MODUL 6: TEOREMA KLEENE

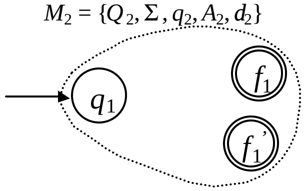
Dari pembahasan sebelumnya NFA-Λ yang dapat mengenali suatu bahasa regular dapat dengan lebih langsung diperoleh karena adanya transisi-Λ. Setelah NFA-Λ yang dapat mengenali bahasa tersebut diperoleh maka suatu NFA, dan seterusnya juga suatu FA, dapat kemudian diperoleh.

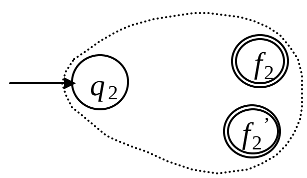
Dalam contoh-contoh yang dibahas transisi-Λ digunakan untuk merealisasikan operasi konkatenasi. Sementara kita ingat juga bahwa suatu suatu bahasa regular dapat dipandang sebagai hasil operasi-operasi penggabungan, konkatenasi dan Kleene * dari bahasa-bahasa paling sederhana dari alfabetnya. Apabila disamping konkatenasi, operasi-operasi itu juga dapat direalisasikan dengan transisi-Λ maka suatu NFA-Λ (dan selanjutnya NFA dan FA) yang dapat mengenali suatu bahasa regular dengan ekspresi regular apa saja dapat dengan mudah ditemukan.

Dalam bagian ini akan metodologi yang lebih lengkap mendapatkan NFA-Λ untuk baik operasi gabungan, irisan, dan Kleene * akan dibahas. (Bandingkanlah dengan metoda yang lebih langsung khusus untuk operasi penggabungan, irisan dan perbedaan pada pembahasan FA).

Misalkan bahasa L_1 dan L_2 dikenali masing-masing oleh NFA-Λ M_1 dan M_2, yaitu:

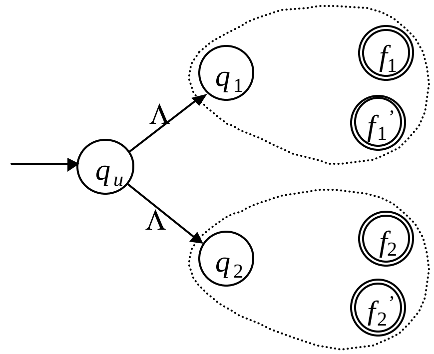
$M_1 = \{Q_1, \Sigma, q_1, A_1, d_1\}$
$M_2 = \{Q_2, \Sigma, q_2, A_2, d_2\}$





Dari kedua mesin tersebut maka NFA-Λ M_u, M_c, dan M_k dapat dibentuk untuk mengenali masing-masing $L_1 \cup L_2$, L_1L_2, dan L_1^*.

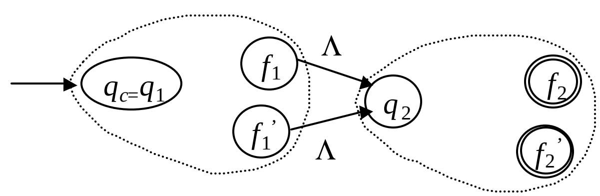
Pembentukan M_u untuk mengenal $L_1 \cup L_2$



$M_u = \{Q_u, \Sigma, q_u, A_u, d_u\}$ dapat dibentuk sebagai berikut.

- q_u adalah status baru yang tidak ada di Q_1 maupun Q_2
- $Q_u = Q_1 \cup Q_2 \cup \{q_u\}$
- $A_u = A_1 \cup A_2$
- d_u dibentuk sbb.
  - untuk setiap $q \in Q_1$ dan $a \in \Sigma \cup \{\Lambda\}$, $d_u(q, a) = d_1(q, a)$
  - untuk setiap $q \in Q_2$ dan $a \in \Sigma \cup \{\Lambda\}$, $d_u(q, a) = d_2(q, a)$
  - untuk setiap $a \in \Sigma$, $d_u(q_u, \Lambda) = \{q_1, q_2\}$ dan $d_u(q_u, a) = \varnothing$

Pembentukan M_c untuk mengenal L_1L_2



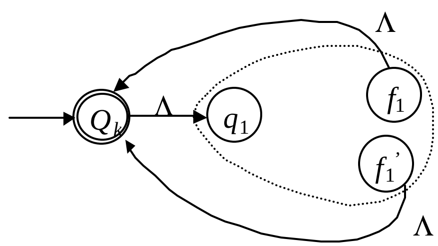
$M_c = \{Q_c, \Sigma, q_c, A_c, d_c\}$ dapat dibentuk sebagai berikut.

