

OTOMATA 04

NON DETERMINISTIC FINITE AUTOMATA

MATERI PERTEMUAN

- Pengertian Non-Deterministik Finite Automata (NFA)
- Komponen NFA
- Konversi NDFA ke DFA
- Konversi RE ke NFA

Pengertian NFA



Pada saat mempelajari DFA, kita telah berkenalan dengan sebuah mesin yang dapat memiliki label edge berupa string. Dapat memiliki lebih dari satu start maupun final state. Dan (yg paling ekstrem):

- dapat memiliki lebih dari satu outgoing edge dengan label sama;
- mampu menerima ϵ -transition (transisi λ)

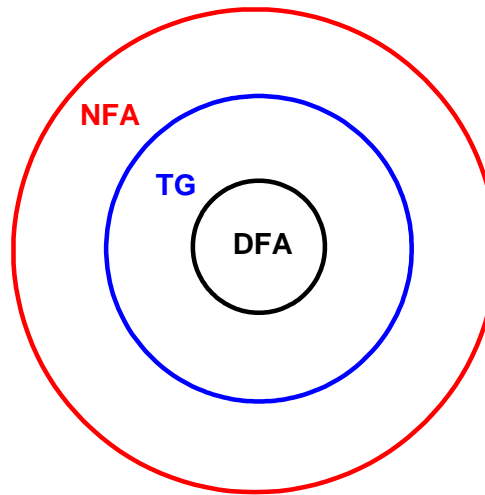
Kedua sifat terakhir ini yang menyebabkan TG tidak 'layak' lagi disebut sebagai varian dari NFA!

Kenapa?...

Komponen NFA (1)

Secara definitif, NFA memiliki komponen-komponen :

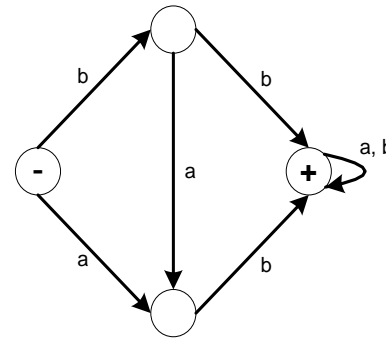
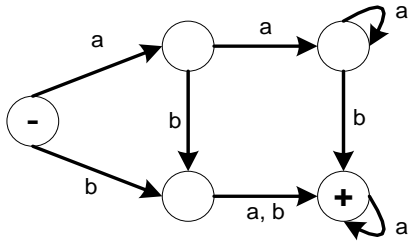
1. S sebagai himpunan berhingga **state** untuk media perpindahan kendali mesin
2. Σ sebagai himpunan berhingga alphabet untuk **input karakter**
3. s_0 adalah salah satu state dari himpunan S yang diperlakukan sebagai **start state**
4. s_n adalah salah satu state dari himpunan S yang diperlakukan sebagai **final state**
(DFA dapat memiliki lebih dari satu final state)
5. δ sebagai himpunan berhingga **fungsi transisi** untuk memindahkan kendali mesin.
Dimungkinkan adanya lebih dari satu outgoing edge dengan label sama yang keluar dari sebuah state.



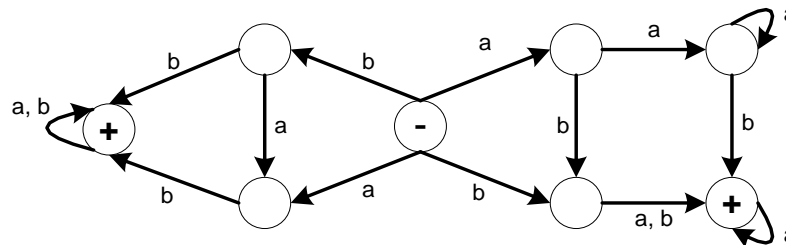
Komponen NFA (2)

Lebih mudah mendefinisikan bahasa dengan NFA ???

