

1. Menurut saya perbedaan Ekuivalensi NFA dengan e-move ke NFA tanpa e-move dengan Ekuivalensi NFA ke DFA adalah bahwa NFA tanpa e-move adalah sebuah NFA yang tidak memiliki transisi kosong, sementara Ekuivalensi NFA ke DFA adalah sebuah proses yang digunakan untuk mengubah sebuah NFA menjadi sebuah DFA yang ekuivalen. Sehingga, meskipun NFA tanpa e-move dan DFA yang dihasilkan melalui Ekuivalensi NFA ke DFA mungkin memiliki beberapa karakteristik yang sama, proses yang digunakan untuk menghasilkan kedua jenis automata tersebut berbeda.

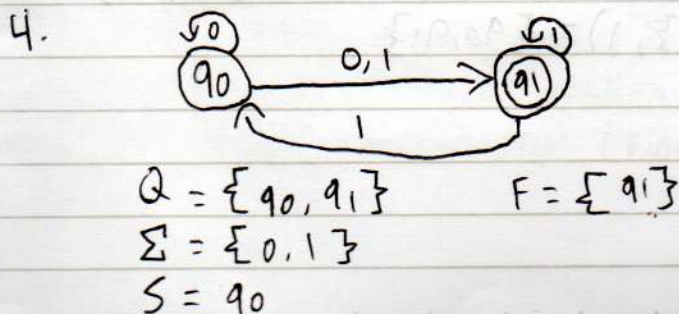
2. Menurut pendapat saya, fungsi dari ekspresi reguler dalam proses pembuatan diagram transisi FSA adalah untuk membantu dalam menentukan set karakter yang diterima oleh FSA, sehingga membantu dalam pembuatan diagram transisi yang tepat dan memudahkan pembuatan kode program untuk menerima input sesuai dengan kebutuhan. Ekspresi reguler dapat diinterpretasikan sebagai aturan yg menentukan apa saja yang dapat diterima oleh FSA sebagai input.

3. Menurut saya fungsi dari ekspresi reguler terhadap diagram transisi FSA adalah sebagai berikut:

1). Menentukan pola yang dapat diterima oleh FSA: Ekspresi reguler menentukan apa saja yang dapat diterima oleh FSA sebagai input, sehingga FSA dapat digunakan untuk memvalidasi string atau kalimat sesuai dengan pola yang telah ditentukan.

2). Mempermudah proses pembuatan diagram transisi FSA: Dengan menggunakan ekspresi reguler sebagai dasar, kita dapat dengan mudah menentukan transisi yang akan dibuat pada FSA, sehingga proses pembuatan diagram transisi FSA menjadi lebih mudah dan efisien.

3). Mempermudah pemahaman FSA: Ekspresi reguler memberikan penjelasan yang jelas dan sistematis tentang pola yang dapat diterima oleh FSA, sehingga mempermudah pemahaman tentang FSA bagi para pemakai atau pengembang.

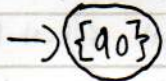


Dengan tabel transisi

δ	0	1
q_0	(q_0, q_1)	(q_1)
q_1	\emptyset	(q_0, q_1)

Selanjutnya dari tabel transisi tersebut dapat dibuat diagram transisi DFA dengan cara:

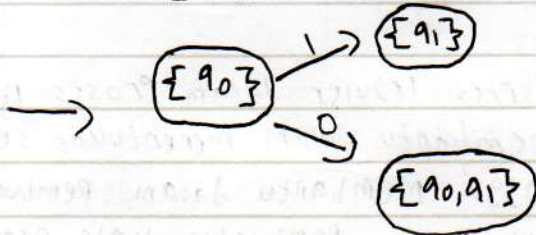
1) Telusuri alur state mulai dari initial state (q_0)



