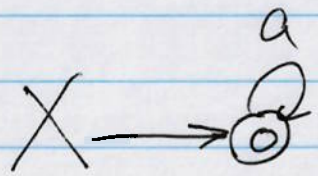
Thompson construction:

bottom up: * NFA for each symbol.

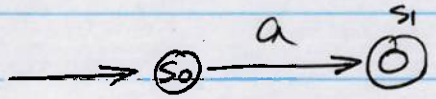
* join NFAs with ε transition.

$\Sigma = \{a, b\}$.

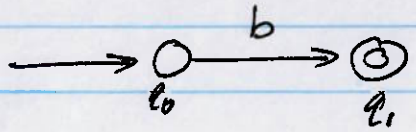
$\{a\}$



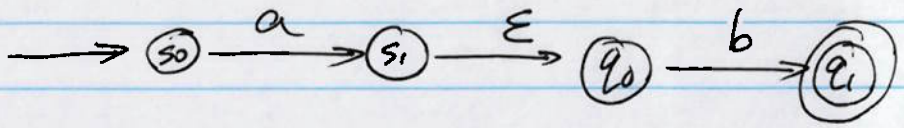
ε a aa



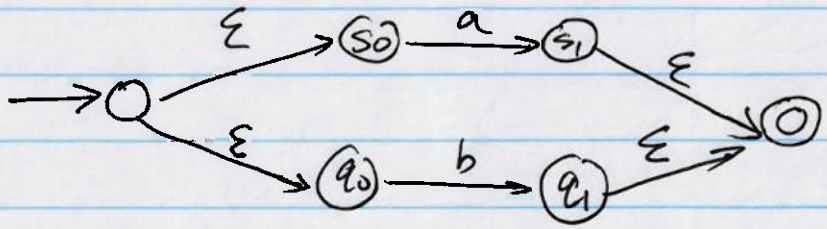
$\{b\}$



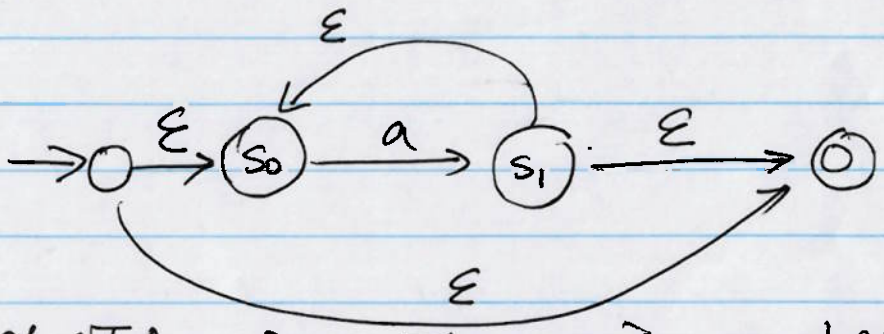
ab



a | b



a^*



properties of NFAs from Thompson's construction.

* . each NFA has one start state.

has <u>only</u> one accepting state

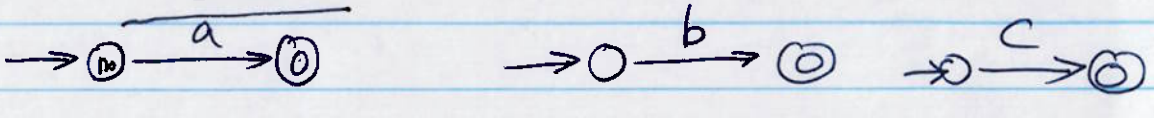
* . start state only has outgoing transitions.

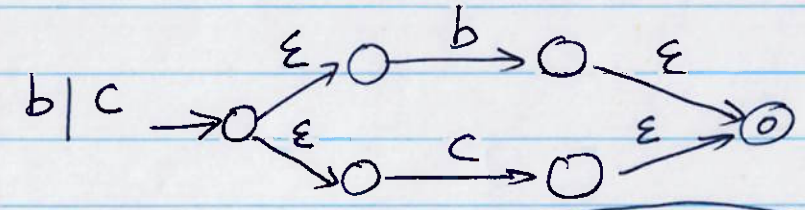
accepting state only has incoming transitions.