Misalkan terdapat 2 bahasa r_1 dan r_2 yang masing-masing berkorespondensi dengan DFA_1 dan DFA_2 :



Jika dilakukan operasi **UNION** terhadap kedua DFA di atas, dan hasilnya boleh berupa NFA, maka operasi dapat diimplementasi dengan mudah :

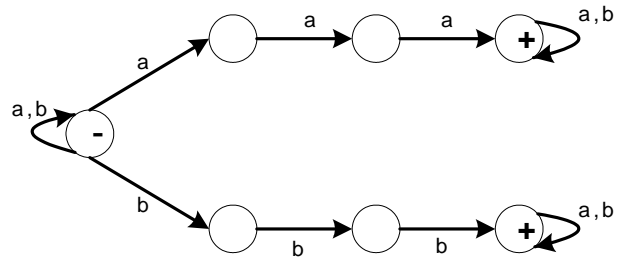


Komponen NFA (3)



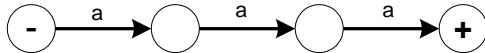
Contoh :

Sebuah bahasa yang menerima semua string yang mengandung substring aaa atau bbb. Melalui NFA, bahasa tersebut dapat didefinisikan sebagai berikut :



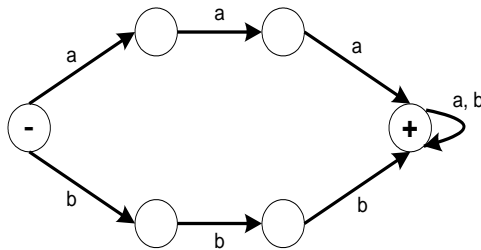
Bagaimana jika didefinisikan melalui DFA ?

Untuk RE: aaa



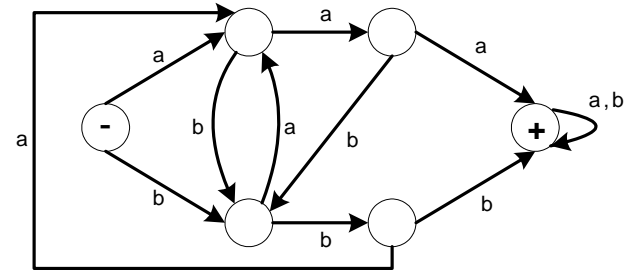
Untuk RE:

aaa(a + b)* atau bbb (a + b)*



Untuk bahasa :

$[(a + b)^* aaa (a + b)^* + (a + b)^* bbb (a + b)^*]$



Konversi NFA ke DFA (1)

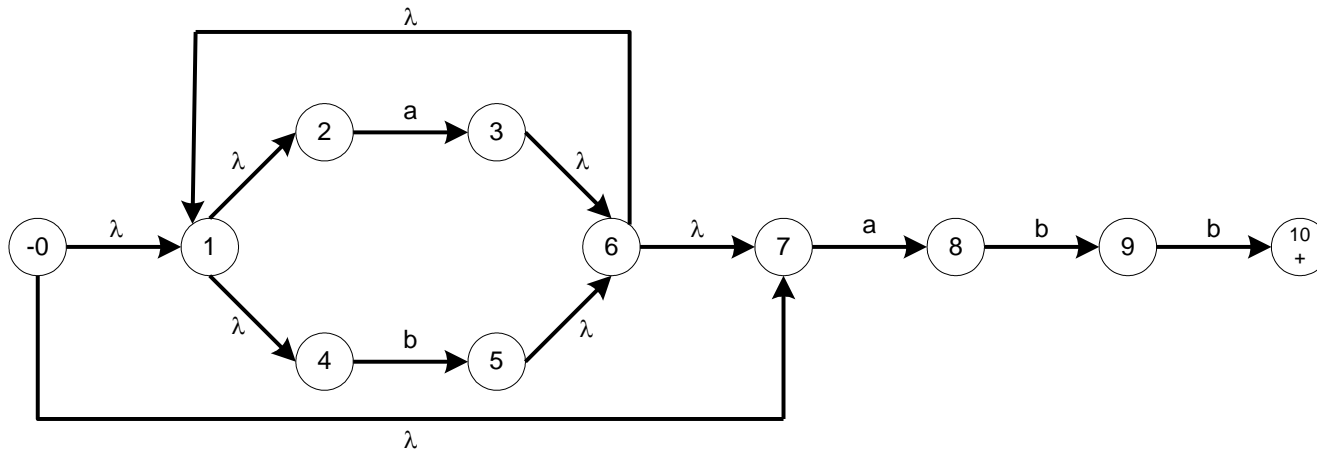
Beberapa operasi yang digunakan dalam proses konversi antara lain :

