Jurnal Matematika Vol.6 No.1,
November 2006 [63 - 70]

EKSPRESI REGULAR PADA SUATU *DETERMINISTIC FINITE STATE AUTOMATA*

Didi Suhaedi

Jurusan Matematika, UNISBA, Jalan Tamansari No 1, Bandung,40116, Indonesia
dsuhaedi@eudoramail.com

Abstrak. Automata merupakan suatu sistem yang terdiri atas sejumlah berhingga state, dimana state menyatakan informasi mengenai input yang lalu, dan dapat juga dianggap sebagai memori mesin. Input pada mesin automata dianggap sebagai bahasa regular yang harus dikenali oleh mesin. Selanjutnya mesin automata membuat keputusan yang mengindikasikan apakah input itu diterima atau ditolak. Representasi suatu bahasa regular pada automata lebih lazim diwakili dengan menggunakan ekspresi regular.

Kata Kunci: bahasa regular; ekspresi regular; finite state automata

1. Pendahuluan

Finite state automata dan ekspresi regular awalnya dikembangkan berdasarkan pemikiran *neural network* dan *switching circuit*. *Finite state automata* merupakan *tool* yang sangat berguna dalam perancangan *lexical analyzer,* yaitu bagian dari kompilator yang mengelompokkan karakter-karakter ke dalam *token,* yang berupa unit terkecil seperti nama, variabel dan *keyword.* Dalam sistem penulisan kompilator secara otomatis akan mentransformasikan ekspresi regular kedalam *finite state automata* untuk dipakai sebagai penganalisa leksikal. *Finite state automata* dan ekspresi regular dipakai juga dalam *text editor, pattern-matching,* sejumlah pemrosesan teks, dan program *file-searching*, serta sebagai konsep matematis untuk aplikasi disiplin ilmu lain seperti logika.

Suatu bahasa pemrograman harus didefinisikan secara tepat. Spesifikasi dari sebuah bahasa pemrograman meliputi: himpunan simbol-simbol (*alphabet*) yang bisa dipakai untuk membentuk program yang benar, himpunan program yang benar secara sintaktik, dan *makna* dari program tersebut.

2. Pengantar Bahasa dan Ekspresi Regular

Sebuah *simbol* adalah suatu entitas abstrak yang tidak didefinisikan secara formal, seperti halnya tidak didefinisikannya "*titik*" dan "*garis*" pada geometri. Huruf dan digit adalah contoh dari simbol yang sering dipakai. Sebuah *string* adalah deretan berhingga dari simbol-simbol. Sebagai contoh '*a*', '*b*', '*c*' adalah simbol-simbol, dan '*abcb*' adalah suatu string. Panjang string adalah jumlah simbol yang membentuk string tersebut. Sebagai contoh string '*asdfgh*' mempunyai panjang string 6.

Sebuah string kosong dinyatakan dengan Λ, dan panjang string kosong, $|\Lambda|$, adalah 0. Suatu *alfabet,* $\sum$, adalah himpunan berhingga dari simbol-simbol. Sebuah *bahasa* adalah himpunan string-string dari simbol-simbol untuk suatu alfabet. Suatu bahasa yang tidak terdiri dari string-string dikenal dengan bahasa kosong, dan disimbolkan dengan $\varnothing$.

Misalkan $\sum$ adalah sembarang alfabet, himpunan semua string atas $\sum$ dinotasikan dengan $\sum^*$, dan suatu bahasa atas $\sum$ adalah himpunan bagian dari $\sum^*$. Misalkan $\sum = \{a, b\}$ maka : $\{a, b\}^* = \{\Lambda, a, b, aa, ab, ba, bb, aaa, aab, aba, abb, baa, \ldots\}$.

Definisi 1.
Misalkan $\sum$ merupakan suatu alfabet. Koleksi dari bahasa-bahasa regular atas $\sum$ didefinisikan secara rekursif sebagai berikut :

a. $\varnothing$ adalah sebuah bahasa regular
b. $\{\Lambda\}$ adalah sebuah bahasa regular
c. Untuk setiap $a \in \sum$, $\{a\}$ adalah sebuah bahasa regular
d. Jika A dan B adalah bahasa-bahasa regular, maka $A \cup B$, $A.B$, A^* adalah bahasa-bahasa regular.
e. Tidak ada bahasa-bahasa lain atas $\sum$ yang regular.