2) Selanjutnya telusuri alur dengan memanfaatkan tabel transisi

→ state $\{q_0\}$ memperoleh inputan 0 menjadi state $\{q_0, q_1\}$

→ state $\{q_0\}$ memperoleh inputan 1 menjadi state $\{q_1\}$

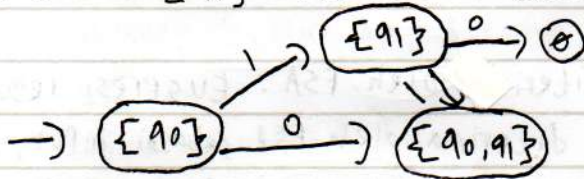


State baru setelah diuji dari tabel transisi

3) Selanjutnya telusuri state yang baru terbentuk

→ state $\{q_1\}$ memperoleh inputan 0 menjadi state \emptyset

→ state $\{q_1\}$ memperoleh inputan 1 menjadi state $\{q_0, q_1\}$

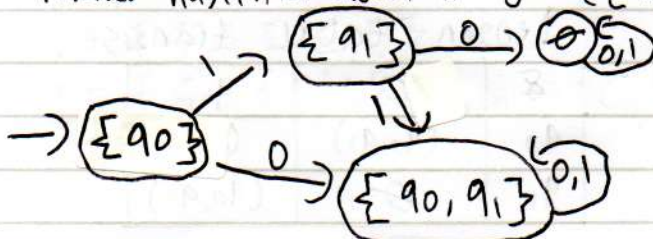


4) Menguji state $\{q_0, q_1\}$ dengan menelusuri state yang baru terbentuk di mana:

→ jika state $\{q_0, q_1\}$ memperoleh input 0 menjadi state $\{q_0, q_1\}$ yg diperoleh dari $\delta(q_0, 0) = \{q_0, q_1\}$ digabung dengan $\delta(q_1, 0) = \{\emptyset\}$ sehingga hasilnya $\delta(\{q_0, q_1\}, 0) = \{q_0, q_1\}$

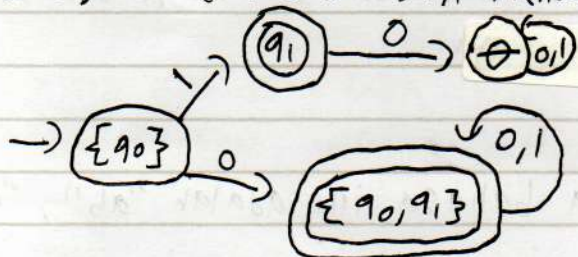


→ state $\{q_0, q_1\}$ jika memperoleh input 1 menjadi state $\{q_0, q_1\}$ diperoleh dari $\delta(q_0, 1) = \{q_1\}$ digabung dengan $\delta(q_1, 1) = \{q_0, q_1\}$ maka hasilnya adalah $\delta(\{q_0, q_1\}, 1) = \{q_0, q_1\}$



- 5) Selanjutnya telusuri state yg baru terbentuk (dimana setiap state yg mengandung q_1 maka akan dijadikan final state)
 \rightarrow state \emptyset menerima inputan 0 dan 1 menjadi state \emptyset atau $\delta(\emptyset, 0) = \emptyset$
 dan $\delta(\emptyset, 1) = \emptyset$

Diagram transisi DFA menjadi:

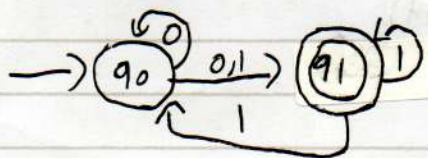


- 6) Uji coba input string dgn diagram transisi NFA dan DFA (yg telah dilakukan proses Ekuivalensi NFA ke DFA)

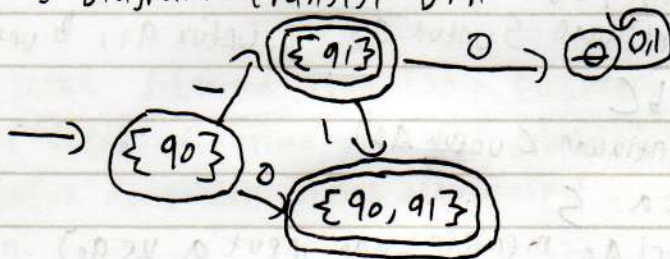
Tabel Transisi

δ	0	1
q_0	$\{q_0, q_1\}$	$\{q_1\}$
q_1	\emptyset	$\{q_0, q_1\}$

\rightarrow Diagram transisi NFA



\rightarrow Diagram transisi DFA



Uji kedua transisi tersebut dengan inputan string

$\rightarrow 0$ $\rightarrow 01110$ $\rightarrow 1100$
 $\rightarrow 0111$ $\rightarrow 00$ $\rightarrow 10$

Buatlah tabel transisi yg baru

δ	0	1
q_0	q_1	q_1
q_1	q_0	q_1

Jadi, untuk setiap simbol Σ yg diberikan, ada 2 kemungkinan transisi yg bisa terjadi, yaitu dari keadaan q_0 ke q_1 atau dari q_1 ke q_0 . Keadaan q_1 merupakan keadaan akhir (final state) karena terdapat dalam set F .