Jenis Operasi	Keterangan
λ -closure(s)	Himpunan state pada NFA yang dapat dikunjungi dari state S melalui input karakter λ
λ -closure(T)	Himpunan state pada NFA yang dapat dikunjungi dari satu/lebih state pada sub-himpunan T melalui karakter λ
Move(T, a)	Himpunan state pada NFA yang dapat dikunjungi melalui input karakter a dari satu/lebih state pada sub-himpunan T

Konversi NFA ke DFA (2)

Contoh :

Misalkan terdapat sebuah NFA yang menerima bahasa $(a + b)^*abb$



Diawali dari start state :

$$\lambda\text{-closure}(0) = \{0, 1, 2, 4, 7\} = \mathbf{A}$$

Jika A diberi input a, maka

$$\text{Move}(A, a) = \{3, 8\}$$

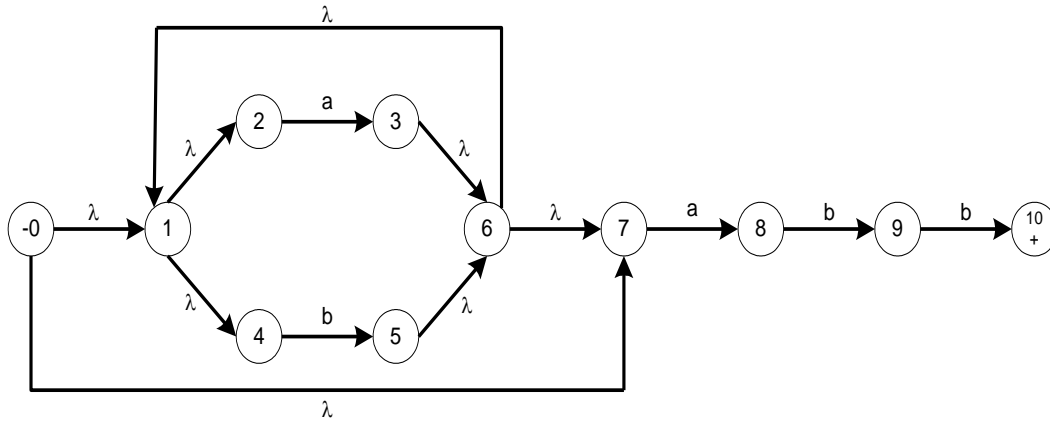
$$\lambda\text{-closure}(\{3, 8\}) = \{1, 2, 3, 4, 6, 7, 8\} = \mathbf{B}$$

Jika A diberi input b, maka

$$\text{Move}(A, b) = \{5\}$$

$$\lambda\text{-closure}(\{5\}) = \{1, 2, 4, 5, 6, 7\} = \mathbf{C}$$

Konversi NFA ke DFA (3)