Contoh 1.
Misalkan $\sum = \{a, b\}$ maka yang berikut adalah benar.

- $\varnothing$ dan $\{\Lambda\}$ adalah bahasa-bahasa regular
- $\{a\}$ dan $\{b\}$ adalah bahasa-bahasa regular
- $\{a,b\}$ adalah bahasa regular yang merupakan gabungan dari $\{a\}$ dan $\{b\}$
- $\{ab\}$ adalah bahasa regular
- $\{a, ab, b\}$ adalah bahasa regular
- $\{a_i \mid i \geq 0\}$ adalah bahasa regular
- $\{a_i\, a_j \mid i \geq 0 \text{ dan } j \geq 0\}$ adalah bahasa regular
- $\{(ab)^i \mid i \geq 0\}$ adalah bahasa regular

Karena suatu bahasa adalah himpunan dari string-string, maka bahasa baru dapat dikonstruksi dengan menggunakan operasi himpunan. Misalkan terdapat dua bahasa atas alfabet $\sum$, maka *union*, *intersection*, dan *difference* adalah merupakan bahasa atas $\sum$. Komplemen bahasa
atas $\sum$ (dengan menetapkan himpunan universal bahasa adalah $\sum^*$) adalah $L' = \sum^* - L$.

Untuk mengkonstruksi bahasa, juga dapat dilakukan melalui operasi *concatenation* atas string-string yang telah diketahui. Jika x dan y adalah elemen dari $\sum^*$, *concatenation* dari x dan y adalah string xy dengan menuliskan simbol-simbol dari x dan y secara berturut-turut. Jika $x = abb$ dan $y = ba$, maka $xy = abbba$ dan $yx = baabb$. Untuk sembarang string x, $x\Lambda = \Lambda x = x$. Jelas bahwa *concatenation* bersifat asosiatif.

Suatu string x merupakan *sub*-string dari y jika terdapat string w dan z (salah satu atau keduanya boleh merupakan string kosong), maka $y = wxz$. String *car* merupakan *sub*-string dari *descartes*, *vicar*, *carthage*, dan *car*, tetapi bukan merupakan *sub*-string dari *charity*. Jika $L_1, L_2 \subseteq \sum^*$ maka $L_1 L_2 = \{ xy \mid x \in L_1 \text{ dan } y \in L_2 \}$.
Sebagai contoh : {*hope, fear* } {*less, fully* } = {*hopeless, hopefully, fearless, fearfully* }.

Notasi eksponensial digunakan untuk mengindikasikan berulangnya jumlah *concatenation.* Notasi tersebut dapat digunakan untuk simbol, string, atau bahasa. Jika $\sum$ adalah alfabet dan jika $a \in \sum$, $x \in \sum^*$, dan $L \subseteq \sum^*$ maka :

$$a^k = aa \ldots a$$
$$x^k = xx \ldots x$$
$$\sum^k = \sum\sum \ldots \sum$$
$$L^k = LL \ldots L$$

Untuk suatu kondisi khusus k = 0

$$a^0 = \Lambda \qquad x^0 = \Lambda$$
$$\sum^0 = \{\Lambda\} \qquad L^0 = \{\Lambda\}$$

Suatu bahasa regular dapat disederhanakan representasinya dengan menggunakan *ekspresi regular*. Secara umum penyederhanaan bahasa regular menjadi ekspresi regular, dengan menghilangkan tanda { } dan mengganti tanda $\cup$ dengan + untuk ekspresi regular.

Misalkan $\sum$ = {0, 1}, berikut adalah contoh korespondensi antara bahasa regular dan ekspresi regular.

