Copyright © 2003 IlmuKomputer.Com

AUTOMATA dan BAHASA FORMAL "Teorema KLEENE"

Aris Eka Subiyanto
Zira_ilkom@yahoo.com

Lisensi Dokumen:
Copyright © 2003 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

Formal language menunjuk kepada suatu ketentuan kebahasaan khusus yang memiliki karakteristik formal yaitu segala rule-rule yang mengatur bahasa tersebut dinyatakan secara eksplisit dalam suatu terms (dalil) yang akan menetukan output berupa string-string dari symbol. Bahasa yang ditampilkan hanya sebagai bentukan symbol-simbol dan bukan merupakan ekspresi dari penyampaian ide manusia. Terminology " formal" menekankan pada bentuk (form) dari output string dengan rule-rule pembentukan formal dan bukan pada pemaknaan kata (word).

Suatu alphabet atau abjad adalah himpunan terbatas tak kosong dari symbol-simbol dilambangkan Σ. Barisan berhingga dari symbol-simbol dari suatu abjad kadang-kadang dinamakan sebuah kata (word) yang terbentuk dari abjad itu dilambangkan w, sedangkan suatu string yang tidak terdapat abjad didalamnya disebut empty string dilambangkan Λ. Kumpulan dari string-string yang dibentuk dari alphabet disebut Bahasa dilambangkan L. Terdapat suatu bahasa yang tidak terdiri dari untai – untai –

Copyright © 2003 IlmuKomputer.Com

bahasa kosong (empty language) dilambangkan ϕ. Terdapat operasi-operasi yang terjadi pada untai dan untai diantaranya : Concatenasi (Perangkaian) , Reversal (Pembalikan), Eksponensiasi.

Suatu Bahasa dapat kita definisikan dengan berbagai cara, dalam pembahasan kali ini menggunakan tiga cara pendefinisisan suatu bahasa yaitu : Dengan menggunakan Ekspresi Reguler, menggunakan Finite Automata, dan menggunakan Graph transisi. **" Jika suatu bahasa dapat didefinisikan oleh salah satu cara pendefinisisan makan akan juga dapat didefinisikan kedua cara yang lainnya "(Teorema Kleene 1956)**. Secara singkat bisa dikatakan bahwa ketiga metode pendefinisian diatas adalah Equivalent.

2. DESKRIPSI DAN PEMBUKTIAN

Pada bagian ini akan ditunjukkan secara Konstruktif kesamaan antara Ekspresi Reguler, Finite Automata, dan Graf Transisi didalam mendefinisikan – atau merepresentasikan – bahasa formal, didahului dengan Teorema Kleene dan skema pembuktiannya.

2.1 Skema Pembuktian

Teorema Kleene :

Suatu bahasa yang dapat didefinisikan oleh :

1. Ekspresi Reguler

Atau 2. Finite Automata

Atau 3. Graph Transisi

Dapat didefinisikan oleh ketiga metode sekaligus

Proof :

Ketiga bagian pembuktiannya adalah :

Bagian 1 Setiap bahasa yang dapat didefinisikan Finite Automata dapat didefinisikan oleh Graph transisi

Bagian 2 Setiap bahasa yang dapat didefinisikan oleh Graph transisi dapat didefinisikan oleh Ekspresi reguler

Copyright © 2003 IlmuKomputer.Com

Bagian 3 Setiap bahasa yang dapat didefinisikan oleh Ekspresi regular dapat didefinisikan dengan Finite Automata

Secara sederhana bisa diilustrasikan kita ingin membuktikan bahwa himpunan TIC, TAC, TOE adalah sama, maka dicari 1. TIC = TAC, 2. TAC = TOE, 3. TIC = TOE (TIC ⊂ TAC ⊂ TOE ⊂ TIC) ≡ (TIC = TAC = TOE)

2.2 Pembuktian

2.2.1 Pembuktian Bagian 1

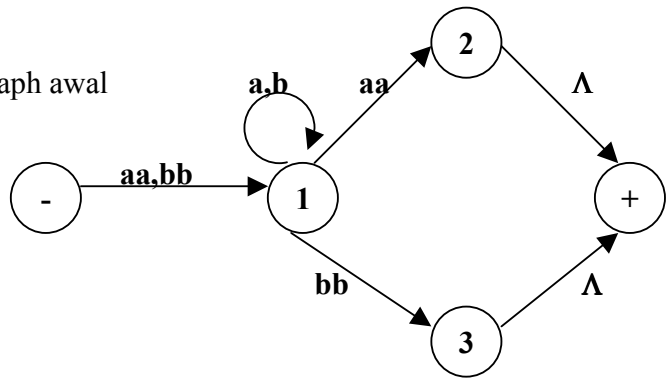
Merupakan pembuktian paling sederhana. Setiap Finite Automata adalah merupakan Graph transisi itu sendiri. Maka, Bahasa yang dapat didefinisikan oleh finite Automata juga terdefinisi dengan Transisi graph secara simultan.

