

AUTOMATA dan BAHASA FORMAL

“Praktika Finite Automata Dengan Output”

Aris Eka Subiyanto
Zira_ilkom@yahoo.com

Lisensi Dokumen:

Copyright © 2009 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

1. PENDAHULUAN

Bisa dikatakan bahwa input string merepresentasikan suatu program dan input data. Proses reading / pembacaan abjad-abjad dari suatu string analog dengan pengekseskuan instruksi-instruksi yang mengisyaratkan perpindahan state pada mesin tersebut. Secara teoritis seharusnya kita mampu menghasilkan output dengan panjang yang terbatas, tetapi bagaimana jika misalnya secara sederhana kita menginginkan pencetakan input string yang bisa jadi hasilnya cukup panjang ?

Pertanyaan inilah yang harus dihadapi sehubungan dengan Model matematika Finite Automata dan Graph transisi yang merepresentasikan Mesin secara fisik. Keterbatasan Finite Automata ditunjukkan dalam keputusannya hanya pada string-string yang diterima atau ditolak, Automata tersebut biasa disebut Language Acceptor atau Recognizer. Merekonstruksi Model matematika suatu Finite State Automata menjadi FSA yang menghasilkan suatu output bukanlah hal yang mustahil, hal ini sudah bisa

dipecahkan dengan ditemukannya model tersebut oleh G.H Meally (1955) dan secara terpisah oleh E.F Moore (1956). Tujuan inti dari penemuan ini adalah untuk mendesain suatu model Matematika bagi penerapan sirkuit sekuensial, yang merupakan abstraksi komponen kerja komputer secara fisik.

Diberikan penjabaran tentang Mesin Mealy dan Mesin Moore secara terpisah sesuai dengan karakteristiknya masing-masing, baik di dalam pendefinisiannya, prinsip kerjanya dan representasi Graphnya. Dengan teori yang ada akan diimplementasikan dalam beberapa contoh penerapan praktis yang cukup familiar sehingga akan semakin jelas pengertian dan nilai kegunaan Mesin-mesin tersebut, baik sekarang maupun nilai prediktif dimasa datang.

2. DEFINISI dan IMPLEMENTASI

Pada bagian ini akan diberikan definisi Mesin Mealy dan Mesin Moore secara terpisah kemudian contoh implementasi pada Sirkuit

2.1 MESIN MOORE

2.1.1 Pendefinisian

Definisi Mesin Moore :

Mesin Moore didefinisikan dalam 6(enam) Tupple, $M_o = (Q, \Sigma, \delta, S, \Delta, \lambda)$,
dimana :

Q = Himpunan state terbatas / letter

Σ = Himpunan symbol input

δ = Fungsi Transisi

S = State awal, $S \in Q$

Δ = Himpunan symbol output / karakter

λ = Fungsi output untuk setiap state

Pada mesin Moore, diawal proses selalu mencetak /menghasilkan karakter yang berada pada start state. Mesin Moore tidak mendefinisikan suatu bahasa dalam menerima untai-untai, karena setiap input string menghasilkan output string dan tidak terdapat final state. Proses akan berhenti jika input letter/abjad terakhir dibaca dan output karakter terakhir dicetak. Jikalau input string misalnya n abjad, maka output string akan

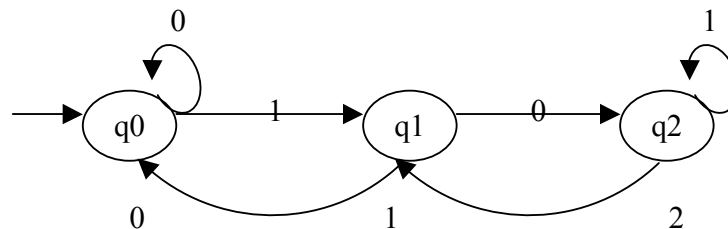
terdapat $n+1$ karakter karena terdapat $n+1$ state yang dilalui dalam prosesnya / path.

2.1.2 Representasi Graf

Misal kita ingin memperoleh sisa pembagian (modulus) suatu bilangan dengan 3. Dengan input yang dinyatakan dalam biner.

