

MODUL 3: FINITE AUTOMATA

DEFINISI FA

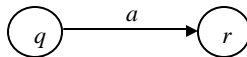
Suatu Finite Automaton (FA) atau kadang-kadang disebut Finite State Automaton (FSA) adalah mesin yang dapat mengenai bahasa reguler tanpa menggunakan storage/memory. Kecuali, sejumlah status dapat didefinisikan pada mesin untuk "mengingat" beberapa hal secara terbatas. Secara formal FA didefinisikan sebagai berikut.

Definisi: Suatu finite automaton (FA) atau Mesin Status-Berhingga adalah 5-tuple $(Q, \Sigma, q_0, A, \delta)$, yang mana:

- Q adalah himpunan berhingga dari **status**
- Σ himpunan berhingga **alfabet** dari simbol masukan
- $q_0 \in Q$ adalah status **inisial** (awal)
- $A \subseteq Q$ yaitu himpunan status **menerima** (accepting state, kadang-kadang disebut **final state**)
- δ adalah **fungsi transisi** yang memetakan $Q \times \Sigma$ ke Q (ditulis $\delta: Q \times \Sigma \rightarrow Q$)

Baris terakhir dijelaskan sebagai berikut. Untuk status-status $q, r \in Q$ dan suatu simbol $a \in \Sigma$ jika $\delta(q, a) = r$ terdefinisi, maka saat mesin berada dalam status q menerima masukan simbol a , mesin akan berubah/bertransisi ke status r .

Dengan diagram status digambarkan dengan bulatan dan transisi dengan panah, sbb:



Untuk pembahasan berikutnya pernyataan

“M adalah suatu FA dengan himpunan status Q , masukan simbol alfabet Σ , dan status awal q_0 , serta himpunan status menerima A , dengan fungsi transisi δ ,”

kita singkat menjadi

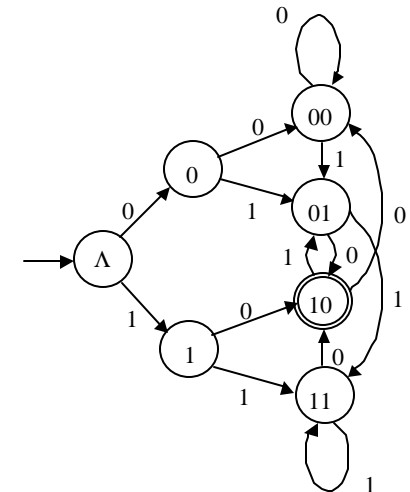
“ $M = (Q, \Sigma, q_0, A, \delta)$ merupakan FA.”

Contoh: bahasa reguler $L \subseteq \{0,1\}^*$ yang setiap stringnya selalu memiliki akhiran 10 Bahasa ini bisa dinyatakan dengan ekspresi reguler $(0+1)^*10$. Untuk mengenalinya maka mesin memerlukan

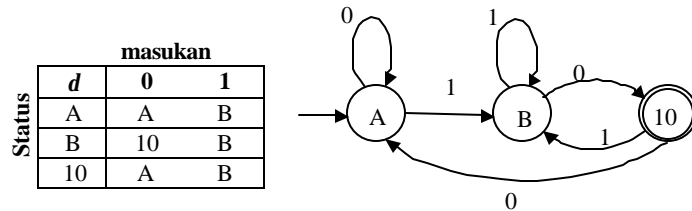
- empat status untuk menyimpan keempat kemungkinan dua simbol terakhir -- kita sebut status-status tsb. 00, 01, 10, dan 11;
- dua status ketika baru menerima satu simbol pertama -- kita sebut status-status 0 dan 1; dan
- satu status untuk mesin berada di awal (belum menerima simbol) -- kita sebut status Λ .

Diagram transisi FA dan tabel transisinya dapat digambarkan sebagai berikut.

		masukan	
δ		0	1
Λ		0	1
0		00	01
1		10	11
00		00	01
01		10	11
10		00	01
11		10	11



Dalam Topik FA Minimum akan dibahas apakah suatu FA sudah mencapai bentuk minimum (jumlah status paling sedikit). FA di atas belum minimum karena dapat disederhanakan lagi menjadi FA dengan tiga status saja, sbb.