No.	Bahasa Regular	Ekspresi Regular
1.	{Λ}	Λ
2.	{0}	0
3.	{001}	001
4.	{0, 1}	0 + 1
5.	{0, 10}	0 + 10
6.	{1, Λ} {001}	(1 + Λ) 001
7.	{110}* {0, 1}	(110)*(0 + 1)
8.	{1}* {10}	1*10
9.	{10 , 111 , 11010}*	(10 + 111 + 11010)*
10.	{0 , 10}* ({11}* ∪ { 001 , Λ})	(0 + 10)*((11)* + 001 + Λ)

Definisi 2.
Kelas bahasa regular R atas $\sum$ berkorespondensi dengan ekspresi regular (r) dengan mengikuti aturan yang didefinisikan sebagai berikut :

1. $\varnothing \in R$ berkorespondensi dengan ekspresi regular $r = \varnothing$
2. $\{\Lambda\} \in R$ berkorespondensi dengan ekspresi regular $r = \Lambda$
3. Setiap $a \in \sum$, $\{a\} \in R$ berkorespondensi dengan ekspresi regular $r = a$
4. $L_1 , L_2 \in R$ dengan ekspresi regular masing-masing r_1 dan r_2 maka
   a. $L_1 \cup L_2 \in R$ berkorespondensi dengan ekspresi regular $r = r_1 + r_2$
   b. $L_1 L_2 \in R$ berkorespondensi dengan ekspresi regular $r = r_1 r_2$
   c. $L_1^* \in R$ berkorespondensi dengan ekspresi regular $r = r_1^*$

Suatu ekspresi regular r^2 sering dinyatakan dengan rr , dan r^+ untuk menyatakan ekspresi regular $r^* r$. Sebagai ilustrasi konversi dari bahasa regular ke ekspresi regular, misalkan $\sum$ = {0, 1} untuk suatu bahasa L = {00, 01, 10, 11}* mempunyai ekspresi regular untuk L adalah r = (00 + 01 + 10 + 11)* atau ekspresi regular yang lebih sederhananya adalah r = ((0 + 1) (0 + 1))*.

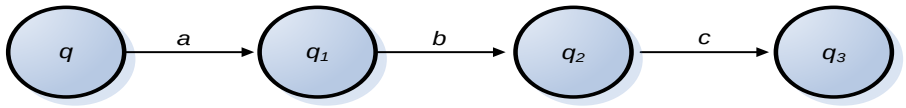
3. Ekspresi Regular pada suatu *Finite State Automata*

Definisi 3.
Suatu *finite state automata* (*FSA*) adalah suatu 5-tuple ($Q, \sum, q_o , \delta , A$) dengan Q adalah *finite set*, $\sum$ adalah *finite alphabet* untuk input simbol, $q_o \in Q$ adalah *state* awal, δ adalah fungsi transisi dari $Q \times \sum \rightarrow Q$, dan $A \subseteq Q$ adalah himpunan target *state*.
Untuk sembarang $q \in Q$ dan $a \in \sum$, $\delta(q , a)$ sebagai fungsi transisi yang akan memindahkan *state* q ke *state* lain dengan input a.

Definisi 4.
Misalkan $M = (Q, \sum , q_o , \delta , A)$ adalah *FSA*, fungsi $\delta^* : Q \times \sum^* \rightarrow Q$ didefinisikan dengan

i. sembarang $q \in Q$, $\delta^*(q , \Lambda) = q$
ii. sembarang $y \in \sum^*$, $a \in \sum$, dan $q \in Q$, $\delta^*(q , ya) \rightarrow \delta(\delta^*(q , y) , a)$

Contoh 2.
Misalkan $M = (Q, \sum , q_o , \delta , A)$ adalah suatu *FSA* dengan diagram transisi :



dengan menggunakan definisi di atas, tentukan penyelesaian dari $\delta^*(q , abc)$!
Penyelesaian :

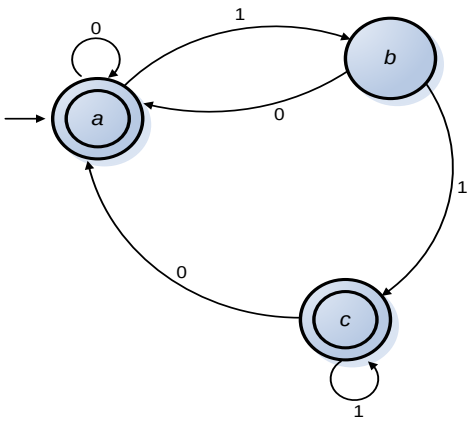
$$\begin{aligned}
\delta^*(q , abc) &= \delta(\delta^*(q , ab), c) \\
&= \delta(\delta(\delta^*(q , a) , b), c) \\
&= \delta(\delta(\delta^*(q , \Lambda a) , b), c) \\
&= \delta(\delta(\delta(\delta^*(q , \Lambda) , a) , b), c) \\
&= \delta(\delta(\delta(q , a) , b), c) \\
&= \delta(\delta(q_1 , b), c) \\
&= \delta(q_2 , c) \\
&= q_3
\end{aligned}$$