Jika B diberi input a, maka

$\text{Move}(B, a) = \{3, 8\}$

$\lambda\text{-closure}(\{3, 8\}) = \mathbf{B}$

Jika B diberi input b, maka

$\text{Move}(B, b) = \{5, 9\}$

$\lambda\text{-closure}(\{5, 9\}) = \{1, 2, 4, 5, 6, 7, 9\}$
 $= \mathbf{D}$

Jika C diberi input a, maka

$\text{Move}(C, a) = \{3, 8\}$

$\lambda\text{-closure}(\{3, 8\}) = \mathbf{B}$

Jika C diberi input b, maka

$\text{Move}(C, b) = \{5\}$

$\lambda\text{-closure}(\{5\}) = \mathbf{C}$

Jika D diberi input a, maka

$\text{Move}(D, a) = \{3, 8\}$

$\lambda\text{-closure}(\{3, 8\}) = \mathbf{B}$

Jika D diberi input b, maka

$\text{Move}(D, b) = \{5, 10\}$

$\lambda\text{-closure}(\{5, 10\}) = \{1, 2, 4, 5, 6, 7, 10\}$
 $= \mathbf{E}$

Jika E diberi input a, maka

$\text{Move}(E, a) = \{3, 8\}$

$\lambda\text{-closure}(\{3, 8\}) = \mathbf{B}$

Jika E diberi input b, maka

$\text{Move}(E, b) = \{5\}$

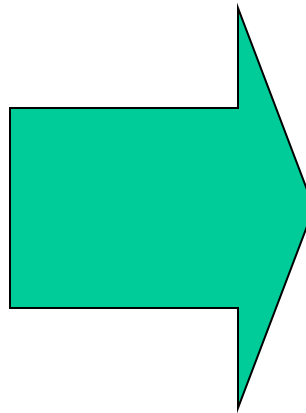
$\lambda\text{-closure}(\{5\}) = \mathbf{C}$

Konversi NFA ke DFA (4)

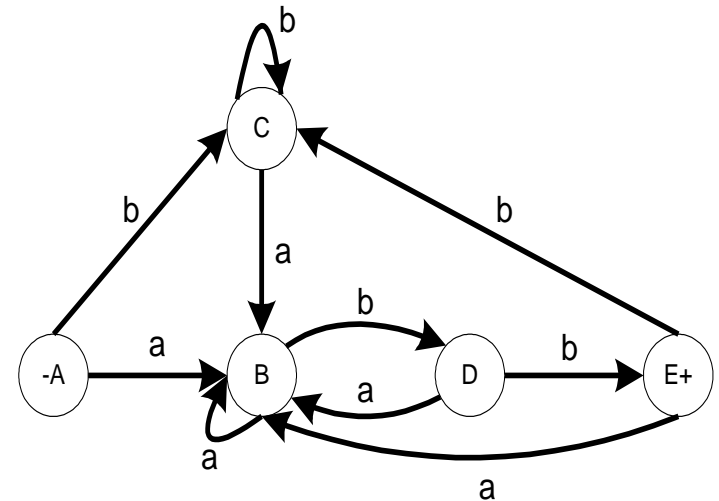


Tabel Transisi

	a	b
- A	B	C
B	B	D
C	B	C
D	B	E
+ E	B	C

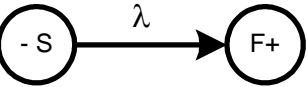
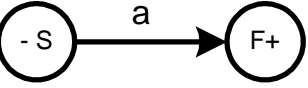


Associated DFA

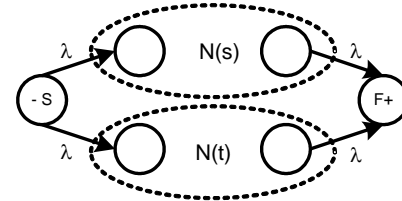


RE: $(a + b)^*abb$

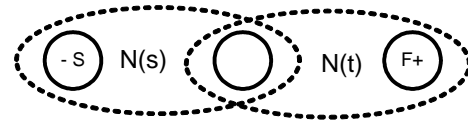
Konversi RE ke NFA (1)