Perluasan fungsi transisi d menjadi d^*

Bila $d(q, a)$ menyatakan status berikutnya setelah q dan menerima input simbol a , maka $d^*(q, x)$ menyatakan status berikutnya setelah q dan menerima input string simbol x . String x adalah string dari simbol-simbol dalam alfabet; x bisa juga Λ .

Jika $x = a_1 a_2 \dots a_{k-1} a_k$ maka $d^*(q, x)$ adalah aplikasi dari fungsi transisi secara berurutan demi simbol pada x , dengan kata lain,

$$d^*(q, x) = d d d \dots (d d d(q, a_1), a_2) \dots (a_{k-1}), a_k).$$

Untuk lebih formal maka kita definisikan sbb.

Definisi: $M = (Q, \Sigma, q_0, A, d)$ merupakan FA. Terdefinisi suatu fungsi d^* yang memetakan $Q \times \Sigma^*$ ke Q dengan sifat sebagai berikut:

- untuk setiap $q \in Q$, berlaku $d^*(q, \Lambda) = q$
- untuk setiap $y \in \Sigma^*$, $a \in \Sigma$, dan $q \in Q$, berlaku $d^*(q, ya) = d(d^*(q, y), a)$

Contoh: Jika $d(q, a) = q_1$, $d(q_1, b) = q_2$, dan $d(q_2, c) = q_3$, maka $d^*(q, abc) = d d d(q, a, b, c) = d d d(d(q, a), b), c) = d d d(d(d(q, \Lambda), a), b), c) = d d d(q, a), b, c) = d d(q_1, b), c) = d(q_2, c) = q_3$.

Menurut definisi di atas $d^*(q, a)$ harus dijabarkan sebagai $d d^*(q, \Lambda), a)$. Namun, sebagai implikasi dari definisi maka dapat kita simpulkan bahwa untuk setiap $a \in$

Σ , dan $q \in Q$, berlaku $d^*(q, a) = d(q, a)$. Jadi untuk penggunaan string dengan panjang satu (1 simbol), maka kedua fungsi bisa dituliskan bertukaran.

Selain itu dapat disimpulkan pula dari definisi tersebut bahwa jika $x, y \in \Sigma^*$ maka $d^*(q, xy) = d^*(d^*(q, x), y)$. Ini bisa and buktikan dengan induksi matematis (silakan mencoba sebagai latihan!). Dari contoh di atas dapat diperlihatkan pula bahwa $d^*(q, abc) = d^*(d^*(q, ab), c)$.

FA Sebagai Recognizer dari Bahasa Regular

Dalam definisi suatu FA terdapat A yaitu himpunan status menerima. Apabila suatu string masukan membawa status FA dari status inisial ke salah satu status dalam A maka string tersebut *diterima* (*dikenali*) sebagai anggota bahasa ybs. Secara formal kita definisikan sbb.

Definisi Penerimaan String: Suatu $M = (Q, \Sigma, q_0, A, d)$ merupakan suatu FA. Suatu string x dikatakan dikenal (diterima) oleh M jika $d^*(q_0, x) \in A$. Jika suatu string tidak diterima maka string itu dikatakan tidak dikenal (ditolak) oleh M .

Definisi Penerimaan Bahasa: Suatu bahasa yang dikenal (atau diterima) oleh M adalah himpunan

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

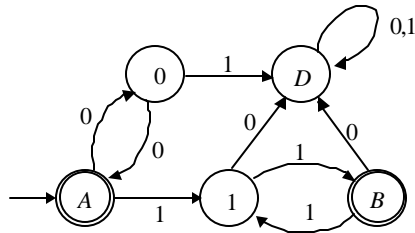
L adalah bahasa pada Σ dan L dikenal (atau diterima) oleh M , jika dan hanya jika $L = L(M)$.

Relasi “jika dan hanya jika” di atas berarti suatu bahasa regular L dikatakan dikenal oleh M jika semua string dari L dikenali oleh M , serta sebaliknya, setiap string dari suatu bahasa regular L harus dikenal oleh M jika L tersebut dikenal oleh M dan menolak setiap string dalam L .