Maka :

$Q = (q_0, q_1, q_2)$, $\Sigma = \{0, 1\}$, $\Delta = \{0, 1, 2\}$, $S = q_0$, $\lambda(q_0) = 0$, $\lambda(q_1) = 1$, $\lambda(q_2) = 2$



Dari gambar diatas :

Misal : $5 \bmod 3 = ?$

Input : 101 maka urutan state yang dicapai : q_0, q_1, q_2, q_2

State terakhir yang dicapai adalah q_2 , $\lambda(q_2) = 3$

2.2 MESIN MEALY

2.2.1 Pendefinisian

Definisi Mesin Moore :

Mesin Mealy didefinisikan 6(enam) tuple, $Me = (Q, \Sigma, \delta, S, \Delta, \lambda)$, dimana :

Q = Himpunan state terbatas / letter

Σ = Himpunan symbol input

δ = Fungsi Transisi

S = State awal, $S \in Q$

Δ = Himpunan symbol output / karakter

λ = Fungsi output untuk setiap transisi

Setiap Edge yang terbentuk dilabel komposisi symbol i/o (input/output).

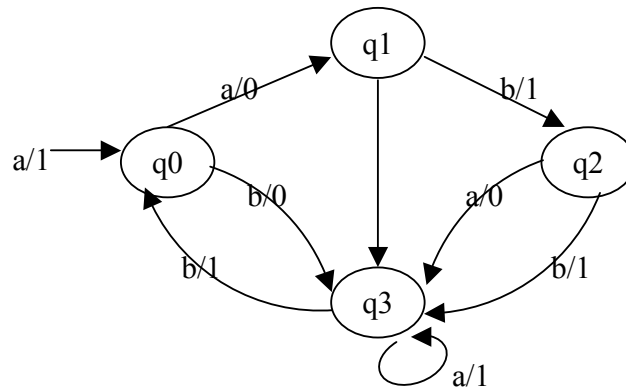
Setiap state harus mempunyai satu buah edge yang mengarah ke state

berikutnya untuk setiap kemungkinan input letter. Edge yang dilalui

tergantung kepada input letter i, ketika menjelajahi edge kita harus mencetak output karakter o. Mesin Moore tidak mendefinisikan suatu bahasa dalam menerima untai-untai, karena setiap input string menghasilkan output string dan tidak terdapat final state. . Jikalau input string misalnya n abjad, maka output string akan terdapat n karakter.

2.1.2 Representasi Graf

Misal

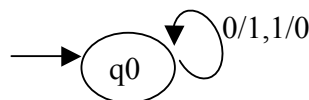


input : aaabb
 State yang ditempuh : q1-q3-q3-q0-q3
 Output : 01110

2.3 Implementasi Pada Sirkuit Sekuensial

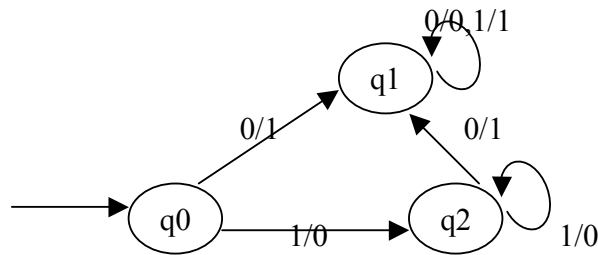
Aplikasi Finite Automata dengan output pada sekuensial sirkuit berkaitan erat dengan Arsitektur computer. Penerapan Model Mesin ini umumnya berfungsi sebagai penjelas dari aksi sekuensial sirkuit yang dimodelkan.