5. $ER = a(a^* U b^*) b$

aturan produksinya:

- 1) simbol awal : a
- 2) Aturan Produksi untuk simbol a : dapat diterjemahkan menjadi a
- 3) Aturan Produksi untuk simbol b : dapat diterjemahkan menjadi b
- 4) Aturan Produksi untuk simbol $*$: menandakan bahwa simbol sebelumnya dapat diulang sebanyak yg diinginkan
- 5) kondisi akhir : simbol b

Contoh kata yg diterima oleh tata bahasa ini adalah "ab", "aabb", "aaabbbbab" dan sebagainya.

Contoh kata yg ditolak oleh tata bahasa ini adalah "a", "b", "abbb" dan sebagainya.

6. Kita bisa mengonstruksi aturan produksi untuk otomata tersebut:

$$T = \{a, b\}$$

$$S = S$$

Kumpulan aturan produksinya kita buat sebagai berikut:

$$S \rightarrow aA \mid bB$$

(identifikasi S untuk q_0 , A untuk q_1 , B untuk q_4)

$$A \rightarrow bC$$

(identifikasi C untuk q_2)

$$C \rightarrow aS$$

(dari q_2 mendapatkan input a ke q_0)

(dari q_3 tidak ada transisi keluar dan bukan state akhir maka transisi ke q_3 diabaikan)

$$B \rightarrow bD$$

(identifikasi D untuk q_5)

$$D \rightarrow bS$$

(dari q_5 mendapatkan input b ke q_0)

(dari q_6 tidak ada transisi keluar dan bukan state akhir maka transisi ke q_6 diabaikan)

Pada kasus ini kita lihat q_0 sebagai state akhir masih memiliki transisi keluar, maka untuk menandaikannya sebagai state akhir kita buat:

$$S \rightarrow \epsilon$$

Maka diperoleh:

$$V = \{S, A, B, C, D\}$$

$$P = \{S \rightarrow aA \mid bB \mid \epsilon, A \rightarrow bC, B \rightarrow bD, C \rightarrow aS, D \rightarrow bS\}$$

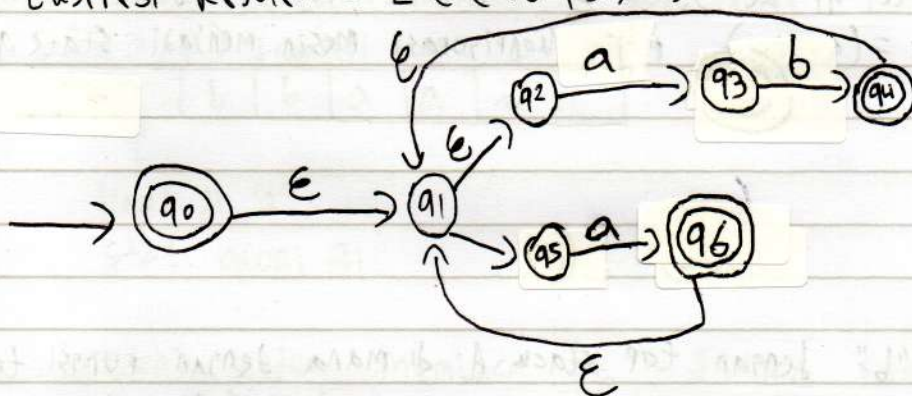
Nama: Andri Firman Saputra

NIM: 201011402125

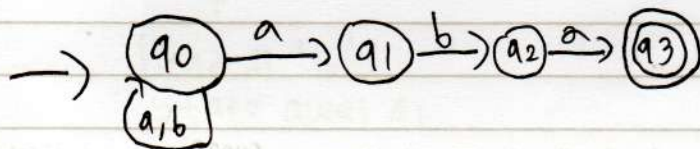
Teori Bahasa dan Automata

UAS Lembar 2

7. Ekspresi Reguler = $L((ab+ba)^*)$



8. Ekspresi Reguler = $L((a+b)^*aba)$



9. Tupel $\rightarrow Q = \{q_1, q_2\}$

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

$S = \{q_1\}$

$F = \{q_2\}$

$Z = \{z\}$

Dengan fungsi transisi sebagai berikut:

1. $\delta(q_1, \epsilon, z) = (q_2, z)$

2. $\delta(q_1, a, z) = (q_1, Az)$

3. $\delta(q_1, b, z) = (q_1, Bz)$

4. $\delta(q_1, a, Az) = (q_1, AAz)$

5. $\delta(q_1, b, Az) = (q_1, \epsilon)$

6. $\delta(q_1, a, Bz) = (q_1, \epsilon)$

7. $\delta(q_1, b, Bz) = (q_1, BBz)$

Uji inputan string "abba" (diterima atau ditolak)

\rightarrow Stack awal \Rightarrow



Konfigurasi awal mesin state q_1 ke stack z Membaca input "a" di mana dengan fungsi transisinya $\{q_1, a, z\} = (q_1, A, z)$, maka konfigurasi mesin menjadi state q_1, A di Push.

Push A \Rightarrow

A
z

Kemudian, terbaca inputan "b" dengan top stack A di mana dengan fungsi transisinya $\{q_1, b, A\} = (q_1, \epsilon)$, maka konfigurasi mesin menjadi state q_1, A di POP

POP A \Rightarrow

z

Kemudian terbaca inputan "b" dengan top stack z di mana fungsi transisinya $\{q_1, b, z\} = (q_1, B, z)$, maka B di Push.

Push B \Rightarrow

B
z

Kemudian terbaca inputan "a" dengan top stack B dimana fungsi transisinya $\{q_1, a, B\} = (q_1, \epsilon)$, maka B di POP

POP B \Rightarrow

z

Karena tidak ada inputan atau ϵ (epsilon) maka fungsi transisinya $\{q_1, \epsilon, z\} = (q_2, z)$, maka sekarang berada di q_2 dengan top stacknya z, maka string "abba" diterima.

10. $Q = \{q_1, q_2\}$

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

$r = \{a, b, \epsilon\}$

$S = \{q_1\}$

$F = \{q_2\}$

Dengan fungsi transisi

1. $\delta\{q_1, a\} = (q_1, a, R)$

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

3. $\delta\{q_1, \epsilon\} = (q_2, \epsilon, L)$

Uji string "abba" (diterima atau ditolak)

- 1) Langkah pertama membaca "a" dengan fungsi transisi $\delta\{q_1, a\} = (q_1, a, R)$ transisi ke-1

b	a	b	b	a	a	b	
---	---	---	---	---	---	---	--



Head di 'a'

State awal q_1

- 2) Langkah kedua membaca "b" dengan fungsi transisi $\delta\{q_1, b\} = (q_1, a, R) \rightarrow$ transisi ke-2

b	a	b	b	a	a	b	
---	---	---	---	---	---	---	--



Head di 'b'

State awal q_1

- 3) Langkah ketiga membaca "b" dengan fungsi transisi $\delta\{q_1, b\} = (q_1, a, R) \rightarrow$ transisi ke-2, maka gambaran deretan pita menjadi:

b	a	a	b	a	a	b	
---	---	---	---	---	---	---	--



Head 'b'

State awal q_1

- 4) Langkah keempat membaca string "a" dengan fungsi transisi $\delta\{q_1, a\} = (q_1, a, R) \rightarrow$ fungsi transisi ke-1

b	a	a	a	a	a	b	
---	---	---	---	---	---	---	--



Head 'a'

State awal q_1

- 5) Langkah kelima membaca string "a" dengan fungsi transisi $\delta\{q_1, a\} = (q_1, a, R) \rightarrow$ fungsi transisi ke-1

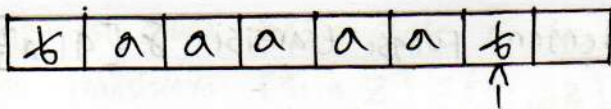
b	a	a	a	a	a	b	
---	---	---	---	---	---	---	--



Head 'a'

State awal q_1

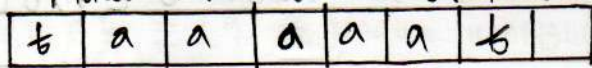
- 6) Langkah keenam dimana pita (deretan) ke-5 mengarah pada pita blank (b) di mana fungsi transisi $\delta\{q_1, b\} = (q_2, b, L) \rightarrow$ fungsi transisi ke-3



Head menunjukan b dan bergerak ke kiri maka hasil deretannya menjadi Langkah 7.

7) Langkah ketujuh deretan pita mengarah pada fungsi transisi 8 {q2, a} di mana fungsi ini tidak transisi lagi dari state q2 (final state)

maka mesin turing berhenti:



Head 'a'

State Final q2

-) Kesimpulan karena pita (deretan) berhenti di final state (q2) maka string "abbaaa" diterima oleh mesin turing.