Definisi 5.
Misalkan $M = (Q, \sum , q_o , \delta , A)$ adalah *FSA.* Suatu string $x \in \sum^*$ diterima oleh M jika $\delta^*(q_o , x) \in A$. Bahasa yang diterima oleh M adalah himpunan

$$L(M) = \{ x \in \sum^* \mid x \text{ diterima oleh } M \}$$

Jika L adalah sembarang bahasa atas $\sum$, maka L dikatakan diterima oleh M jika dan hanya jika $L = L(M)$.

Contoh 3.
Diberikan *FSA* yang dideskripsikan dengan diagram berikut :



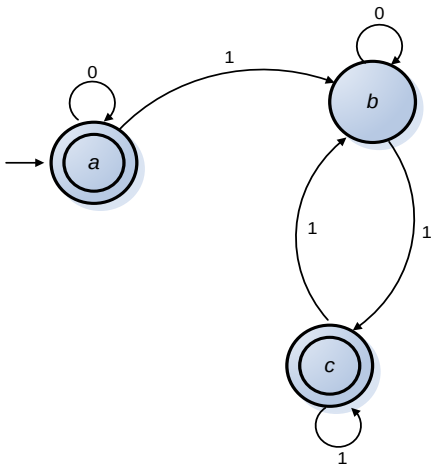
Tunjukkan bahwa string 23 diterima sebagai bahasa oleh *M.*

Penyelesaian :
Dari diagram tersebut diketahui bahwa $Q = \{a , b , c \}$, $\sum = \{0 , 1\}$, $q_o = a$, $A = \{a , c \}$.
Bilangan desimal 23 terlebih dulu dikonversi ke biner menjadi, 23 = 10111.

$\delta^*(a, 10111) = \delta(\delta^*(a, 1011),1)$
$= \delta(\delta(\delta^*(a, 101),1),1)$
$= \delta(\delta(\delta(\delta^*(a, 10),1),1),1)$
$= \delta(\delta(\delta(\delta(\delta^*(a, 1),0),1),1),1)$
$= \delta(\delta(\delta(\delta(\delta^*(a, \Lambda 1),0),1),1),1)$
$= \delta(\delta(\delta(\delta(\delta(\delta^*(a, \Lambda),1),0),1),1),1)$
$= \delta(\delta(\delta(\delta(\delta(a,1),0),1),1),1)$
$= \delta(\delta(\delta(\delta(b,0),1),1),1)$
$= \delta(\delta(\delta(a,1),1),1)$
$= \delta(\delta(b,1),1)$
$= \delta(c,1)$
$= c$

dari uraian tersebut didapat $\delta^*(a, 10111) = c \in A$, dengan demikian string 23 diterima oleh *FSA.*

Ekspresi regular sebagai suatu pola untuk string merupakan pembangun (generator) dari bahasa. Berikut diberikan contoh permasalahannya. Misalkan diberikan *FSA* sebagai berikut :



Tentukan ekspresi regularnya.

Penyelesaian.

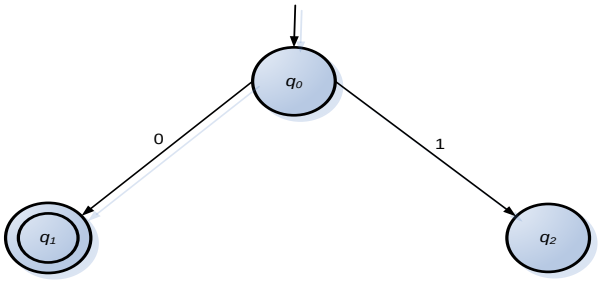
- $\Lambda \in L$ karena $\delta^*(a, \Lambda) = a \in A$ sehingga Λ diterima oleh L.
- $0^* \in L$ karena $\delta^*(a, 0^*) = a \in A$ sehingga 0^* diterima oleh L.
- $(10^*1(01)^*1^*)^* \subseteq L$

  $\delta^*(a, 10^*1(01)^*1^*10^*1(01)^*1^*)$
  $= \delta^*(c, 10^*1(01)^*1^*)$
  $= \delta^*(c, 0^*1(01)^*1^*)$; terdapat dua kemungkinan $0^* \neq \Lambda$ dan $0^* = \Lambda$
  Untuk $0^* \neq \Lambda$
  $\delta^*(c, 0^*1(01)^*1^*)$
  $= \delta^*(b, 1(01)^*1^*)$
  $= \delta^*(c, (01)^*1^*)$
  $= c$
  Untuk $0^* = \Lambda$
  $\delta^*(c, 0^*1(01)^*1^*)$
  $= \delta^*(c, 1(01)^*1^*)$
  $= \delta^*(c, (01)^*1^*)$
  $= c$

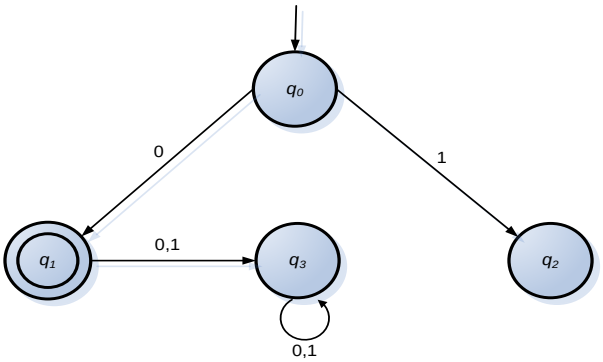
Dari uraian tersebut, dihasilkan bahasa $L = \{ 0^* \cup (10^*1(01)^*1^* \}$ dengan ekspresi regularnya adalah $r = 0^* + (10^*1(01)^*1^*$

Contoh 4.
Diketahui $r = (11 + 110)^*0$, konstruksi D*FSA* dengan menggunakan ekspresi regular tersebut. Misalkan state awal adalah q_o

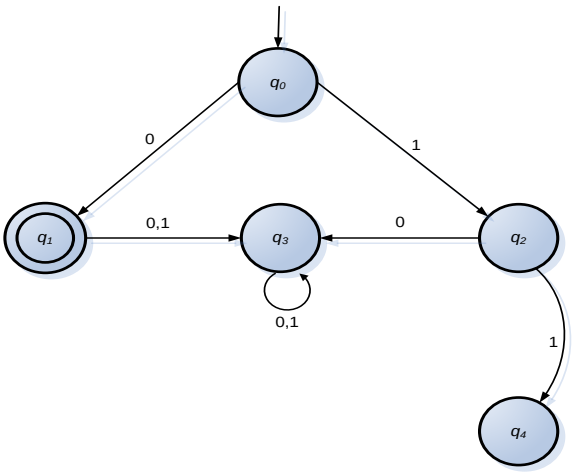
- $\Lambda \notin L$ karena string minimal anggota L adalah 0 sehingga state awal $q_o \notin A$
- $0 \in L$ sehingga $\delta^*(q_o, 0) = q_1 \in A$
- $1 \notin L$ sehingga $\delta^*(q_o, 1) = q_2 \notin A$



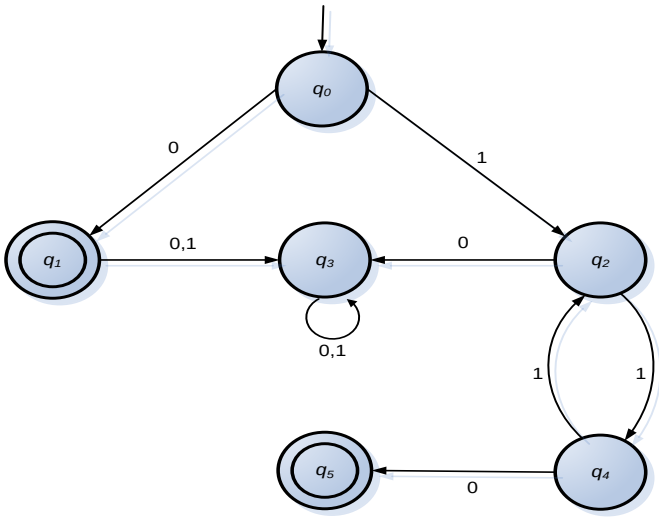
- $00 \notin L$ dan tidak ada string yang dimulai dengan 00 anggota L , sehingga $\delta^*(q_1, 0) = q_3 \notin A$
- $01 \notin L$ dan tidak ada string yang dimulai dengan 01 anggota L , sehingga $\delta^*(q_1, 1) = q_3 \notin A$



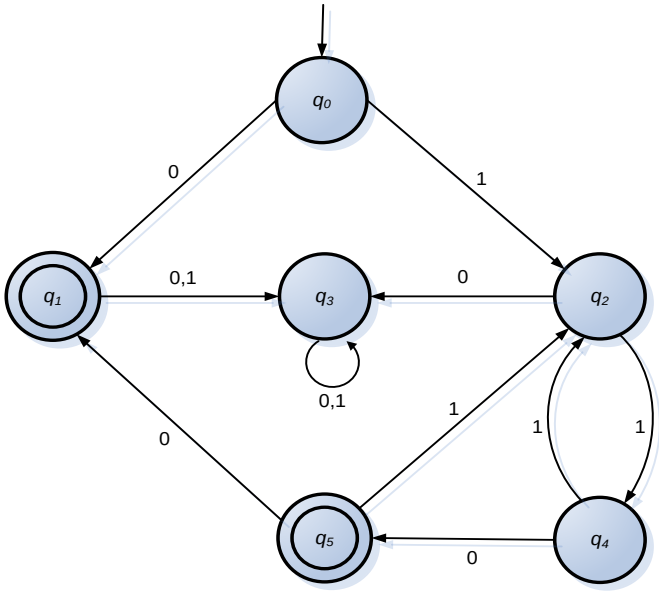
- $10 \notin L$ dan tidak ada string yang dimulai dengan 10 anggota L , sehingga $\delta^*(q_2, 0) = q_3 \notin A$
- Misal $110 \in L$ atau $1100 \in L$ hal ini berarti ada state baru q_4 dan $\delta^*(q_2, 1) = q_4 \in L$



- $110 \in L$ maka $\delta^*(q_4, 0) = q_5 \in A$
- $111 \notin L$ sehingga jika q_2 diberikan input 1 akan berpindah ke state yang bukan merupakan target state yang paling realistis masuk ke q_2 sehingga $\delta^*(q_4, 1) = q_2$



- $1100 \in L$ state yang merupakan target state adalah q_1 maka $\delta^*(q_5, 0) = q_1 \in A$
- $11010 \notin L$, $110110 \in L$ state yang paling mungkin adalah q_2 sehingga $\delta^*(q_5, 1) = q_2$



4. Penutup

Untuk setiap *Deterministic Finite State Automata* (DFSA) ada suatu ekspresi regular dari bahasa yang diterima oleh *DFSA*. Jika diketahui diagram dari suatu *DFSA* maka dapat ditentukan ekspresi regular untuk *DFSA* tersebut, dan sebaliknya jika diketahui suatu ekspresi regular maka dapat dikonstruksi diagram *DFSA*-nya.

Pustaka

[1] Dea Kelly, *Automata and Formal Language : An Introduction*, 1995, Prentice – Hall, Inc., Englewood Clift, New Jersey.

[2] Doerr A. dan Levasseur K., *Applied Discrete Structures for Computer Science*, 1989, Science Research Associates, Inc., Toronto.

[3] Lipschutz, Seymour (1992), Discrete Mathematics, McGrawHill, New York.

[4] Liu, C. L., *Element of Discrete Mathematics*, 1985, McGraw-Hill, Inc., Toronto.

[5] Rosen, K.H., (2003). Discrete Mathematics and Its Applications, Fifth Ed., McGrawHill, Singapore.

[6] Utdirartatmo F., *Teori Bahasa dan Otomata*, 2001, J&J Learning, Yogyakarta.