Teorema. Bahasa L pada Σ adalah regular jika dan hanya jika terdapat suatu FA yang mengenali L .

Teorema ini juga menyatakan bahwa untuk setiap M , yaitu sebarang FA, terdapat ekspresi regular yang terkait dengan $L(M)$; dan di lain pihak, untuk regular ekspresi r tersebut, terdapat suatu FA yang mengenal bahasa tsb.

Contoh 1: Di samping ini diagram suatu FA dan kita akan mendapatkan ekspresi regular dari bahasa L yang dikenal FA tersebut.

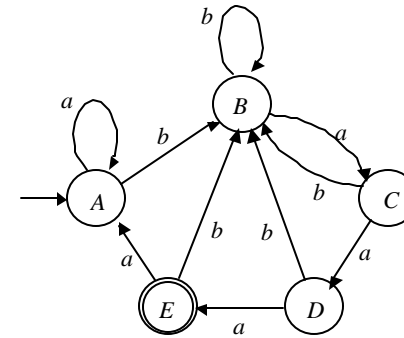


- Status A merupakan status inisial sekaligus status menerima, maka $\Lambda \in L$. Setiap string x dimana $\delta^*(A, x) = A$ adalah deretan simbol 0 dengan panjang merupakan bilangan genap. Kedua hal jika digabungkan menghasilkan ekspresi regular $(00)^*$.
- Status D adalah status yang tidak akan pernah bertransisi ke status menerima. Jadi suatu string x dimana $\delta^*(A, x) = D$, tidak akan pernah menjadi prefiks dari string dalam L .
- Status terima B hanya dapat dicapai melalui status 1 yang langsung dicapai dari status A dengan jumlah genap simbol 1. Jika tidak terjadi transisi berulang ke status A (tidak revisit A) maka x memiliki juga ekspresi regular $(11)^+$. Jika terjadi transisi berulang ke status A maka ekspresi regularnya menjadi $(00)^*(11)^+$.
- Menggabungkan kedua status regular di atas maka:

$$(00)^* + (00)^*(11)^+ = (00)^*(\Lambda + (11)^+) = (00)^*(11)^*$$

Contoh 2: Sekarang perhatikan diagram berikut ini dari suatu FA untuk string $\in \{a, b\}^*$. Dalam FA tersebut status terima hanya satu yaitu E . Status ini hanya dapat dicapai dari D dengan simbol a . D hanya dapat dicapai dari C dengan simbol a .

C hanya dapat dicapai dari B dengan simbol a .



B pertama kali dicapai dari A dengan simbol b . Sementara pada status mana pun setiap simbol b akan selalu membawa ke status B . Jadi setiap string yang membawa FA ke status terima E harus memiliki sufiks *baaa*. Ekspresi regularnya adalah:

$$(a+b)^*baaa$$

Cara untuk mendapatkan ekspresi regular di atas adalah dengan cara yang kira-kira (intuitif). Cara yang lebih sistematis akan dibahas di topik mendatang.

OPERASI-OPERASI HIMPUNAN

Misalkan L_1 dan L_2 keduanya merupakan bahasa-bahasa regular pada alfabet Σ . Sudah tentu sesuai dengan Teorema 3.1 terdapat FA M_1 dan M_2 yang masing-masing mengenal L_1 dan L_2 . Sesuai definisi Bahasa Regular maka bahasa $L_1 \cup L_2$ adalah juga bahasa regular, yang juga akan dikenal oleh suatu FA. Bagaimana dengan irisan, komplement dan difference serta operasi-operasi himpunan lain?

Untuk suatu bahasa regular L , maka jelas L' (komplement dari L) adalah bahasa regular karena jika $M = (Q, \Sigma, q_0, A, \delta)$ adalah FA untuk mengenal L maka $M' = (Q, \Sigma, q_0, B, \delta)$ dapat dibuat dengan menjadikan semua status yang bukan menerima di M menjadi menerima di M' dan sebaliknya semua status menerima di M menjadi bukan status menerima M' . Atau, $B = Q - A$.