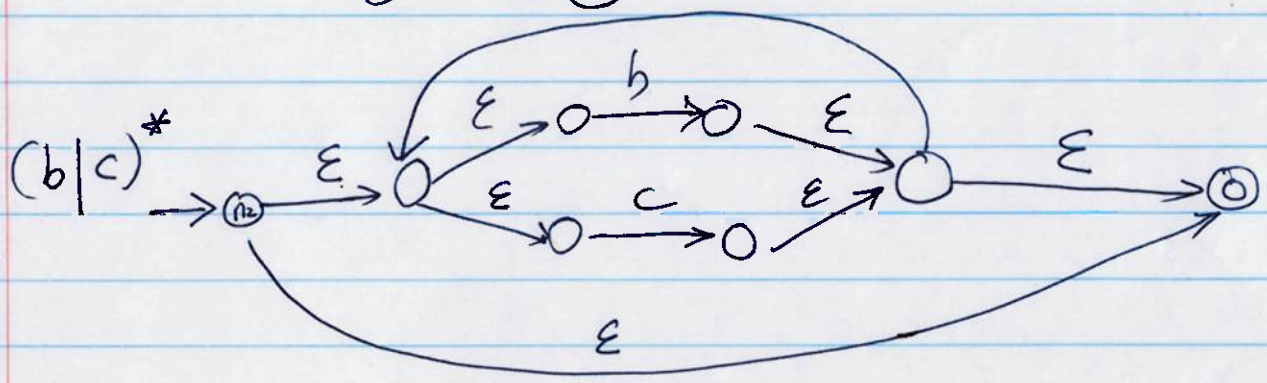
* each state has ① at most 2 incoming & transition.
② and at most 2 outgoing ε transitions.
③ at most 1 incoming/entering transition/move on a symbol in the alphabet
④ at most 1 outgoing/exit transition/move on a symbol in the alphabet.

02/05/2025

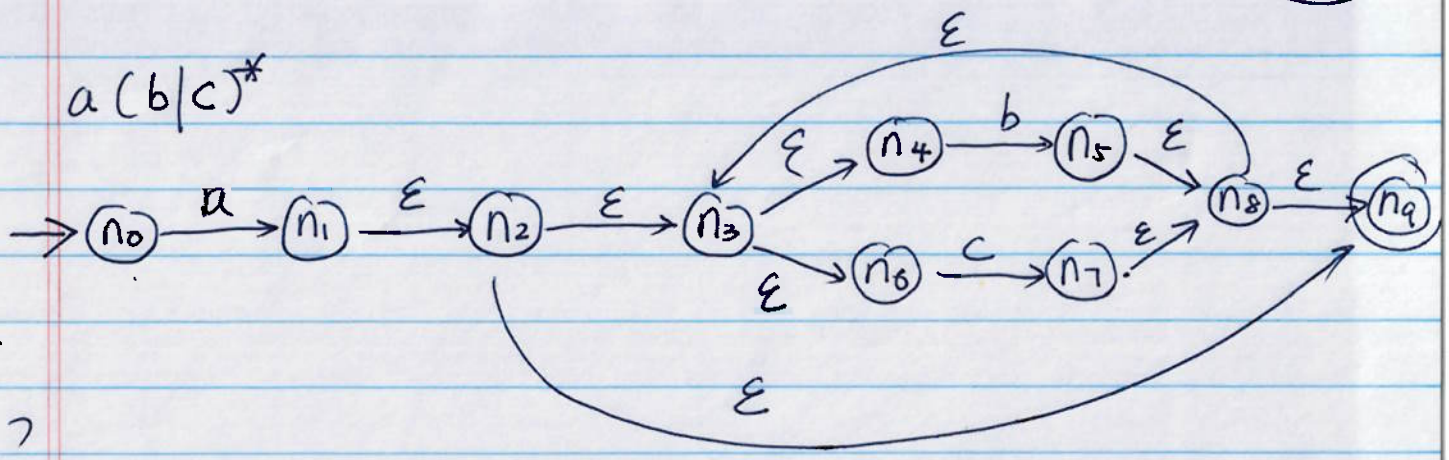
example. $a\,(b|c)^*$ Thompson construction → NFA





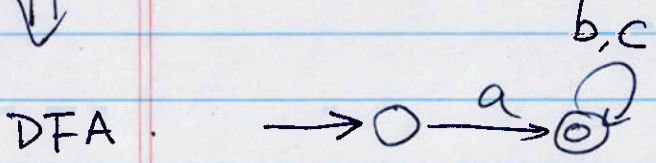


$a(b|c)^*$



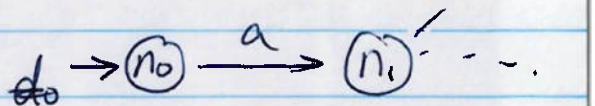
NFA.