1. Untuk ekspresi λ : 
2. Untuk karakter a : 
3. Misal $N(s)$ dan $N(t)$ yang masing-masing adalah NFA untuk RE s dan t

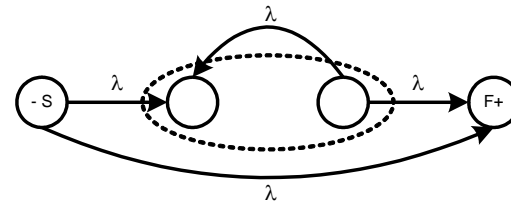
a. Operasi **UNION** $s|t$ akan menghasilkan NFA yang bersesuaian $N(s|t)$:



b. Operasi **CONCATENATE** st menghasilkan NFA yang bersesuaian $N(st)$:



c. Operasi **CLOSURE** s^* menghasilkan NFA yang bersesuaian $N(s^*)$:



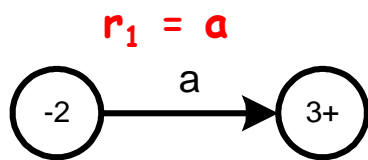
d. Untuk ekspresi (s) tidak menghasilkan perubahan apapun pada NFA asal.

Konversi RE ke NFA (2)

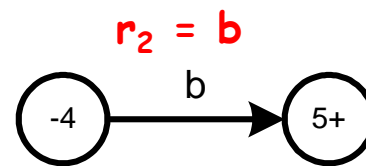
Contoh :

NFA yang bersesuaian untuk bahasa $r = (a + b)^* abb$

Untuk

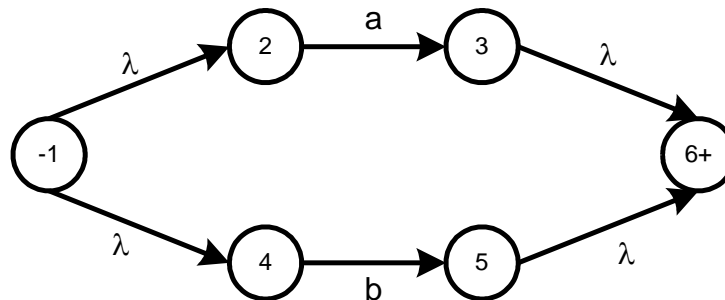


dan



Untuk

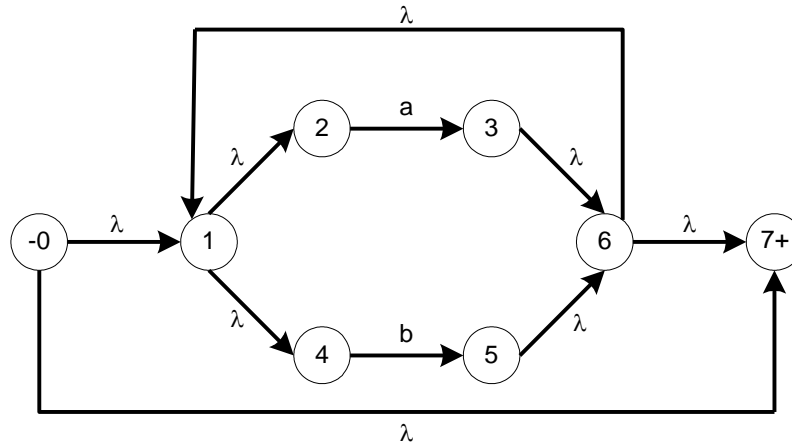
$r_3 = r_1 + r_2$



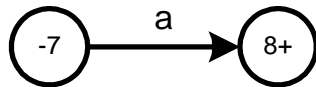
Untuk (r_3) , NFA sama dengan yang di atas

Konversi RE ke NFA (3)

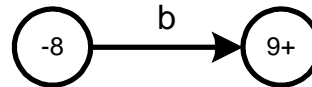
Untuk $(r_3)^*$:



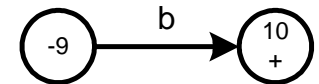
Untuk $r_4 = a$



$r_5 = b$



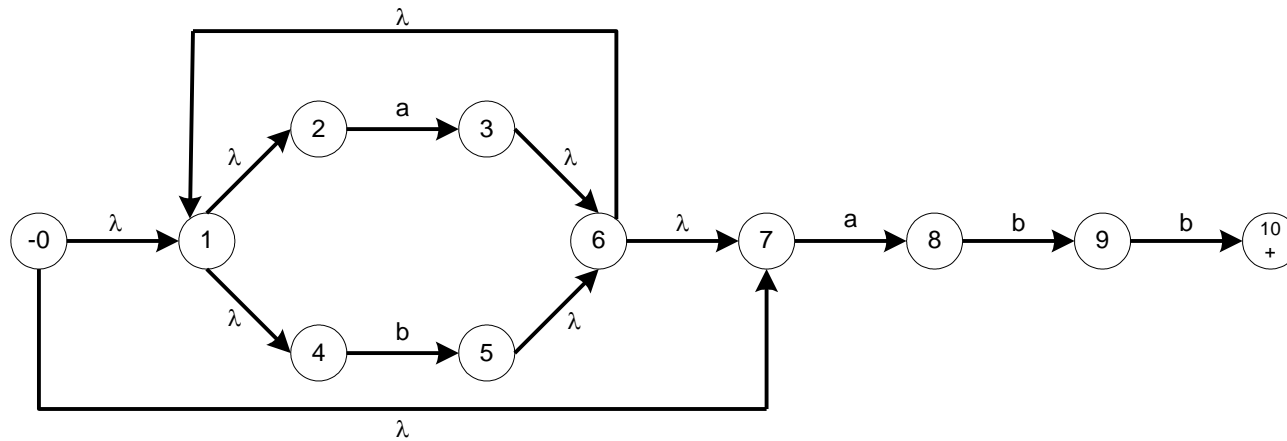
$r_6 = b$



Konversi RE ke NFA (4)

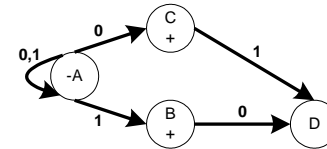
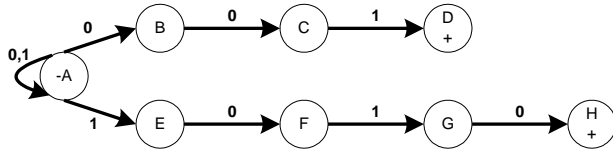


Sehingga jika $(r_3)^* r_4 r_5 r_6$ akan menghasilkan NFA seperti berikut :

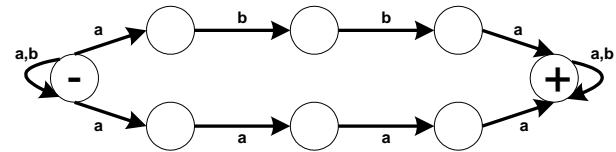
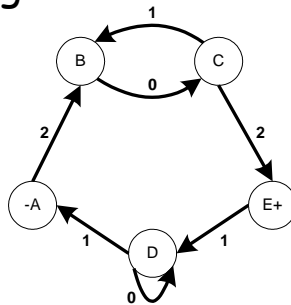
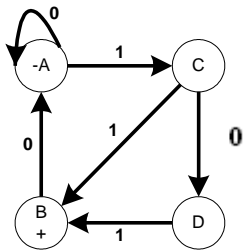


Tugas

1. Konversikan NFA berikut menjadi DFA :



2. Deskripsikan bahasa yang diterima oleh NFA berikut :



3. Carilah NFA yang ekivalen (dengan state lebih sedikit) dengan DFA berikut :

