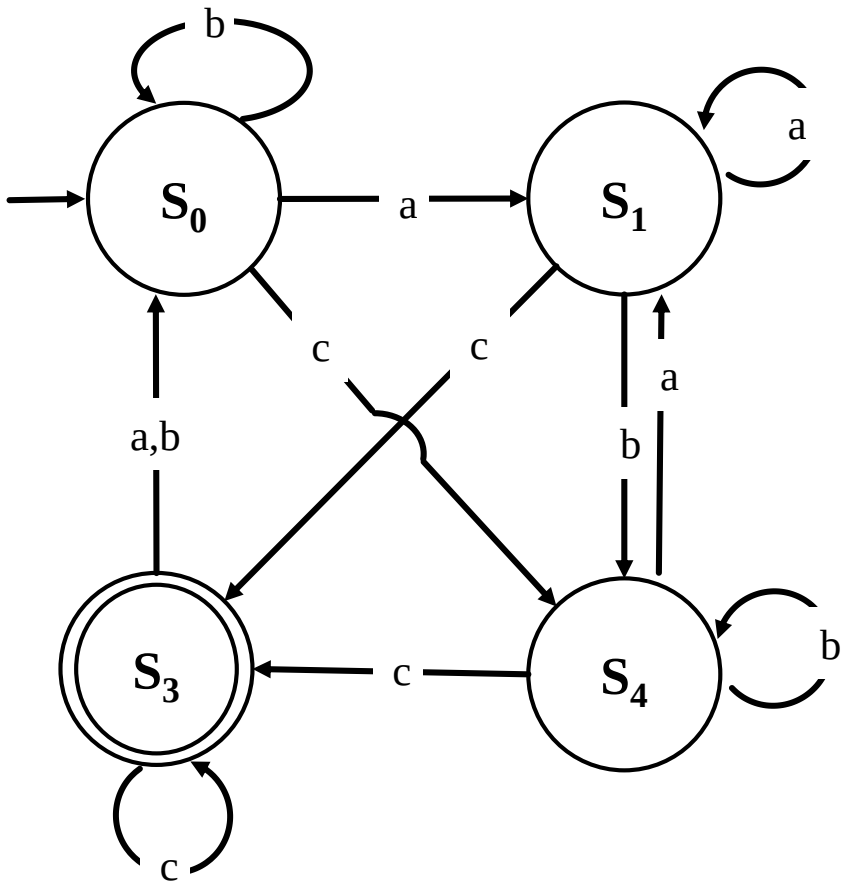
Finite State Automata (2)

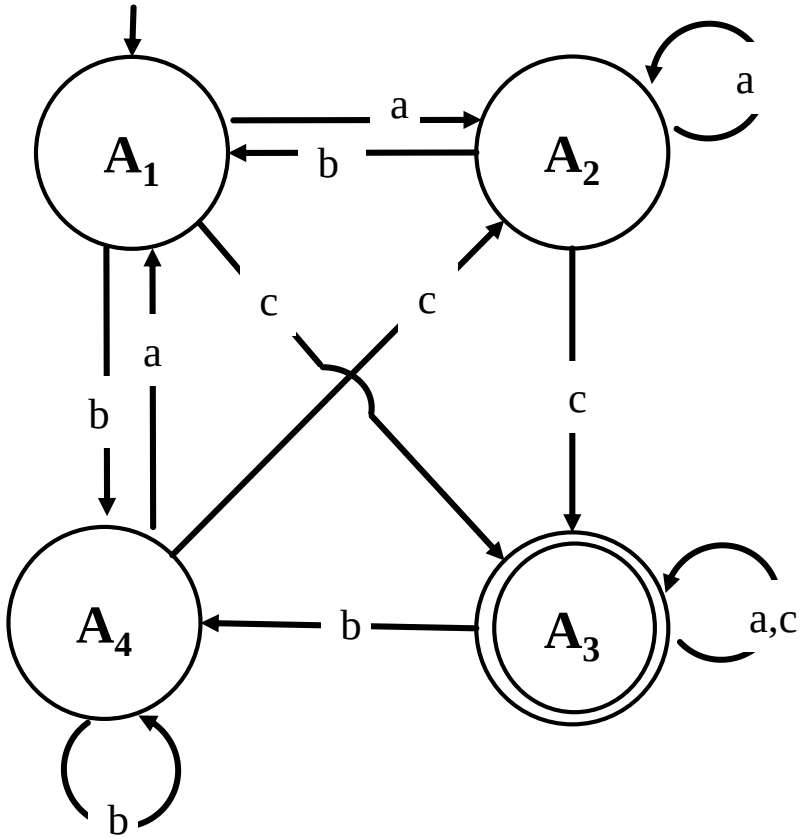
Buat DFA yang dapat menerima bahasa berikut dengan 4 buah state :

a) $L(A) = \{x \mid x = a^n b^n c^n, n \geq 1, x \in \{a,b,c\}\}$
b) $L(A) = \{x \mid x = (ab)^m C^n, m,n\ n \geq 1,\ x \in \{a,b,c\}\}$
c) $L(A) = \{x \mid x = (ab)^* c, x \in \{a,b,c\}\}$