?

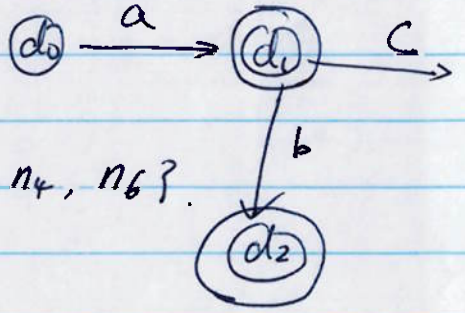


2.4.3 simulating an NFA with DFA : subset construction

NFA. initial state n_0.



DFA: $d_0 = \{n_0\}$.



$d_1 = \{n_1, n_2, n_3, n_9, n_4, n_6\}$.

Core state

$d_2 = \{n_5, n_8, n_3, n_4, n_6, n_9\}$.

each ∄ state in DFA corresponds to a subset of states of NFA.

2 key functions:

Followepsilon (s): s: a set of state in NFA.

the set of states reachable from some state of s by ε-transitions.

Followepsilon ({n_0}) = {n_0}.

Followepsilon ({n_3}) = {n_4, n_6, n_3}.

Followepsilon ({n_3, n_8}) = {n_3, n_8, n_4, n_6, n_9}.

Delta (s, a): a subset of states reachable from some states in s via a transition labeled "a" in NFA.

Delta ({n_2, n_3, n_4} a) = { }. = ϕ

Delta ({n_2, n_3}, b) = ϕ

Delta ({n_2, n_4}, b) = {n_5}.

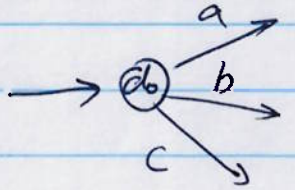
Followepsilon(Delta (s, a))

subset construction.

input: NFA $(N, \Sigma, \delta, n_0, N_A)$

output DFA: (D, Σ, T, d_0, D_A)

steps: $d_0 = \text{Follow}_{\text{epsilon}}(\{n_0\})$



$D = \{d_0\}$

$W = \{d_0\}$. list of states need to compute transition

if $d_0 \cap N_A \neq \phi$, add d_0 to D_A.

while $(W \neq \phi)$

remove a state s from W (5)

for each $\alpha \in \Sigma$

$t \Leftarrow \text{Follow}_{\text{epsilon}}(\text{Delta}(s, a))$

$T[s, \alpha] = t$

if $t \notin D$

add t to D // $D = D \cup \{t\}$

add t to W // $W = W \cup \{t\}$.

if $t \cap N_A \neq \phi$

add t to D_A // $D_A = D_A \cup \{t\}$.

followepsion (Delta(S,α)).

DFA state.	NFA states.	a	b	c
d_0	$\{n_0\}$	$\{n_1, n_2, n_3, n_4, n_6, n_9\}$	ϕ	ϕ
d_1	$\{n_1, n_2, n_3, n_4, n_6, n_9\}$.	ϕ	$\{n_5, n_8, n_3, n_9, n_4, n_6\}$	$\{n_7, n_8, n_9, n_3, n_4, n_6\}$
d_2.	$\{n_5, n_8, n_3, n_9, n_4, n_6\}$	ϕ	$\{n_5, n_8, n_3, n_9, n_4, n_6\}$.	$\{n_7, n_8, n_9, n_3, n_4, n_6\}$.
d_3	$\{n_7, n_8, n_9, n_3, n_4, n_6\}$	ϕ	d_2	d_3

d_0 —a→ d_1

$w = \{d_1\}$.