Setiap operasi himpunan lain dapat dinyatakan sebagai operasi gabungan dan komplemen. Misalnya $L_1 \cap L_2 = (L_1' \cup L_2)'$, dan $L_1 - L_2 = (L_1' \cup L_2)'$. Dengan demikian setiap operasi himpunan pada beberapa bahasa regular akan menghasilkan bahasa regular. Dengan demikian juga akan ada mesin FA yang dapat mengenalnya.

Masalahnya sekarang dapatkah kita membentuk mesin FA dari bahasa hasil operasi himpunan tersebut berdasarkan kombinasi mesin-mesin bahasa asalnya secara langsung?

Berikut ini akan dibahas pembentukan FA untuk bahasa penggabungan kedua bahasa ($L_1 \cup L_2$). Untuk kasus irisan ($L_1 \cap L_2$) dan perbedaan ($L_1 - L_2$) dapat dilakukan dengan sedikit modifikasi.

Diberikan $M_1 = (Q_1, \Sigma, q_1, A_1, \mathbf{d}_1)$ dan $M_2 = (Q_2, \Sigma, q_2, A_2, \mathbf{d}_2)$ untuk mengenal masing L_1 dan L_2 . Suatu FA $M = (Q, \Sigma, q_0, A, \mathbf{d})$ akan dibentuk dari M_1 dan M_2 untuk dapat mengenal ($L_1 \cup L_2$).

Ide dari pembentukan M adalah bahwa saat memeriksa suatu masukan string $x \in (L_1 \cup L_2)$ atau bukan adalah dengan melakukan sekaligus kedua pemeriksaan apakah $x \in L_1$ dan $x \in L_2$. Jadi dalam M akan terdapat status-status yang merupakan kombinasi antara status-status yang ada dalam M_1 dengan M_2 , dpl. untuk $p \in Q_1, q \in Q_2$ akan ada status $(p, q) \in Q$. Demikian pula status inisial q_0 adalah (q_1, q_2) . Bila M berada dalam status (p, q) , bila ia menerima simbol a maka ia akan bertransisi ke $(\mathbf{d}_1(p, a), \mathbf{d}_2(q, a))$.

Himpunan status menerima A adalah himpunan status (p, q) yang mana p atau q (salah satu atau keduanya!) adalah status menerima dari mesin-mesin asalnya.

Perbedaan untuk FA bahasa-bahasa bentukan dari operasi irisan ($L_1 \cap L_2$) dan perbedaan ($L_1 - L_2$) adalah terletak pada himpunan status menerima A tsb.

- Untuk irisan ($L_1 \cap L_2$), himpunan status menerima A adalah himpunan status (p, q) yang mana p dan q (harus keduanya!) adalah status menerima dari mesin-mesin asalnya.

- Untuk perbedaan ($L_1 - L_2$), himpunan status menerima A adalah himpunan status (p, q) yang mana p adalah status menerima dari M_1 dan q bukan status menerima dari M_2 .

Untuk lebih formal maka berikut ini teorema mengenai bahasa-bahasa operasi-operasi penggabungan, irisan dan perbedaan tersebut.

Teorema 3.4. Misalkan $M_1 = (Q_1, \Sigma, q_1, A_1, \mathbf{d}_1)$ dan $M_2 = (Q_2, \Sigma, q_2, A_2, \mathbf{d}_2)$ masing-masing mengenal L_1 dan L_2 . Bila $M = (Q, \Sigma, q_0, A, \mathbf{d})$ suatu FA dimana

$$Q = Q_1 \times Q_2$$

$$q_0 = (q_1, q_2)$$

\mathbf{d} terdefinisi sebagai $\mathbf{d}(p, q, a) = (\mathbf{d}_1(p, a), \mathbf{d}_2(q, a))$

untuk setiap $p \in Q_1, q \in Q_2$ dan $a \in \Sigma$ maka,

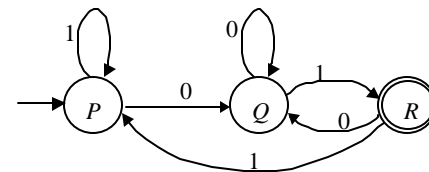
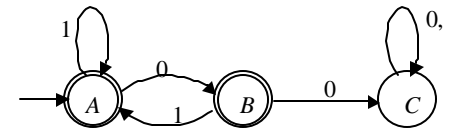
1. Bila $A = \{(p, q) \mid p \in A_1 \text{ atau } q \in A_2\}$, M mengenal bahasa ($L_1 \cup L_2$)
2. Bila $A = \{(p, q) \mid p \in A_1 \text{ dan } q \in A_2\}$, M mengenal bahasa ($L_1 \cap L_2$)
3. Bila $A = \{(p, q) \mid p \in A_1 \text{ dan } q \notin A_2\}$, M mengenal bahasa ($L_1 - L_2$)