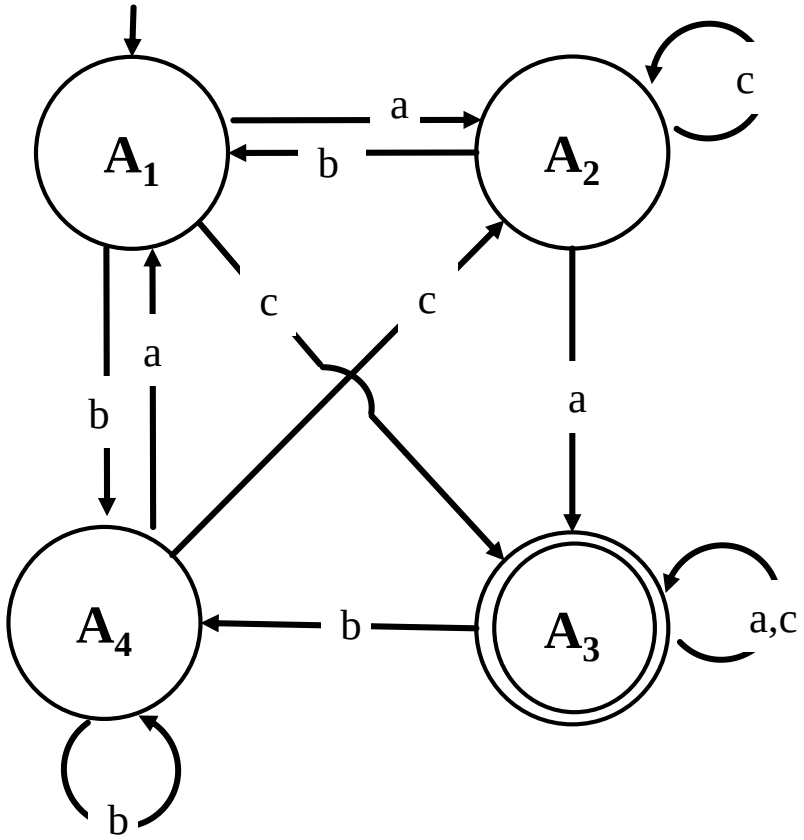
$L(A) = \{x \mid x = a^n b^n c^n, n \geq 1, x \in \{a,b,c\}\}$



$L(A) = \{x \mid x = (ab)^m C^n,\ m,n\ n \geq 1,\ x \in \{a,b,c\}\}$



$L(A) = \{x \mid x = (ab)^*c,\ x \in \{a,b,c\}\}$



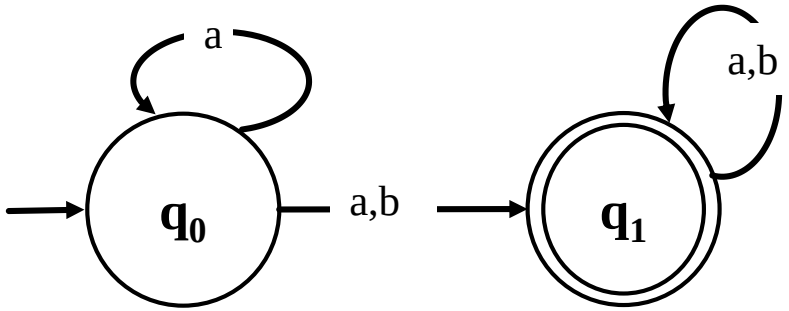
Pada intinya, paling mendasar perbedaan DFA dan NDFA, jika DFA apabila state diberi input, maka state akan selalu tepat menuju 1 state. Berbeda dengan NDFA, jika state diberi input, mungkin saja bisa menuju ke beberapa state selanjutnya.

DFA	NDFA
DFA tidak dapat menggunakan transisi *string* kosong (*empty string*)	NDFA dapat menggunakan transisi *string* kosong (*empty string*)
DFA dipahami sebagai sebuah mesin	NDFA dipahami sebagai beberapa mesin kecil yang melakukan komputasi di waktu bersamaan
DFA untuk *state* selanjutnya bisa ditetapkan dengan jelas	NDFA untuk *state* selanjutnya mempunyai banyak kemungkinan
DFA lebih sulit dibuat	NDFA lebih mudah dibuat
Waktu yang dibutuhkan untuk mengeksekusi string input lebih sedikit	Waktu yang dibutuhkan untuk mengeksekusi string input lebih banyak
Semua DFA merupakan NDFA	Tidak semua NDFA adalah DFA
DFA membutuhkan lebih banyak ruang (*space*)	NDFA membutuhkan lebih sedikit ruang (*space*)

Non Deterministic Finite Automata (NDFA)

- NDFA diperkenalkan pada tahun 1959 oleh **Michael O. Rabin** dan **Dana Scott**.
- Pada NDFA dari suatu state bisa terdapat nol (0), satu (1), atau lebih busur keluar (transisi) berlabel simbol yang sama. Jadi setiap pasangan state-input, kita bisa memiliki 0 atau lebih pilihan untuk state berikutnya

Contoh 1 :



Pada NDFA diatas terdapat dua busur keluar berlabel input ‘a’.