- $q_c = q_1$.

- $Q_c = Q_1 \cup Q_2$
- $A_c = A_2$
- d_c dibentuk sbb.
    - untuk setiap $q \in Q_1$ dan $a \in \Sigma \cup \{\Lambda\}$, $d_c(q, a) = d_1(q, a)$, kecuali jika $q \in A_1$ maka ditambahkan $d_c(q, \Lambda) = q_2$
    - untuk setiap $q \in Q_2$ dan $a \in \Sigma \cup \{\Lambda\}$, $d_c(q, a) = d_2(q, a)$

Pembentukan M_k untuk mengenal L_1^*

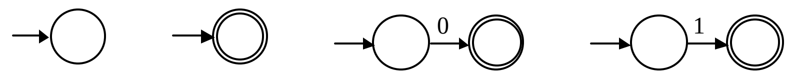
$M_k = \{Q_k, \Sigma, q_k, A_k, d_k\}$ dapat dibentuk sebagai berikut.



- q_k adalah status baru yang tidak ada di Q_1
- $Q_k = Q_1 \cup \{q_k\}$
- $A_k = \{q_k\}$
- d_k dibentuk sbb.
    - untuk setiap $q \in Q_1$ dan $a \in \Sigma$, $d_k(q, \Lambda) = q_1$ dan $d_k(q, a) = \varnothing$
    - untuk setiap $q \in Q_1$ dan $a \in \Sigma \cup \{\Lambda\}$, $d_k(q, a) = d_1(q, a)$, kecuali jika $q \in A_1$ maka ditambahkan $d_k(q, \Lambda) = q_k$
    - $d_u(q_u, \Lambda) = \{q_1, q_2\}$ dan $d_u(q_u, a) = \varnothing$ untuk setiap $a \in \Sigma$

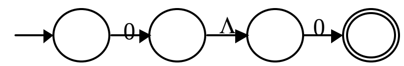
Dengan demikian setiap bahasa dapat dipandang sebagai tersusun atas bahasa-bahasa sederhana melalui operasi-operasi penggabungan, konkatenasi dan Kleene*.

Contoh: untuk setiap bahasa $L \subseteq \{0,1\}^*$ dapat dibuat NFA-Λ yang dapat mengenalnya dan tersusun atas FA-FA untuk bahasa-bahasa regular paling sederhana sbb. Masing-masing FA berikut untuk bahasa $\varnothing$, $\{\Lambda\}$, $\{0\}$, dan $\{1\}$.

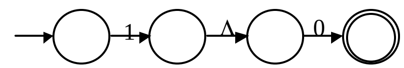


Contoh: Seandainya untuk bahasa dengan ekspresi regular $(00+1)^*(10)^*$ akan ditemukan NFA-Λ yang dapat mengenalnya sebagai bentukan bertahap mulai dari FA bahasa-bahasa paling sederhananya.

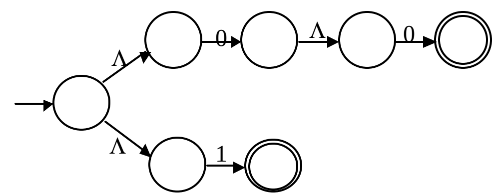
- membentuk NFA-Λ bahasa {00} berdasarkan konkatenasi {0} dan {0}.



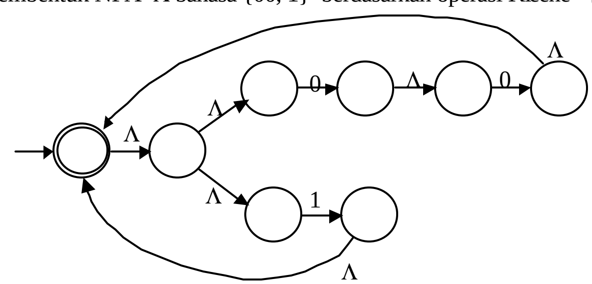
- membentuk NFA-Λ bahasa {10} berdasarkan konkatenasi {1} dan {0}.