Contoh: misalkan L_1 dan $L_2 \subseteq \{0, 1\}^*$, dengan

$$L_1 = \{x \mid \text{tidak ada substring } 00 \text{ dalam } x\}$$

$$L_2 = \{x \mid x \text{ berakhiran } 10\}$$

Kedua bahasa ini sudah dibahas masing-masing sebagai contoh sebelumnya dengan diagram transisi sbb.



Masing-masing memiliki tiga status sehingga mesin baru yang dibentuk dari kedua mesin tsb. berisikan 9 status sbb. (untuk menyederhanakan penulisan maka kita tuliskan (A,P) sebagai AP , dst.).

$Q = \{AP, AQ, AR, BP, BQ, BR, CP, CQ, CR\}$

Status awal AP

dari $d_1(A, 0) = B$ dan $d_2(P, 0) = Q$, maka terdapat $d(AP, 0) = BQ$

dari $d_1(A, 1) = A$ dan $d_2(P, 1) = P$, maka terdapat $d(AP, 1) = AP$

Selanjutnya kita melihat BQ ,

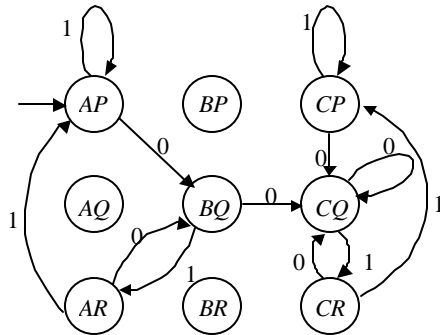
dari $d_1(B, 0) = C$ dan $d_2(Q, 0) = Q$, maka terdapat $d(BQ, 0) = CQ$

dari $d_1(B, 1) = A$ dan $d_2(Q, 1) = R$, maka terdapat $d(BQ, 1) = AR$

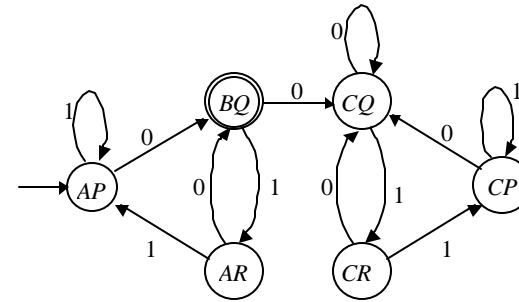
Berikutnya dari CQ ,

dari $d_1(C, 0) = C$ dan $d_2(Q, 0) = Q$, maka terdapat $d(CQ, 0) = CQ$

dari $d_1(C, 1) = C$ dan $d_2(Q, 1) = R$, maka terdapat $d(CQ, 1) = CR$ dan seterusnya maka akan diperoleh diagram transisi berikut ini.



Dalam diagram terlihat hanya enam status dari sembilan yang dapat tercapai dari AP . Jadi ketiga status lainnya bisa dihilangkan. Jika bahasa bentukan baru tersebut adalah $L_1 - L_2$ maka himpunan status menerima A dapat dilakukan sehingga $A = \{AP, BQ\}$ dan digambarkan pada diagram berikut.



Penyederhanaan yang dibahas sebelumnya dapat dilakukan. Status CR dan CP memiliki transisi yang sama dapat digabungkan menjadi satu. Selanjutnya status hasil gabungan ini juga memiliki transisi yang sama dengan status CQ sehingga akhirnya menghasilkan satu status gabungannya. Selanjutnya diagram setelah penyederhanaan diperlihatkan sebagai berikut.