Dari state q_0 bila mendapat input ‘a’ bisa berpindah ke state q_0 atau q_1 yang secara formal dinyatakan : $\delta (q_0, a) = \{q_0, q_1\}$

Konfigurasi NDFA pada contoh diatas secara formal adalah sebagai berikut :

- $Q = \{q_0, q_1\}$
- $\Sigma = \{a, b\}$
- $S = \{q_0\}$
- $F = \{q_1\}$

Fungsi-fungsi transisinya :

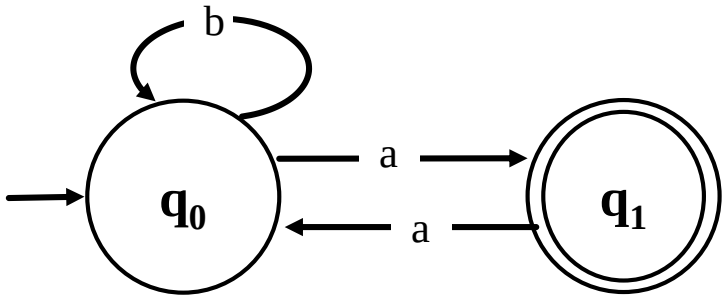
$\delta (q_0, a) = \{q_0,q_1\}$, $\delta (q_0, b) = \{q_1\}$,

$\delta (q_1, a) = \{q_1\}$ $\delta (q_1, b) = \{q_1\}$,

δ	a	b
q_0	$\{q_0,q_1\}$,	$\{q_1\}$
q_1	$\{q_1\}$	$\{q_1\}$

Perhatikan :
Dalam cara penulisan state hasil transisi pada tabel transisi untuk NDFA, digunakan kurung kurawal ‘{‘ dan ‘}’ karena hasil transisisnya merupakan suatu himpunan state.

Contoh 2 :



Konfigurasi NDFA :

Q = $\{q_0, q_1\}$

Σ = {a, b}

S = q_0

F = $\{q_1\}$

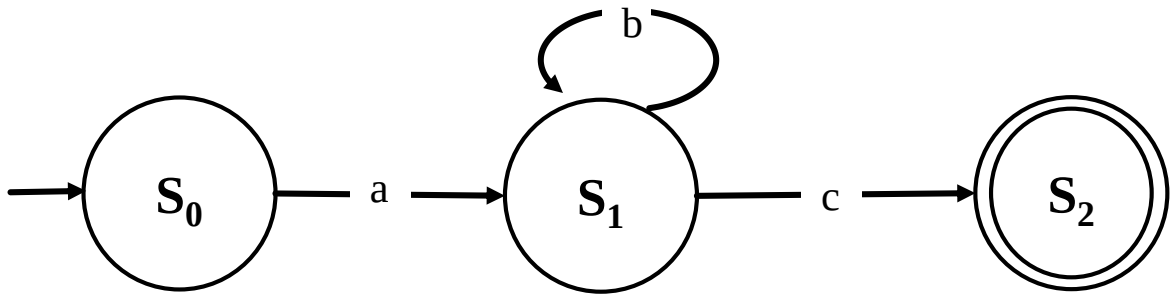
Fungsi-fungsi transisinya :

δ (q_0, a) = q1, δ (q_0, b) = q0,

δ (q_1, a) = q0, δ (q_1, b) = Ø,

δ	a	b
q_0	$\{q_1\}$,	$\{q_0\}$
q_1	$\{q_0\}$	Ø

Contoh 3 :



$Q = \{S_0, S_1, S_2\}$
$\Sigma = \{a, b, c\}$
$S = \{S_0\}$
$F = \{S_2\}$

δ	a	b	c
S_0	$\{S_1\}$	Ø	Ø
S_1	Ø	S_1	S_2
S_2	Ø	Ø	Ø

Fungsi-fungsi transisinya :

$\delta(S_0, a) = \{S_1\}$ $\quad \delta(S_0, b) = Ø$ $\quad \delta(S_0, c) = Ø,$
$\delta(S_1, a) = Ø$ $\quad \delta(S_1, b) = \{S_1\}$ $\quad \delta(S_1, c) = \{S_2\}$
$\delta(S_2, a) = Ø$ $\quad \delta(S_2, b) = Ø$ $\quad \delta(S_2, c) = Ø$