2.2.2 Pembuktian Bagian 2

Secara sederhana metode pembuktian dalam tahap ini akan mengetahui Equivalensi yang terjadi secara bersamaan ketika melakukan prosedur simplifikasi (penyederhanaan suatu Graph transisi), yaitu dengan mengubah label pada edge dengan suatu Ekspresi Reguler yang relevan tanpa mengubah hasil akhir outputan sampai kita temukan pola yang lebih sederhana (Biasanya dengan jumlah Edge yang berkurang karena pemadatan Simbol-simbol pada Edge – edge Graph transisi menjadi Ekspresi reguler yang relevan).

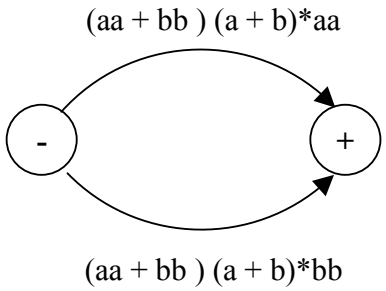
Teladan :

Graph awal



Copyright © 2003 IlmuKomputer.Com

Graph Akhir (setelah simplifikasi)



Hasil akhir berupa Ekpresi Reguler yang menghasilkan output yang sama dengan Graph transisi :

(aa+bb)(a+b)*(aa) + (aa+bb)(a+b)*(bb) = (aa+bb)(a+b)*(aa+bb)

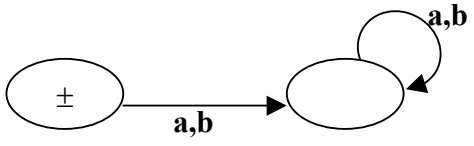
2.2.3 Pembuktian Bagian 3

Pada pembuktian bagian ketiga ini menggunakan recursive algoritma dan Constructive Algoritma secara bersamaan. Kita tahu bahwa suatu Ekspresi reguler dibangun oleh symbol-simbol dalam suatu Alphabet dan Λ oleh aplikasi aturan (rule) yang diulang : addition, concatenation, dan clousure. Terdapat 4 rule yang akan dibahas yaitu :

2.2.3.1 Rule 1

Terdapat Suatu Finite Automata (FA)yang menerima hanya abjad khusus dari suatu alpabet Σ. Terdapat suatu FA yang hanya menerima untai Λ.

Teladan. 1. FA yang hanya menerima input Λ



2.2.3.2 Rule 2

Jika terdapart suatu FA yaitu FA1 yang menerima bahasa yang didefinisikan oleh suatu Ekspresi reguler r1

Copyright © 2003 IlmuKomputer.Com

dan terdapat suatu FA yaitu FA2 yang menerima bahasa didefinisikan oleh ER r2, maka terdapat FA tertentu yaitu FA3 yang menerima suatu bahasa yang didefinisikan oleh ER (r1 + r2).

Konstruksi mesin FA3 yang menerima bahasa yang didefinisikan oleh ER (r1+r2) adalah:

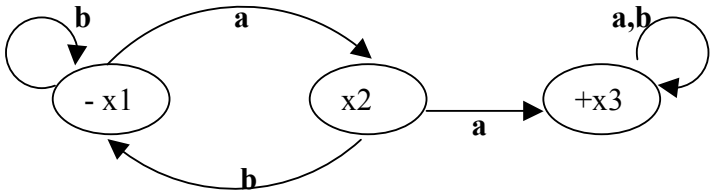
Algoritma mesin FA3 ->FA1dengan state x1, x2, x3… dan FA2 dengan state y1, y2, y3…, membangun FA3 dengan state z1, z2, z3…dimana untuk setiap z dibentuk dari " xi or yi ". Jika bagian x atau y adalah inal state maka korespondensi z adalah final state. Formulasi transisi z

Z baru setelah abjad p = [x baru setelah abjad p] or [y baru setelah abjad p]

Teladan

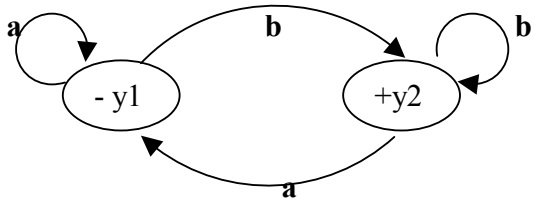
1. FA1 menerima untai dengan double aa dan FA2 menerima semua untai berakhiran b.

