QUIZ Teori Bahasa dan Otomata

Semester Ganjil 2012/2013

Jurusan Informatika

UII

1. Diketahui, definisi bahasa sebagai berikut **(Bobot 20).**

Σ = {0,1}

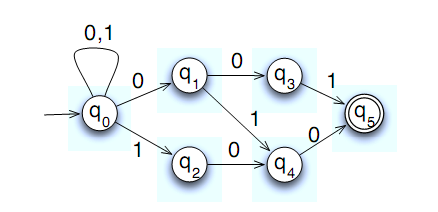
L = {x10y | x, y ϵ Σ \*}

Buatlah DFA yang menerima bahasa L!

* L = {x10y | x, y ϵ Σ \*} adalah bahasa yang menerima seluruh string biner yang memiliki substring 10.
* Membuat DFA bisa dimulai dengan membuat NFA-nya terlebih dahulu kemudian dikonversi menjadi DFA.

|  |  |
| --- | --- |
| NFA | DFA |
| D:\uii\!Workshop\TBO\jawaban quiz-1-a.png | D:\uii\!Workshop\TBO\jawaban quiz-1.png |

1. Diketahui suatu NFA:



Ubahlah menjadi DFA! **(Bobot 20)**

Jawab:

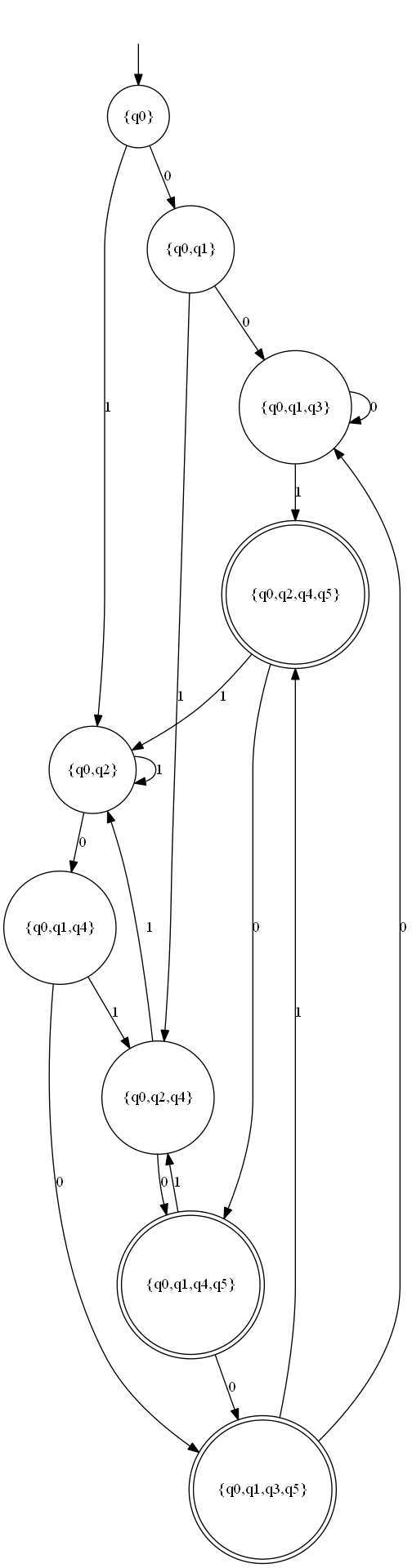
**Tabel Transisi NFA**

|  |  |  |
| --- | --- | --- |
|  | 0 | 1 |
| 🡪q0 | {q0,q1} | {q0,q2} |
| q1 | {q3} | {q4} |
| q2 | {q4} | Ø |
| q3 | Ø | {q5} |
| q4 | {q5} | Ø |
| \*q5 | Ø | Ø |

**Langkah membuat DFA dari Tabel NFA**

* 1. Cari fungsi transisinya mulai dari start state NFA yaitu q0
  2. Lanjutkan pencarian fungsi transisi untuk setiap status baru yang terbentuk
  3. Final state DFA adalah seluruh status yang mengandung status final NFA (q5)
  4. Ubah tabel transisi NFA menjadi diagram otomata DFA

|  |  |  |
| --- | --- | --- |
|  | 0 | 1 |
| 🡪{q0} | {q0,q1} | {q0,q2} |
| {q0,q1} | { q0,q1,q3} | { q0,q2,q4} |
| {q0,q2} | { q0,q1,q4} | {q0,q2} |
| { q0,q1,q3} | { q0,q1,q3} | {q0,q2,q4,q5} |
| { q0,q2,q4} | { q0,q1,q4,q5} | {q0,q2} |
| { q0,q1,q4} | { q0,q1,q3,q5} | { q0,q2,q4} |
| \*{q0,q2,q4,q5} | { q0,q1,q4,q5} | {q0,q2} |
| \*{q0,q1,q4,q5} | { q0,q1,q3,q5} | { q0,q2,q4} |
| \*{ q0,q1,q3,q5} | { q0,q1,q3} | {q0,q2,q4,q5} |



1. Diberikan ε-NFA sebagai berikut:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Ɛ | 0 | 1 |
| 🡪p | {q, r} | {q} | {r} |
| q | {r} | {p} | {p, r} |
| \*r | Ø | Ø | Ø |

Carilah eclose untuk setiap state yang ada. **(Bobot 5)**

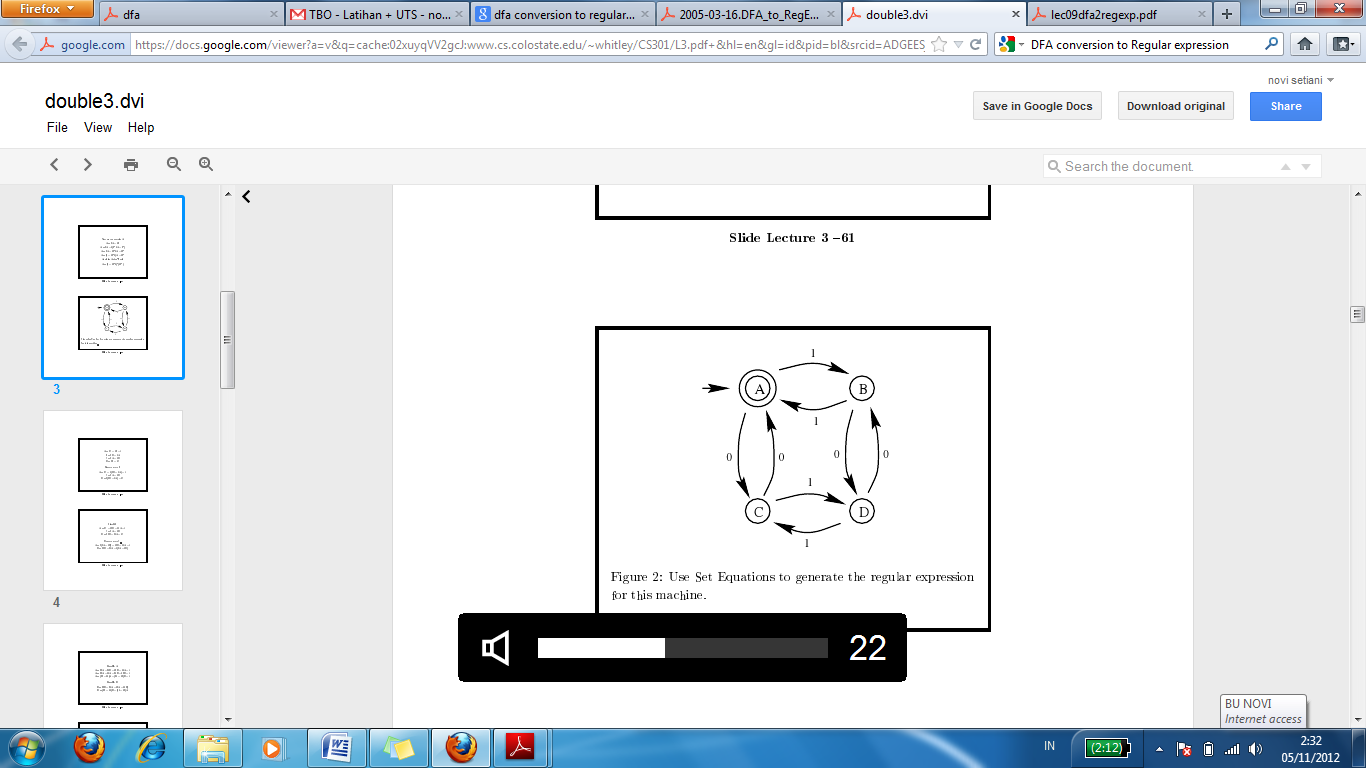
Jawab:

Eclose (p) = {p,q,r}

Eclose (q) = (q,r)

Eclose (r) = {r}

1. Konversi DFA di bawah ini menjadi Regular Expression dengan menggunakan eliminasi status atau induksi k-path **(Bobot 20)**.



**Eliminasi status:**

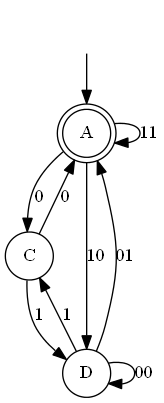
1. Eliminasi status B

A-B-A : 11

D-B-D : 00

A-B-D:10

D-B-A:01

****

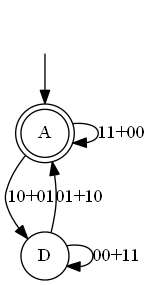
1. Eliminasi status C

A-C-A : 00

D-C-D : 11

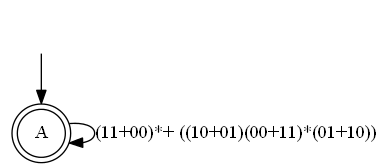
A-C-D:01

D-C-A:10

****

1. Eliminasi status D

A-D-A : (10+01)(00+11)\*(01+10)

****

Reguler ekspresi yang dihasilkan dengan menggunakan eliminasi status adalah (A-A)\* yaitu : ((11+00)\*+((10+01)(00+11)\*(01+10)))\*

**Induksi k-path:**

Misal,

status A = 1

status B = 2

status C = 3

status D = 4

Untuk melakukan konversi DFA menjadi Ekspresi Reguler dengan menggunakan induksi k-path, maka cari di mana, i = status awal (1), j = status final (1) dan k = jumlah status. Sehingga untuk soal ini, yang dicari adalah:

1. Basis, k=0

|  |  |
| --- | --- |
|  |  |
|  | ∅ + Ɛ = Ɛ |
|  | 1 |
|  | 0 |
|  | ∅ |
|  | 1 |
|  | ∅ + Ɛ = Ɛ |
|  | ∅ |
|  | 0 |
|  | 0 |
|  | ∅ |
|  | ∅ + Ɛ = Ɛ |
|  | 1 |
|  | ∅ |
|  | 0 |
|  | 1 |
|  | ∅ + Ɛ = Ɛ |

1. Induksi, k = 1

Rumus :

Substitusi nilai i dan j untuk setiap status, lalu gunakan hasil dari Basis (k=0).

|  |  |  |
| --- | --- | --- |
|  | Substitusi | Penyederhanaan |
|  | Ɛ + Ɛ(Ɛ)\* Ɛ | Ɛ |
|  | 1 + Ɛ(Ɛ)\*1 | 1 |
|  | 0 + Ɛ(Ɛ)\*0 | 0 |
|  | ∅ + Ɛ(Ɛ)\* ∅ | ∅ |
|  | 1 + 1(Ɛ)\* Ɛ | 1 |
|  | Ɛ + 1(Ɛ)\*1 | Ɛ +11 |
|  | ∅ + 1(Ɛ)\*0 | 10 |
|  | 0 + 1(Ɛ)\* ∅ | 0 |
|  | 0 + 0(Ɛ)\*Ɛ | 0 |
|  | ∅ + 0(Ɛ)\*1 | 01 |
|  | Ɛ + 0(Ɛ)\*0 | Ɛ + 00 |
|  | 1 + 0(Ɛ)\*∅ | 1 |
|  | ∅ + ∅(Ɛ)\*Ɛ | ∅ |
|  | 0 + ∅(Ɛ)\*1 | 0 |
|  | 1 + ∅(Ɛ)\*0 | 1 |
|  | Ɛ + ∅(Ɛ)\*∅ | Ɛ |

1. Induksi, k =2

Rumus :

Substitusi nilai i dan j untuk setiap status, lalu gunakan hasil dari Induksi k=1.

|  |  |  |
| --- | --- | --- |
|  | Substitusi | Penyederhanaan |
|  | Ɛ + 1(Ɛ+11)\* 1 | Ɛ + 1(11)\*1 |
|  | 1 + 1(Ɛ+11)\* (Ɛ+11) | 1(11)\* |
|  | 0 + 1(Ɛ+11)\*10 | 0+1(11)\*10 |
|  | ∅ + 1(Ɛ+11)\*0 | 1(11)\*0 |
|  | 1 + (Ɛ+11) (Ɛ+11)\* Ɛ | 1+(11)\* |
|  | Ɛ + 1(Ɛ+11)\*1 | 1(11)\*1 |
|  | ∅ + 1(Ɛ+11)\*0 | 1(11)\*0 |
|  | 0+(Ɛ+11) (Ɛ+11)\*0 | (11)\*0 |
|  | 0+01(Ɛ+11)\*1 | 0+01(11)\*1 |
|  | 01+01(Ɛ+11)\* (Ɛ+11) | 01(11)\* |
|  | Ɛ + 00+01(Ɛ+11)\*10 | Ɛ + 00+01(11)\*10 |
|  | 1+01(Ɛ+11)\*0 | 1+01(11)\*0 |
|  | ∅+0(Ɛ+11)\*1 | 0(11)\*1 |
|  | 0+0(Ɛ+11)\* (Ɛ +11) | 0(11)\* |
|  | 1+0(Ɛ+11)\*10 | 1+0(11)\*10 |
|  | Ɛ+0(Ɛ+11)\*0 | Ɛ+0(11)\*0 |

1. Induksi, k = 3

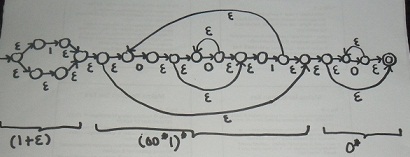
Substitusi nilai i dan j untuk setiap status, lalu gunakan hasil dari Induksi k=2.

|  |  |  |
| --- | --- | --- |
|  | Substitusi | Penyederhanaan |
|  | (Ɛ + 1(11)\*1)+ (0+1(11)\*10)( Ɛ + 00+01(11)\*10)\* 01(11)\* | (Ɛ + 1(11)\*1)+ (0+1(11)\*10)( 00+01(11)\*10)\* 01(11)\* |
|  | 1(11)\*0+(0+1(11)\*10)( Ɛ + 00+01(11)\*10)\*( 1+01(11)\*0) | 1(11)\*0+(0+1(11)\*10)( 00+01(11)\*10)\*( 1+01(11)\*0) |
|  | (Ɛ+0(11)\*0)+(1+0(11)\*10)( Ɛ + 00+01(11)\*10)\*( 1+01(11)\*0) | Ɛ+0(11)\*0)+(1+0(11)\*10)( 00+01(11)\*10)\*( 1+01(11)\*0) |
|  | 0(11)\*1+(1+0(11)\*10)( Ɛ + 00+01(11)\*10)\* 01(11)\* | 0(11)\*1+(1+0(11)\*10)(00+01(11)\*10)\* 01(11)\* |

Silakan masukkan hasil induksi k=3 ke dalam rumus sebagai reguler ekspresi final hasil konversi.

1. Ubahlah ekspresi reguler berikut menjadi ε-NFA **(Bobot 10)**:

(1 + ε)(00\*1)\*0\*



1. Buktikan apakah bahasa berikut adalah bahasa reguler atau bukan dengan menggunakan pumping lemma.
2. {0n1m | n<=m} **(bobot 15)**

Jawab:

Bahasa di atas termasuk bahasa non reguler, gunakan string yang termasuk {0n1m | n<=m} untuk n=m, seperti 01, 0011, 000111,....

1. L((0+1) \*0) = {0,00,10,100,110,010,000, ....} **(bobot 10)**

Langkah pembuktian dengan menggunakan pumping lemma:

1. Misal, konstanta n = 3
2. Ambil string w ϵ L, |w| >= n

Misal, w = 110

1. Pecah w menjadi xyz, di mana |xy| <= n dan y ≠ Ɛ

Misal, x=1, y=1, z=0

1. Lakukan pumping xykz,

k = 0 🡪 1100 = 10

k = 1 🡪 1110 = 110

k = 2 🡪 1120 = 1110

k = 3 🡪 1130 = 11110

Oleh karena hasil pumping lemma termasuk ke dalam bahasa L, maka L adalah bahasa reguler.