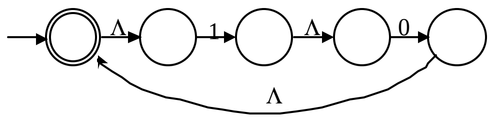
- membentuk NFA-Λ bahasa {00, 1} berdasarkan penggabungan {00} dan {1}.



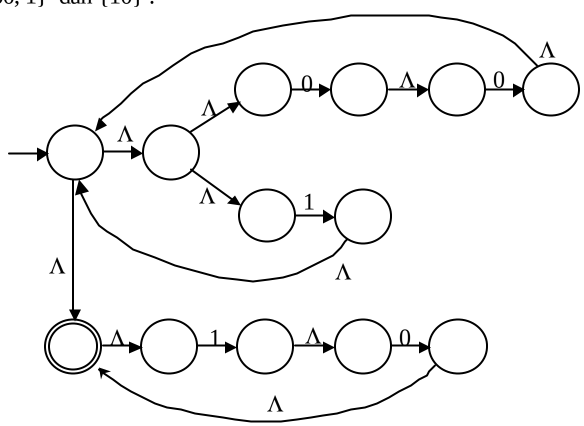
- membentuk NFA-Λ bahasa $\{00, 1\}^*$ berdasarkan operasi Kleene* {00, 1}.



- membentuk NFA-Λ bahasa $\{10\}^*$ berdasarkan operasi Kleene* {10}.



- membentuk NFA-Λ bahasa $\{00, 1\}^*\{10\}^*$ berdasarkan operasi konkatenasi $\{00, 1\}^*$ dan $\{10\}^*$.



Dalam contoh tersebut selama pembentukan sama sekali tidak dilakukan penyederhanaan untuk mengurangi transisi-Λ. Penyederhanaan -dapat dilakukan sehingga diagram yang diperoleh tidak penuh oleh transisi-Λ tersebut dengan mengaplikasikan konversi NFA-Λ ke NFA pada setiap tahapan pembentukan.

Mendapatkan Ekspresi Regular dari $L(M)$ dengan M Suatu FA

Untuk menjelaskan teknik untuk dapat melakukan hal ini memerlukan pendefinisian sbb.

- $L(p, q) = \{x \in \Sigma^* \mid d^*(p, x) = q\}$ yaitu himpunan string yang mentransisikan M dari status p ke q.

Dengan definisi tersebut maka $L = L(M)$ adalah gabungan semua $L(p, q)$ dimana p adalah status inisial q_0, dan $q \in A$. Dituliskan

$$L = \bigcup_{q \in A} L(q_0, q)$$

Jika ekspresi regular dari $L(p, q)$ adalah $r(p, q)$ maka ekspresi regular L adalah

$$r = \sum_{q \in A} r(q_0, q)$$

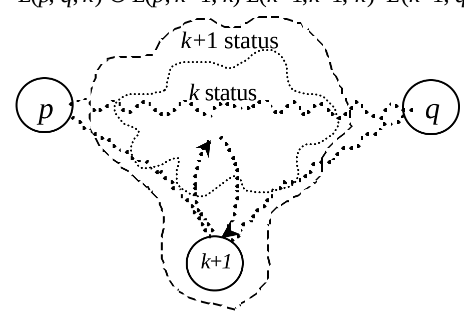
Sementara untuk menjelaskan $L(p, q)$ maka kita perlu pendefinisian $L(p, q, k)$ sebagai subset dari $L(p, q)$ yaitu string-string dimana saat transisi M bergerak melintasi tidak lebih dari k status yang berbeda (tidak termasuk p dan q). Untuk bisa diformulasikan maka status-status dalam M perlu dinomori 1, 2, …, n (n jumlah status dalam M). Maka dengan penomoran tsb $L(p, q, k)$ dapat diformulasikan sebagai berikut ini.

Khusus untuk $k = 0$,

$$L(p, q, 0) = \begin{cases} \{a \in \Sigma \mid d(p, a) = q\} & \text{dimana } p \neq q \\ \{a \in \Sigma \mid d(p, a) = p\} \cup \{\Lambda\} \end{cases}$$

Baris kedua (dimana $p = q$) menyatakan M bisa bertranslasi dengan simbol a atau tidak. Untuk $0 < k < n$., secara rekursif,

$$L(p, q, k+1) = L(p, q, k) \cup L(p, k+1, k)\, L(k+1, k+1, k)^*\, L(k+1, q, k)$$



Dari definisi tersebut maka $L(p, k) = L(p, k, n)$. karena pada saat semua status terlewati maka tidak ada lagi maka ruas kedua meliputi semula ruas pertama.

Dalam komputasinya akan lebih sederhana jika menggunakan ekspresi regularnya Jika ekspresi regular dari $L(p, q, k)$ adalah $r(p, q, k)$, maka

$r(p, q, 0) = a$ untuk $d(p, a)=q$ dan $p \neq q$,

$r(p, q, 0) = \Lambda + a$ untuk $d(p, a)=q$, serta

$r(p, q, k+1) = r(p, q, k) + r(p, k+1, k)\, r(k+1, k+1, k)^* \, r(k+1, q, k)$,

untuk $0 < k < n$.

Contoh: kita akan mendapatkan ekspresi regular dari bahasa yang dikenali FA berikut ini.

Dalam komputasinya sebenarnya dimulai dari $r = r(1, 1) + r(1, 2) = r(1, 1, 3) + r(1, 2, 3)$, dst. Namun agar penulisannya lebih rapih maka komputasi dimulai dari k=0, 1, 2. Jadi komputasi dimulai dari penghitungan $r(p,q, 0)$ (untuk semua kemungkinan p dan q)

$r(1, 1, 0) = a + \Lambda$	$r(1, 2, 0) = b$	$r(1, 3, 0) = \varnothing$
$r(2, 1, 0) = a$	$r(2, 2, 0) = \Lambda$	$r(2, 3, 0) = b$
$r(3, 1, 0) = a$	$r(3, 2, 0) = b$	$r(3, 3, 0) = \Lambda$

Untuk $k = 0$ dihitung (untuk semua kemungkinan p dan q)

$r(1, 1, 1) = r(1, 1, 0) + r(1, 1, 0)\, r(1, 1, 0)^* \, r(1, 1, 0)$

$= (r(1, 1, 0))^+ = (a + L)^+ = a^*$

$r(1, 2, 1) = r(1, 2, 0) + r(1, 1, 0)\, r(1, 1, 0)^* \, r(1, 2, 0)$

$= b + (a + \Lambda)(a + \Lambda)^* b = (a + \Lambda)^* b = a^* b$

$r(1, 3, 1) = r(1, 3, 0) + r(1, 1, 0)\, r(1, 1, 0)^* \, r(1, 3, 0)$

$= \varnothing + (a + \Lambda)(a + \Lambda)^* \varnothing = \varnothing$

$r(2, 1, 1) = r(2, 1, 0) + r(2, 1, 0)\, r(1, 1, 0)^* \, r(1, 1, 0)$

$= a + a(a + \Lambda)^* (a + \Lambda) = a(a + \Lambda)^* = a^+$

$r(2, 2, 1) = r(2, 2, 0) + r(2, 1, 0)\, r(1, 1, 0)^* \, r(1, 2, 0)$

$= \Lambda + a(a + \Lambda)^* b = \Lambda + a^+ b$

$r(2, 3, 1) = r(2, 3, 0) + r(2, 1, 0)\, r(1, 1, 0)^* \, r(1, 3, 0)$

$= b + a(a + \Lambda)^* \varnothing = \text{b} + \varnothing = b$

$r(3, 1, 1) = r(3, 1, 0) + r(3, 1, 0)\, r(1, 1, 0)^* \, r(1, 1, 0)$

$= a + a(a + \Lambda)^* (a + \Lambda) = a^+$

$r(3, 2, 1) = r(3, 2, 0) + r(3, 1, 0)\, r(1, 1, 0)^* \, r(1, 2, 0)$

$= b + a(a + \Lambda)^* b = b + a^* b = a^* b$

$r(3, 3, 1) = r(3, 3, 0) + r(3, 1, 0)\, r(1, 1, 0)^* \, r(1, 3, 0)$

$= \Lambda + a(a + \Lambda)^* \varnothing = \Lambda$

Selanjutnya untuk k=1

$r(1, 1, 2) = r(1, 1, 1) + r(1, 2, 1)\, r(2, 2, 1)^*$ rr2 9e0 Tj Tc (.75 TD /F0 9 Tf ( ) Tj 2.25 0 TD /F2 9 Tf 0.2

$$
\begin{aligned}
&= b + (a^{+}b)^{*}b = (a^{+}b)^{*}b \\
r(3, 1, 2) &= r(3, 1, 1) + r(3, 2, 1)\, r(2, 2, 1)^{*}\, r(2, 1, 1) \\
&= a^{+} + a^{*}b(\Lambda + a^{+}b)^{*}a^{+} \\
&= a^{+} + a^{*}b(a^{+}b)^{*}\, a^{+} = a^{+} + a^{*}(ba^{+})^{*}\, ba^{+} \\
&= a^{+} + a^{*}(ba^{+})^{+} \\
r(3, 2, 2) &= r(3, 2, 1) + r(3, 2, 1)\, r(2, 2, 1)^{*}\, r(2, 2, 1) \\
&= a^{*}b + a^{*}b(\Lambda + a^{+}b)^{*}(\Lambda + a^{+}b) \\
&= a^{*}b + a^{*}b(a^{+}b)^{*} = a^{*}b\,(a^{+}b)^{*} \\
r(3, 3, 2) &= r(3, 3, 1) + r(3, 2, 1)\, r(2, 2, 1)^{*}\, r(2, 3, 1) \\
&= \Lambda + a^{*}b(\Lambda + a^{+}b)^{*}b \\
&= \Lambda + a^{*}b(a^{+}b)^{*}b
\end{aligned}
$$

Untuk $k = n$ komputasi hanya dilakukan pada $r(p, q, 3)$ dengan $p = 1$ (status inisial) dan $q \in A = \{1, 2\}$.

$$
\begin{aligned}
r(1, 1, 3) &= r(1, 1, 2) + r(1, 3, 2)\, r(3, 3, 2)^{*}\, r(3, 1, 2) \\
&= a^{*}(ba^{+})^{*} + a^{*}(ba^{+})^{*}bb(\Lambda + a^{*}b(a^{+}b)^{*}b)^{*}(a^{+} + a^{*}(ba^{+})^{+}) \\
&= a^{*}(ba^{+})^{*} + a^{*}(ba^{+})^{*}bb(a^{*}(ba^{+})^{*}bb)^{*}(a^{+} + a^{*}(ba^{+})^{+}) \\
&= a^{*}(ba^{+})^{*} + (a^{*}(ba^{+})^{*}bb)^{+}(a^{+} + a^{*}(ba^{+})^{+})
\end{aligned}
$$

$$
\begin{aligned}
r(1, 2, 3) &= r(1, 2, 2) + r(1, 3, 2)\, r(3, 3, 2)^{*}\, r(3, 2, 2) \\
&= a^{*}(ba^{+})^{*}b + a^{*}(ba^{+})^{*}bb(\Lambda + a^{*}b(a^{+}b)^{*}b)^{*}a^{*}(ba^{+})^{*}b \\
&= a^{*}(ba^{+})^{*}b + a^{*}(ba^{+})^{*}bb(a^{*}(ba^{+})^{*}bb)^{*}a^{*}(ba^{+})^{*}b \\
&= a^{*}(ba^{+})^{*}b + (a^{*}(ba^{+})^{*}bb)^{*}a^{*}(ba^{+})^{*}b \\
&= (a^{*}(ba^{+})^{*}bb)^{*}a^{*}(ba^{+})^{*}b
\end{aligned}
$$

Kemudian $r = r(1, 1, 3) + r(1, 2, 3)$

$= a^{*}(ba^{+})^{*} + (a^{*}(ba^{+})^{*}bb)^{+}(a^{+} + a^{*}(ba^{+})^{+}) + (a^{*}(ba^{+})^{*}bb)^{*}a^{*}(ba^{+})^{*}b$

$= a^{*}(ba^{+})^{*} + (a^{*}(ba^{+})^{*}bb)^{*}a^{*}(ba^{+})^{*}(bba^{+} + bba^{*}(ba^{+})^{+} + b)$

Dengan mungkinnya secara teoritis mendapatkan NFA-Λ dari satu regular ekspresi dan sebaliknya mendapatkan regular ekspresi dari suatu FA maka teorema Kleene yang menyatakan "suatu bahasa L adalah regular jika dan hanya jika ada suatu FA yang dapat mengenalinya".