Teladan : 1> Mesin penghasil 1's komplemen yang bisa dievolusikan menjadi mesin pengurangan



input : 001010
 output : 110101

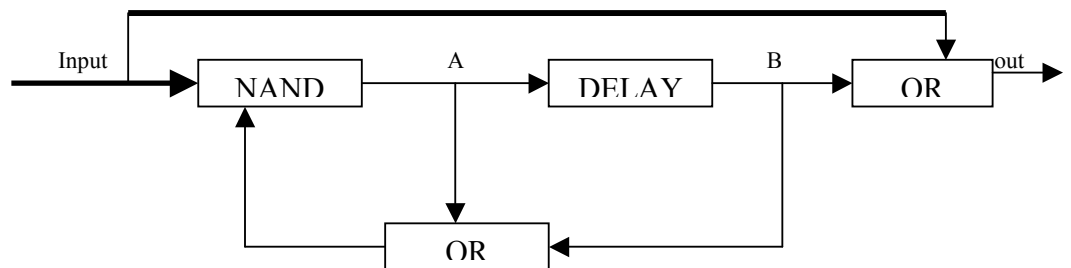
2> Incrementer yang biasa kita temukan pada CPU



input : 1011 -> dibaca terbalik 1101
 output : 0011 -> ditulis terbalik 1100

3> Flip-flop merupakan suatu rangkaian gerbang logika yang mempunyai dua keadaan stabil pada outputnya, dalam hal ini direpresentasikan dengan bit 0,1. Terdapat beberapa jenis Flip-flop yaitu : SR Flip-Flop, JK Flip-flop, Master-Slave JKFF, dan D Flip-flop. Flip-flop berfungsi menyimpan data.

- D Flip Flop,



Diketahui terdapat 4(empat) state berdasarkan ada tidaknya arus pada titik A dan B di sirkuit

Q0 adalah A=0 B=0

Q1 adalah A=0 B=1

Q2 adalah A=1 B=0

Q3 adalah A=1 B=1

Rule perubahan state setelah input 0 atau 1

New B = Old A

$$\text{New A} = (\text{input}) \text{ NAND } (\text{Old A or Old B})$$

$$\text{Output} = (\text{input}) \text{ OR } (\text{old B})$$

Pada pulsa diskrit yang bervariasi pada penerimaan input Time clock, state berubah dan outputpun dihasilkan.

Misalkan kita berada pada state q0 dan menerima input 0 :

$$\text{New B} = \text{Old A} = 0$$

$$\text{New A} = (\text{input}) \text{ NAND } (\text{Old A or Old B})$$

$$= (0) \text{ NAND } (0 \text{ OR } 0)$$

$$= 0 \text{ NAND } 0$$

$$= 1$$

$$\text{Output} = 0 \text{ or } 0 = 0$$

State baru adalah q2 (karena New A=1, New B=0).

Jika kita berada pada q0 dan menerima input 1 :

$$\text{New B} = \text{Old A} = 0$$

$$\text{New A} = 1 \text{ NAND } (0 \text{ OR } 0) = 1$$

$$\text{Output} = 1 \text{ OR } 0 = 1$$

State baru adalah q2 (karena New A=1, New B=0).

Jika kita di q1 dan menerima input 0 :

$$\text{New B} = \text{Old A} = 0$$

$$\text{New A} = 0 \text{ NAND } (0 \text{ OR } 1) = 1$$

$$\text{Output} = 0 \text{ OR } 1 = 1$$

State baru adalah q2

Jika kita berada pada q1 dan menerima input 1 :

$$\text{New B} = \text{Old A} = 0$$

$$\text{New A} = 1 \text{ NAND } (0 \text{ OR } 0) = 1$$

$$\text{Output} = 1 \text{ OR } 1 = 1$$

State baru adalah q0

Jika kita berada pada q2 dan menerima input 0 :

$$\text{New B} = \text{Old A} = 1$$

$$\text{New A} = 0 \text{ NAND } (1 \text{ OR } 0) = 1$$

$$\text{Output} = 0 \text{ OR } 0 = 0$$

State baru adalah q3 (karena New A=1, New B=1).

Jika kita berada pada q2 dan menerima input 1 :

$$\text{New B} = \text{Old A} = 1$$

$$\text{New A} = 1 \text{ NAND } (1 \text{ OR } 0) = 0$$

$$\text{Output} = 1 \text{ OR } 0 = 1$$

State baru adalah q1

Jika kita berada pada q3 dan menerima input 0 :

$$\text{New B} = \text{Old A} = 0$$

$$\text{New A} = 0 \text{ NAND } (1 \text{ OR } 1) = 1$$

$$\text{Output} = 0 \text{ OR } 1 = 1$$

State baru adalah q3

Jika kita berada pada q3 dan menerima input 1 :

$$\text{New B} = \text{Old A} = 1$$

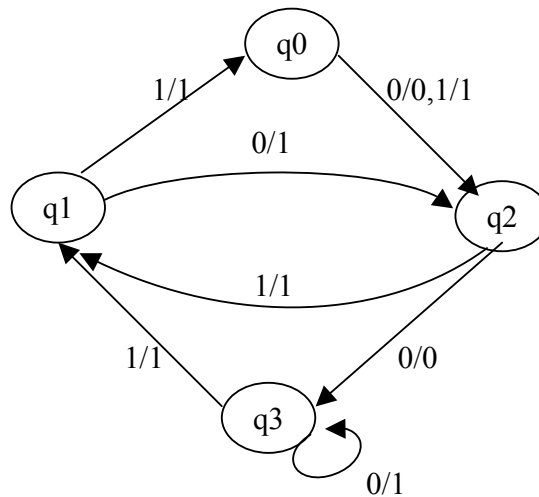
$$\text{New A} = 1 \text{ NAND } (1 \text{ OR } 1) = 0$$

$$\text{Output} = 1 \text{ OR } 1 = 1$$

State baru adalah q1.

Old state	Setelah input 0		Setelah input 1	
	New state	Output	New state	Output
q0	q2	0	q2	1
q1	q2	1	q0	1
q2	q3	0	q1	1
q3	q3	1	q1	1

Representasi aksi dari sirkuit tersebut dalam Mesin Mealy



3. PENUTUP

Finite State Automata dengan output merupakan Evolusi dari Finite State Automata konvensional yang mempunyai kemampuan lebih yaitu menghasilkan output. Mesin Mealy dan Mesin Moore merupakan contoh konkret dari FSA dengan output. Tujuan inti dari penemu Mesin ini adalah untuk mendisain suatu model matematika untuk sirkuit sekuensial.

Output pada mesin Mealy berasosiasi dengan Rule transisi sementara pada Mesin Moore Output berasosiasi dengan state. Pada Mesin Mealy maupun Moore tidak terdapat Final state sehingga tidak bisa mendefinisikan suatu bahasa dari untai-untai masukan.

4. REFERENSI

Cohen, Daniel I. A, Introduction to Computer Theory, John Wiley & Sons, Inc. Singapura : Singapura, 1991.

Kelley, Dean. Otomata Dan Bahasa Formal, PT.Prenhalindo, Jakarta : Indonesia

Wang, Randy. <http://www.cs.Princeton.edu/courses/C126>

BIOGRAFI PENULIS



Aris Eka Subiyanto.

Lahir di Kota dingin Malang, 11 Januari 1984. Menempuh Pendidikan Dasar hingga menengah di Tumpang – Malang. Sempat mengenyam pendidikan ala Teknik – Teknik sipil, menikmati indahny Sastra Inggris – ala Humaniora dan pelabuhan terakhir tersintesakan dalam Ilmu Komputer - ala Science Universitas Brawijaya Malang. Bersyukur bisa menikmati Indahny Ilmu Komputer beserta keseluruhan kompleksitasnya kendatipun masih ada banyak dan terlalu banyak hal lagi yang masih perlu diselami dari disiplin baru “mengagumkan” ini.

Aktif mengajar di LBB, asistensi praktikum keMIPAan dan sangat tertarik dengan dunia pendidikan khususnya pendidikan Matematika dan Ilmu Komputer. Rindu mendalami aspek *Philosophy Computer Science* karena memimpikan Indonesia yang tidak hanya menjadi negara pengekspor beras, gula, jagung saja tapi ekspor theory suatu saat nanti. Pendengar Lagu sweet Jazz,

Gending Jawa, Chorale beserta main musiknya....

*Ngelmu iku,
kalakone kanthi laku,
lekasane lawan kas,
tegese kas nyantosani,
Setya budaya pangekese dur angkara. (Serat Wedhatama)*

Alamat Kontak :
Zira_ilkom@yahoo.com