FA1 :



r1 = (a + b)*aa(a+b)*

FA2 :



r2 = a*bb*(aa*bb*)*

Copyright © 2003 IlmuKomputer.Com

maka FA3 :

-z1 = x1 or y1

pada z1 jika input a, maka terjadi S(x2 or y1) = z2.

Pada z1 ika input b, maka terjadi S(x1 or y2) = z3

merupakan final state.

Pada z2 jika input a, maka terjadi S(x3 or y1) = z4

merupakan final state.

Pada z2 jika input b, maka terjadi S(x1 or y2) = z3

Pada z3 jika input a, maka terjadi S(x2 or y1) = z2

Pada z3 jika input b, maka terjadi S(x1 or y2) = z3

Pada z4 jika input a, maka terjadi S (x3 or y1) = z4

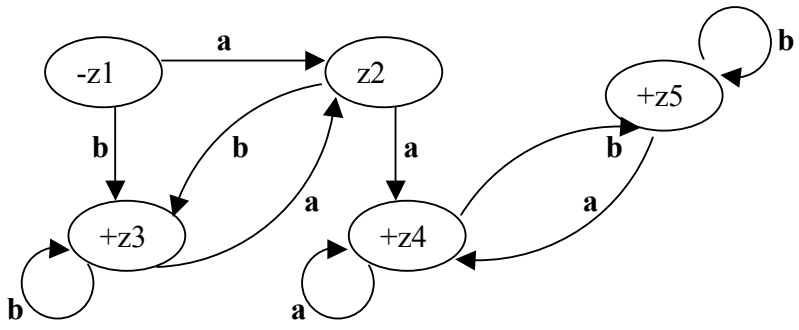
Pada z4 jika input a, maka terjadi S(x3 or y2) = z5,

merupakan final state.

Pada z5 jika input a, maka terjadi S(x3 or y1) = z4

Pada z5 jika input b, maka terjadi S(x3 or y2) = z5

Konstruksi mesin baru (FA3) yang menerima ER (r1 +r2):



Copyright © 2003 IlmuKomputer.Com

2.2.3.3 Rule 3

Jika terdapat suatu FA yaitu FA1 yang menerima bahasa yang didefinisikan oleh ER r1dan terdapat suatu FA yaitu FA2 yang menerima bahasa yang didefinisikan oleh ER r2, maka terdapat suatu FA yaitu FA3 yang menerima bahasa yang didefinisikan oleh ER (r1.r2)

Algoritma mesin FA3 ->FA1dengan state x1, x2, x3… dan FA2 dengan state y1, y2, y3…, membangun FA3 dengan state z1, z2, z3.

Pertama kali kita buat state z untuk setiap nonfinal x pada FA1, untuk setiap Final state di FA1 kita munculkan suatu z yang mengekspresikan pilihan apakah berlanjut pada FA1 atau memulai pada FA2, maka formulasinya :

Apakah pada xi berlanjut di FA1

Or

Baru memulai y1dan berlanjut ke FA2

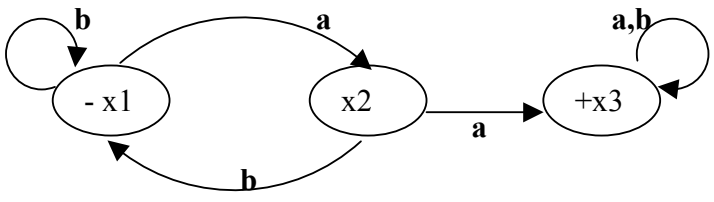
Or

Apakah pada yi berlanjut pada FA2

Teladan

2. FA1 menerima untai dengan double aa dan FA2 menerima semua untai berakhiran b.

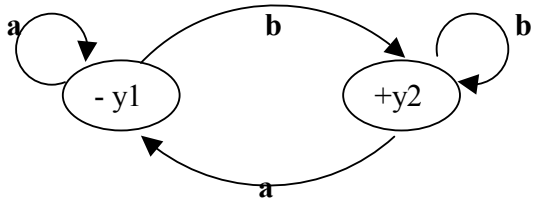
FA1 :



r1 = (a + b)*aa(a+b)*

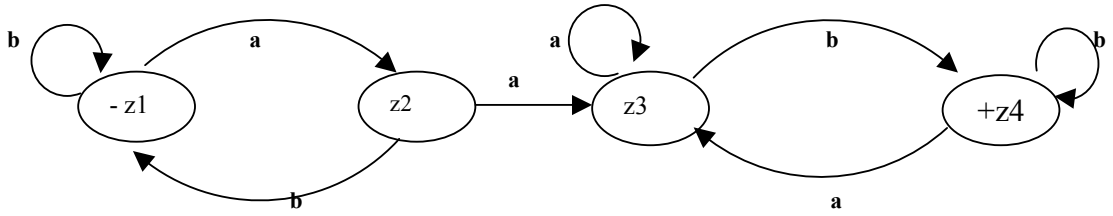
Copyright © 2003 IlmuKomputer.Com

FA2 :



r2 = a*bb*(aa*bb*)*

Maka konstruksi mesin FA3 yang menerima bahasa yang didefinisiskan oleh ER(r1r2)



2.2.3.4 Rule 4

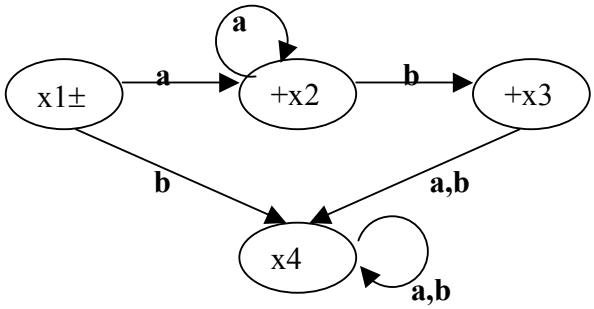
Jika r adalah ER dan FA1 menerima bahasa yang didefinisikan oleh r, maka terdapat FA yaitu FA2 yang akan menerima bahasa yang didefinisikan oleh r*.

Langkah awal yang sangat penting adalah mengkondisikan start state pada mesin baru agar dapat menerima null string Λ. Karena inilah ciri yang mendasar yang membedakan antara suatu ER tertentu dengan clousurenya. Karena berada didalam clousurenya maka pada setiap + state mesin lama secara otomatis memungkinkan start ulang menuji posisi awalnya.

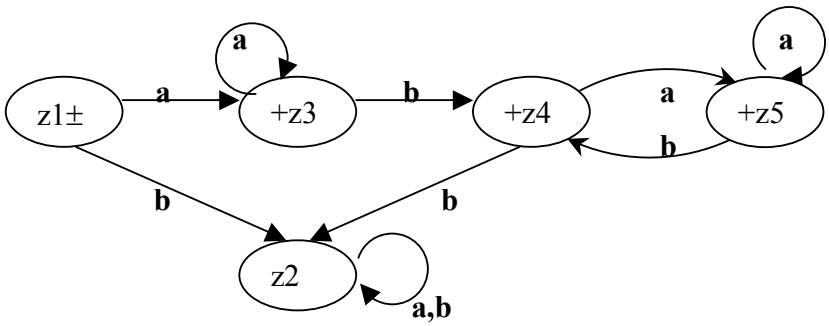
Copyright © 2003 IlmuKomputer.Com

Teladan : 1.

FA1(r = a*+aa*b)



Maka Konstruksi mesin FA2 :



r* = (a*+aa*b)

Copyright © 2003 IlmuKomputer.Com

3. PENUTUP

Telah dibuktikan secara constructive bahwa terdapat kesesuaian antara Finite Automata, Graph transisi, dan Ekspresi Reguler didalam mendifinisikan suatu bahasa formal dimana suatu bahasa yang diterima oleh FA dapat diterima oleh Graf Transisi, bahasa yang diterima oleh TG diterima oleh Ekspresi Reguler, dan Bahasa yang diterima oleh ER dapat diterima oleh FA.

4. REFERENSI

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

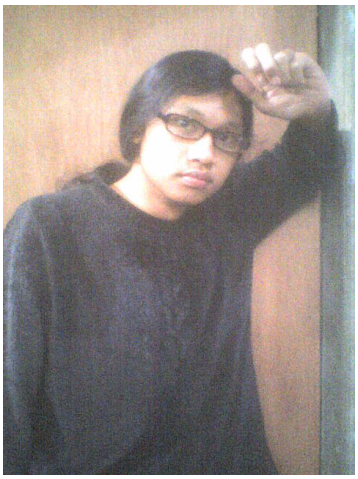
Kelley, Dean. Otomata Dan Bahasa Formal, PT.Prenhalindo, Jakarta : Indonesia

http://nexus.cs.usfca.edu /~parrt/course/652/lectures/formal.language.html

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

Copyright © 2003 IlmuKomputer.Com

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 indahnya Sastra Inggris – ala Humaniora dan pelabuhan terakhir tersintesakan dalam Ilmu Komputer - ala Science Universitas Brawijaya Malang. Bersyukur bisa menikmati Indahnya 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 memainkan musiknya….

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

Alamat Kontak :
Zira_ilkom@yahoo.com