$D_A = \{d_1\}$

d_1 —b→ d_2

$w = \{d_2\}$

$D_A = \{d_1\}$

d_1 —b→ d_2

$w = \{d_2\}$.

$D_A = \{d_1, d_2\}$

d_1 —c→ d_3

$w = \{d_2, d_3\}$

$D_A = \{d_1, d_2, d_3\}$.

$w = \{d_3\}$.

$D_A = \{d_1, d_2, d_3\}$

d_2 —b→ d_2

$w = \{d_3\}$

$D_A = \cdots$

d_1 —c→ d_3

$w =$

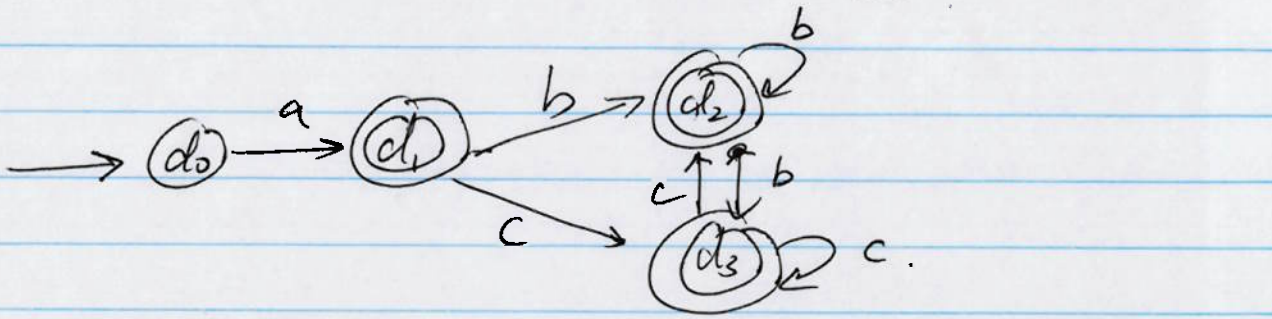
D_A

$w = \phi$.

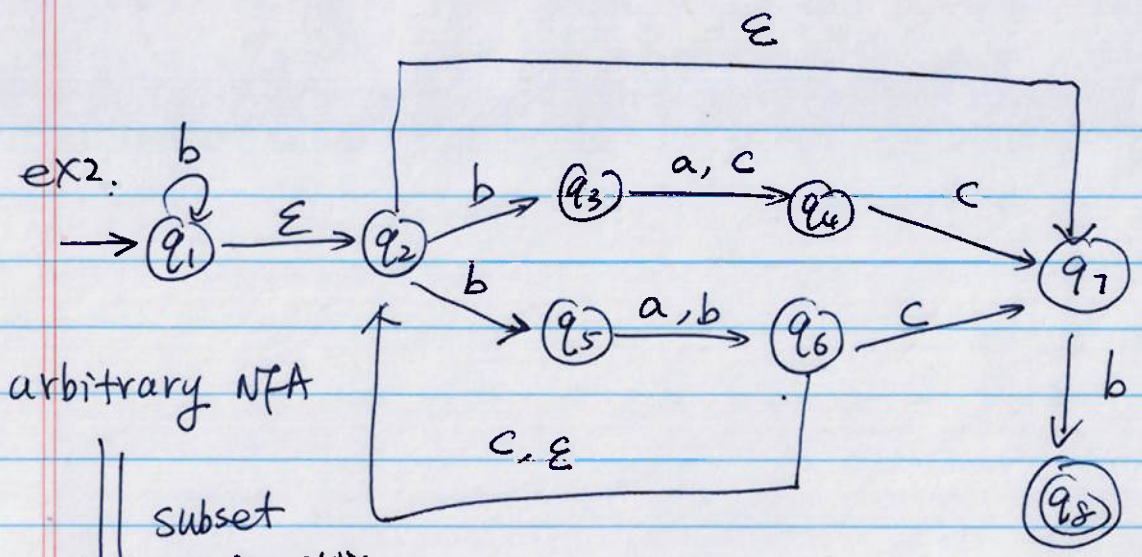
$D_A = \cdots$

d_3 —b→ d_2

d_3 —c→ d_3



ex2.



arbitrary NFA

subset construction.

DFA: