OTOMATA 5

MASALAH REGULARITAS

MATERI PERTEMUAN

- Bahasa Regular
- Observasi pada DFA
- Bahasa Non Regular
- Tugas Mingguan VIII

Bahasa Regular (1)

Bahasa Regular → Bahasa yang dapat didefinisikan melalui Regular Expression (RE)

Tetapi, bukankah dalam mendefinisikan RE, kita dapat memanfaatkan operasi^2 himpunan seperti UNION, CONCATENATION, dan CLOSURE ?!

Oleh karenanya, operasi^2 himpunan itu harus dapat pula diterapkan pada Transition Graph yang berasosiasi dengan RE tersebut

Bahasa Regular (2)

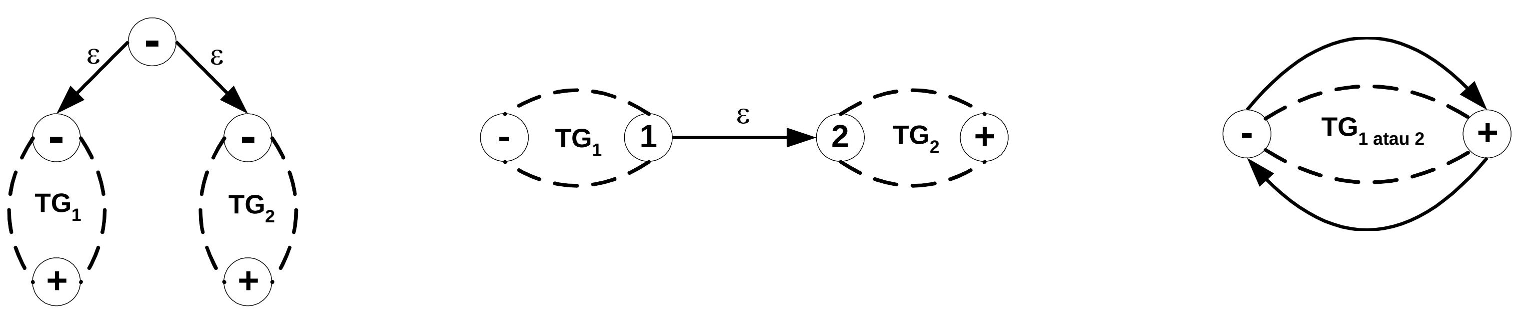
Teorema

Jika L_1 dan L_2 masing2 adalah bahasa regular, maka bahasa2 sbg hasil operasi L_1+L_2 , $L_1 L_2$ dan L_1^* serta L_2^* adalah bahasa regular

Bukti

a. Representasi bahasa regular L adalah melalui RE. Dan RE dapat menerima operasi himpunan. Sehingga hasilnya pun adalah bahasa regular.

b. Selain RE, bahasa regular L dapat pula direpresentasikan dengan Transition Graph (TG). Dan TG pun harus dapat menerima operasi himpunan. Sehingga hasil operasi tsb juga berupa bahasa regular (ini yg perlu dibuktikan)



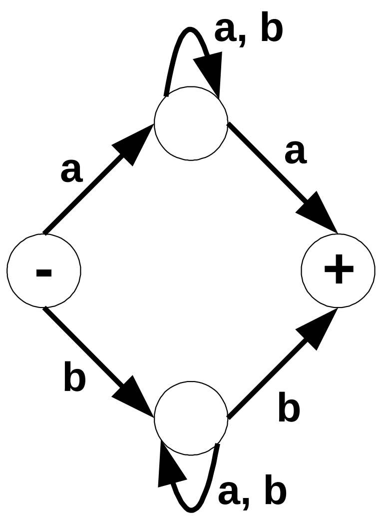
Bahasa Regular (3)

Contoh :

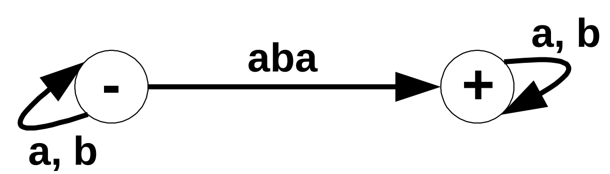
$R_1 = a\,(a + b)^*\,a \;+\; b\,(a + b)^*\,b$

$R_2 = (a + b)^*\,aba\,(a + b)^*$

TG_1



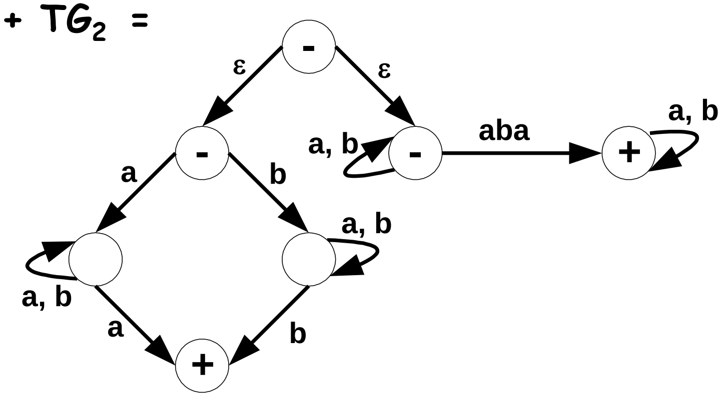
TG_2



Bahasa Regular (4)

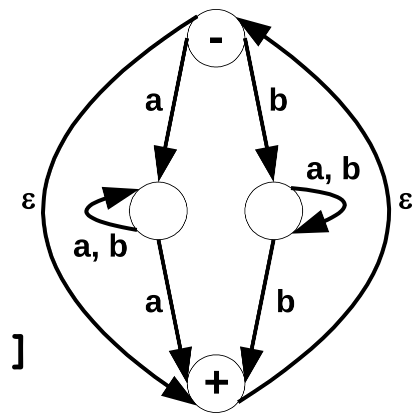
Jika $R_1 + R_2$ = $[\ a\ (a + b)^*\ a\ +\ b\ (a + b)^*\ b\ +\ (a + b)^*\ aba\ (a + b)^*\ ]$

Maka $TG_1 + TG_2$ =



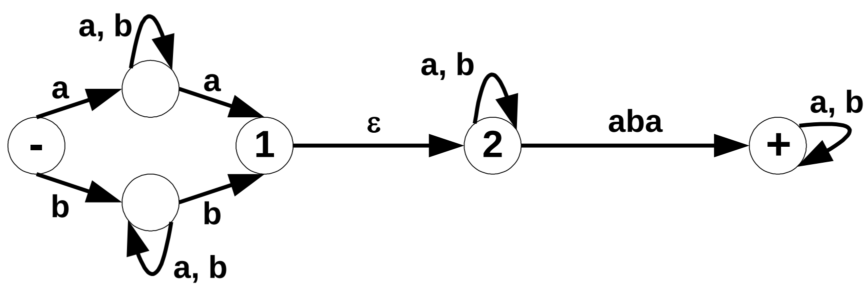
Jika R_1^* = $[\ a\ (a + b)^*\ a\ +\ b\ (a + b)^*\ b\ ]^*$

Maka TG_1^* =



Jika $R_1\ R_2$ = $[\ a\ (a + b)^*\ a\ +\ b\ (a + b)^*\ b\ ][\ (a + b)^*\ aba\ (a + b)^*\ ]$

Maka $TG_1\ TG_2$ =



Bahasa Regular (5)

Teorema

Jika L adalah bahasa regular, maka komplemen dari L (L^1) juga bahasa regular

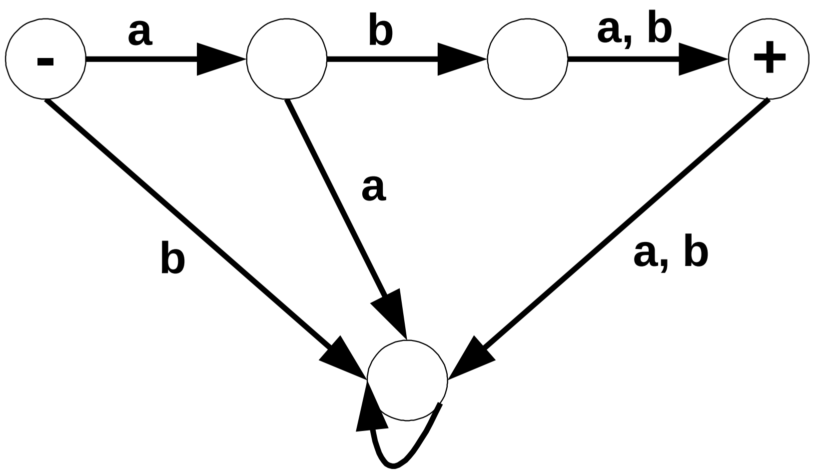
Bukti

komplemen dari bahasa L, adalah bahasa yang menerima semua string, SELAIN string yang diterima oleh bahasa L.

Implementasi operasi komplemen pada DFA adalah dengan menjadikan semua final state menjadi NON final state. dan sebaliknya.

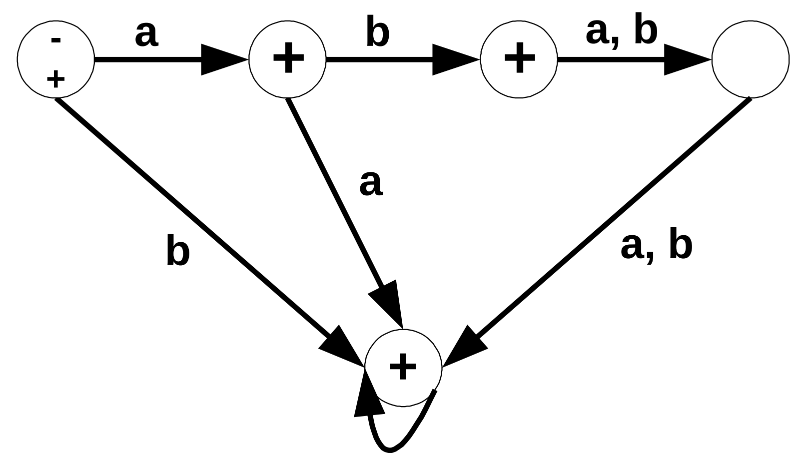
Bahasa Regular (6)

DFA bahasa L = { aba, abb } dapat digambarkan sbb :



Maka L^1 = { semua string kecuali aba dan abb }.

DFA untuk L^1 adalah spt berikut :



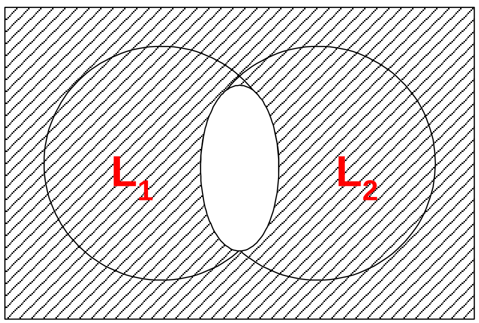
Bahasa Regular (7)

Teorema Jika L_1 dan L_2 masing-masing adalah bahasa regular, maka $L_1 \cap L_2$ adalah juga bahasa regular

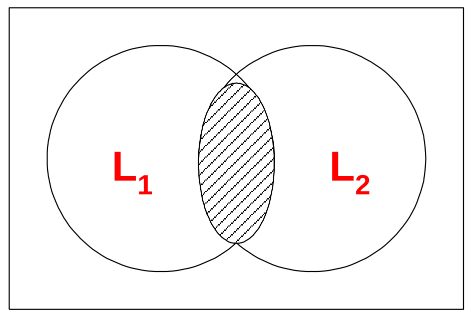
Bukti Pada sebarang himpunan berlaku hukum De Morgan :

$$L_1 \cap L_2 = (L_1{}^1 + L_2{}^1)^1$$

Dan hal ini dapat dijelaskan melalui Diagram Venn :



$(L_1{}^1 + L_2{}^1)$

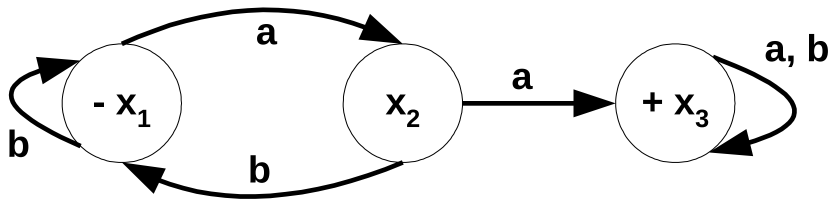


$(L_1{}^1 + L_2{}^1)^1 = L_1 \cap L_2$

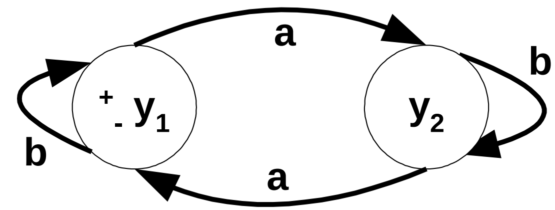
Bahasa Regular (8)

Contoh :

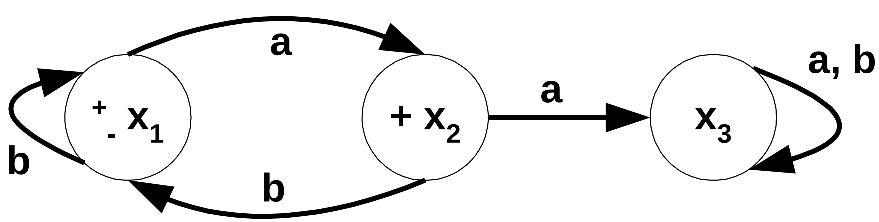
DFA$_1$:



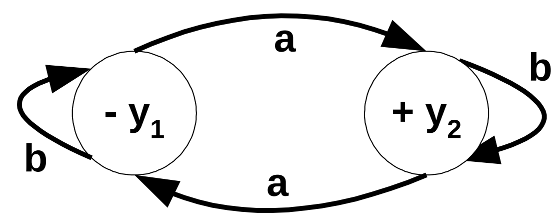
DFA$_2$:



DFA$_1{}^1$:



DFA$_2{}^1$:

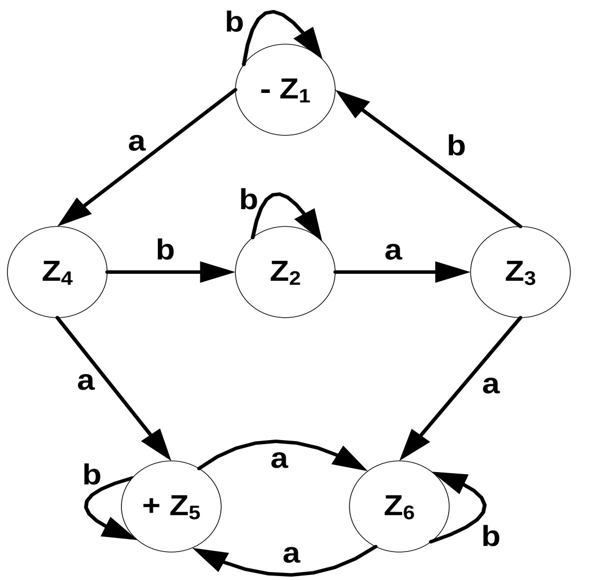


Bahasa Regular (9)

Untuk menyelesaikan masalah irisan, kita dapat melakukan dengan me-union-kan komplemen DFA[1] dan DFA[2], untuk kemudian hasilnya di-komplemen-kan lagi.

$DFA_1 + DFA_2$:

	a	b	State Asal
- Z_1	Z_4	Z_1	$X_1 \mid Y_1$
Z_2	Z_3	Z_2	$X_1 \mid Y_2$
Z_3	Z_6	Z_1	$X_2 \mid Y_1$
Z_4	Z_5	Z_2	$X_2 \mid Y_2$
+ Z_5	Z_6	Z_5	$X_3 \mid Y_1$
Z_6	Z_5	Z_6	$X_3 \mid Y_2$



Observasi pada DFA (1)

Ada 3 macam observasi yang dapat kita lakukan pada DFA :

Observasi I

Kita dapat meneliti apakah sebuah DFA dapat menerima/mendefinisikan bahasa atau tidak.

Observasi II

Kita dapat menguji apakah 2 atau lebih DFA mendefinisikan bahasa yang sama atau tidak.

Observasi III

Kita dapat menentukan apakah sebuah DFA atau RE mendefinisikan bahasa yang berhingga (finite) atau tidak berhingga (infinite).

Observasi pada DFA (2)

Observasi I

Metode **Effective Decision Procedures** dapat digunakan untuk meneliti apakah sebuah DFA menerima/mendefinisikan bahasa. Metode dapat dilakukan melalui 2 pendekatan :

Pendekatan I

1. Konversikan DFA ke dalam bentuk RE;
2. Hilangkan semua closure;
3. Hilangkan semua simbol UNION (+) dan semua substring di sebelah kanan simbol UNION tersebut;
4. Hilangkan simbol "(" dan ")";
5. Rangkailah substring yang tersisa menjadi sebuah string yang utuh;

Contoh :

RE	$= (a + \lambda)(ab^* + ba^*)^*(\lambda + b^*)^*$
Eliminasi closure	$= (a + \lambda)(ab + ba)(\lambda + b)$
Eliminasi " + substring "	$= (a)(ab)(\lambda)$
Eliminasi " (" dan ") "	$= a\ ab\ \lambda$
String minimal	$= aab$

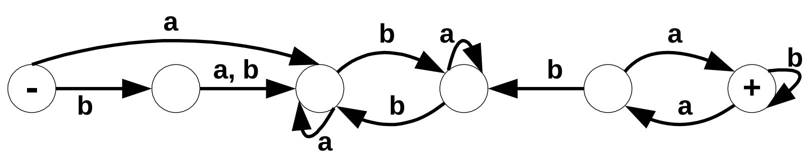
Observasi pada DFA (3)

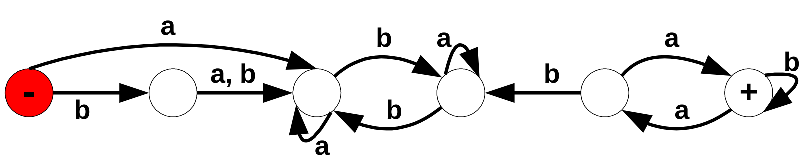
Pendekatan II

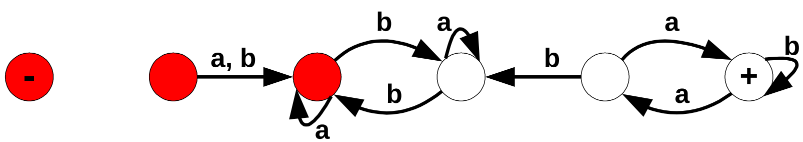
Mengevaluasi apakah terdapat setidaknya satu path dari start state menuju final state.

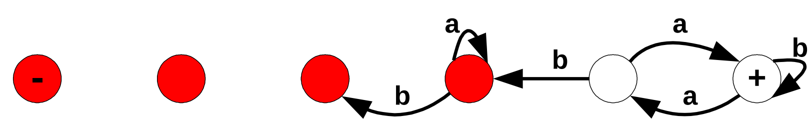
1. Tandailah start state.
2. Ikutilah outgoing edge dari state bertanda untuk mencari adjacent state, kemudian tandailah adjacent state tsb dan hapuslah outgoing edge-nya.
3. Ulangi langkah 2 sampai tidak dapat diimplementasikan lagi.
4. Periksalah apakah terdapat final state di antara state-state bertanda. Jika ya, maka DFA tsb dapat menerima string.

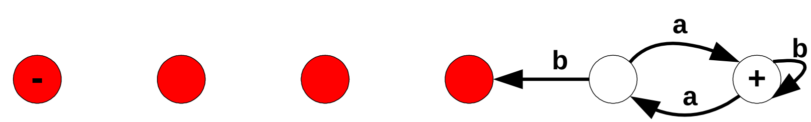
Contoh :











Observasi pada DFA (4)

Observasi II

Untuk menguji similaritas DFA dapat digunakan metode sbb :

$$(L_1 \cap L_2^{1}) + (L_2 \cap L_1^{1})$$

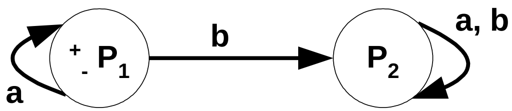
Sebuah mesin dapat menerima semua input string di L_1, tetapi tidak di L_2 (atau, sebaliknya, menerima semua input string di L_2 tetapi tidak di L_1)

Dan jika L_1 dan L_2 adalah bahasa yang sama/ekivalen, maka seharusnya mesin di atas tidak dapat menerima input string apapun !!!

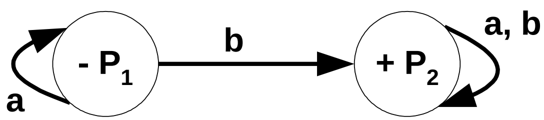
Observasi pada DFA (5)

Contoh :

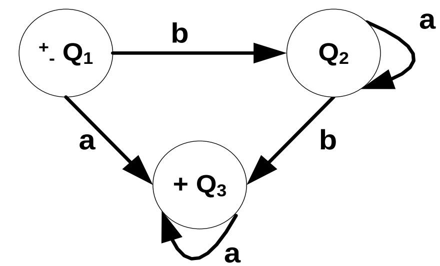
DFA$_1$:



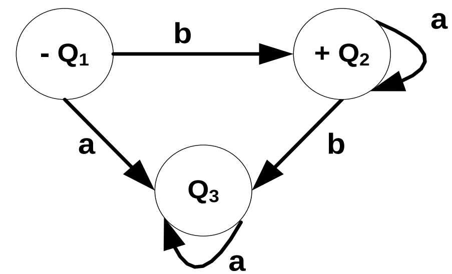
DFA$_1{}^1$:



DFA$_2$:



DFA$_2{}^1$:



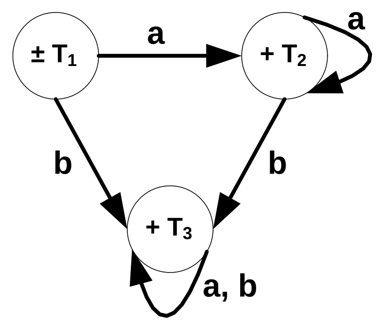
$$(L_1 \cap L_2{}^1) + (L_2 \cap L_1{}^1) = (L_1{}^1 + L_2)^1 + (L_2{}^1 + L_1)^1$$

Observasi pada DFA (6)

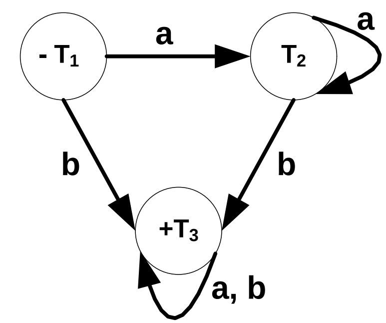
$DFA_1^1 + DFA_2$

	a	b	
$\pm T_1$	T_2	T_3	$P_1 \mid Q_1$
$+ T_2$	T_2	T_3	$P_1 \mid Q_2$
T_3	T_3	T_3	$P_2 \mid Q_2$

$DFA_1^1 + DFA_2$



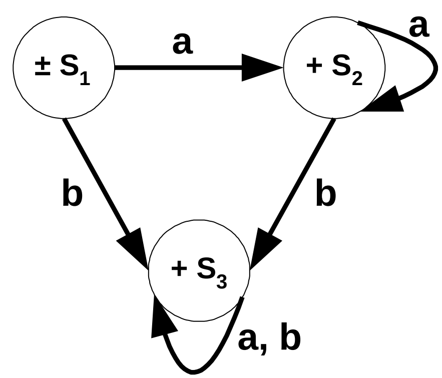
$(DFA_1^1 + DFA_2)^1$



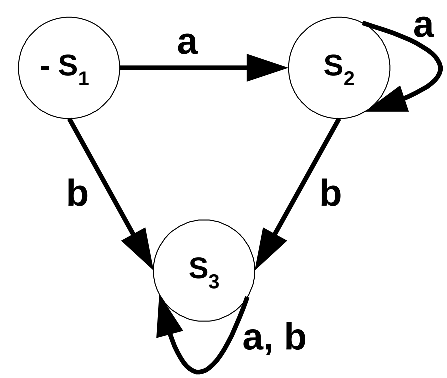
$DFA_2^1 + DFA_1$

	a	b	
$\pm S_1$	S_2	S_3	$Q_1 \mid P_1$
$+ S_2$	S_2	S_3	$Q_2 \mid P_1$
$+ S_3$	S_3	S_2	$Q_2 \mid P_2$

$DFA_2^1 + DFA_1$



$(DFA_2^1 + DFA_1)^1$



$(DFA_1^1 + DFA_2)^1 + (DFA_2^1 + DFA_1)^1$

Bahasa Non Regular (1)

Bahasa yang TIDAK DAPAT didefinisikan melalui Regular Expression disebut Bahasa Non Regular

Contoh :

1. $L = \{ \lambda, ab, aabb, aaabbb, \ldots \}$
   $L = \{ a^n b^n$, dengan $n = 0, 1, 2, 3, \ldots \}$

2. Bahasa Prima = { aa, aaa, aaaaa, . . . }
   Bahasa Prima = $\{a^n$ dengan n adalah bilangan prima $\}$

Identifikasi sifat non regular pada sebuah bahasa dapat menggunakan PUMPING LEMMA.

Lemma ini akan mem-'pompa'-kan sejumlah substring ke dalam string tertentu.

Bahasa Non Regular (2)

PUMPING LEMMA

Misal L adalah sebarang bahasa regular yang memiliki himpunan string tak berhingga. String-string pada bahasa regular tersebut dapat dikelompokkan menjadi 3 substring X, Y, dan Z (dimana Y bukan merupakan null string), sedemikian hingga ketiga substring di atas dapat dikembangkan menjadi string-string baru yang berbentuk :

$$X\ Y^n\ Z \text{ untuk } n = 1,\ 2,\ 3,\ .\ .\ .$$

dimana string di atas adalah anggota dari bahasa L.

Bahasa Non Regular (3)

Contoh :

Buktikan bahwa bahasa L = {$a^n b^n$, untuk n = 0, 1, 2, . . .} adalah bahasa non regular.

Asumsikan bahwa bahasa L adalah bahasa regular, sehingga string-stringnya dapat dikelompokkan ke dalam subtring XYZ.

Misal kita ambil salah satu string: aabb.

kemudian kita misalkan X = a, Y = ab, dan Z = b.

Multiplikasi pada substring Y akan menghasilkan string-string baru:

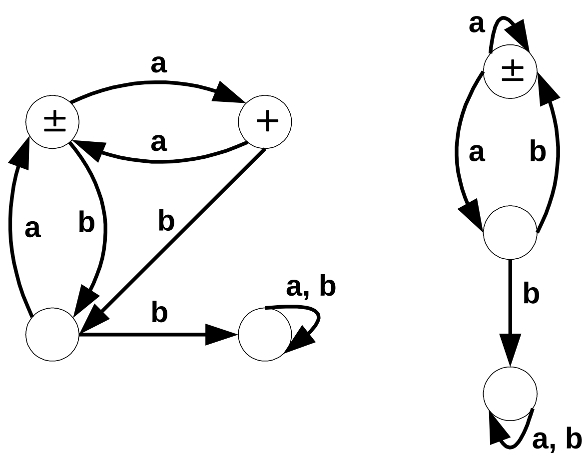
XYZ = a ab b, XYYZ = a abab b, XYYYZ = a ababab b, . . .

Bisa dilihat bahwa string-string baru di atas bukan anggota dari bahasa L.

Kesimpulannya Pumping Lemma tidak dapat diterapkan pada bahasa L, sehingga bahasa L adalah bahasa non regular.

Tugas

1. Tunjukkan apakah kedua DFA berikut ekivalen :



Tentukan apakah kedua bahasa di bawah bersifat regular atau non regular :

2. Kuadrat Ganda = { $a^{n^2}b^n$, dengan n = 1, 2, 3, ...}
3. { $a^n b^{n+1}$ } = { abb, aabbb, aaabbbb, ...}

4. Apakah DFA berikut dapat menerima string ?