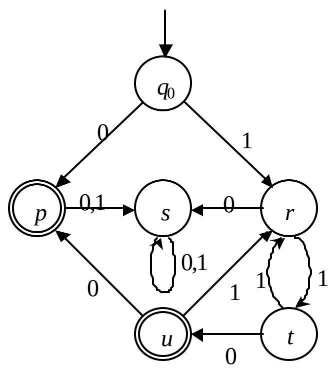
MODUL 4: NONDETERMINISTIC FINITE AUTOMATA (NFA)

Menemukan FA dari suatu bahasa regular seperti dalam pembahasan sebelumnya umumnya tidak semudah seperti itu. Dalam bagian ini kita akan mempelajari kelas-kelas FA yang memiliki sifat-sifat khusus tertentu sehingga kita dapat secara lebih langsung mendapatkan FA tersebut dari ekspresi regularnya.

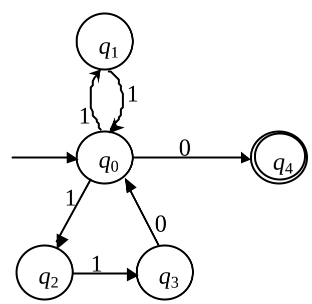
Pada contoh sebelumnya kita sudah mencoba mendapatkan FA (sebagai pada diagram transisi berikut ini) dari suatu bahasa regular dalam $\{0,1\}^*$ dengan ekspresi regularnya adalah $(11+110)^*0$.



Sekarang perhatikan diagram berikut ini.

Diagram transisi ini jelas bukan FA karena tidak memenuhi beberapa sifat FA:

- adanya status-status yang tidak memiliki transisi untuk kedua kemungkinan simbol, dan
- adanya status-status yang memiliki lebih dari satu transisi untuk simbol yang sama.



Namun diagram tsb. menggambarkan secara lebih eksplisit ekspresi regular $(11+110)^*0$.

- loop q_0-q_1-q_0 menunjukkan kemunculan berulang substring 11 dan
- loop q_0-q_2-q_3-q_0 menunjukkan kemunculan berulang substring 110 dengan masing-masing jumlah kemunculan ≥ 0 kali, serta
- transisi terakhir ke status terima adalah adanya simbol 0 di akhir string.

Bila diagram tsb. berfungsi sebagai recognizer maka setiap transisi yang dilakukan tidak selalu dengan pasti dilakukan dari satu status ke status yang lain melainkan dari satu status ke "sejumlah kemungkinan status". Kita katakan bahwa $\delta(q, a)$ adalah transisi ke sejumlah kemungkinan status (dalam himpunan P yaitu subset dari Q dan P bisa himpunan kosong).

Bayangkan jika mesin adalah algoritma multihreading maka setiap cabang akan dijalani secara paralel dan jika ada salah satu thread yang mencapai status menerima saat simbol terakhir dibaca maka mesin mengenali string tersebut.

Karena adanya ketidak pastian tersebut (baca: sejumlah alternatif langkah) maka recognizer tersebut adalah suatu "FA yang memungkinkan adanya ketidak pastian". Kita namakan jenis mesin ini Nondeterministic Finite Automata (NFA).

Definisi NFA

Definisi: Suatu Nondeterministic Finite Automaton (NFA) adalah 5-tuple $M = (Q, \Sigma, q_0, A, d)$. Q adalah himpunan status, Σ adalah alfabet, q_0 adalah status inisial, A adalah himpunan status terima dan d fungsi transisi sebagai berikut.

$$d: Q \times \Sigma \rightarrow 2^Q$$

Fungsi Perluasan Transisi d^*

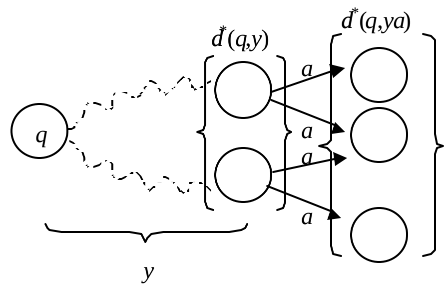
Sebagaimana pada FA (seharusnya disebut Deterministic Finite Automata - DFA), kita perlu fingsi perluasan transisi δ^* agar pada pembahasan selanjutnya penulisan-penulisan bisa dipersingkat. Pada NFA δ^* didefinisikan sebagai fungsi perluasan transisi sebagai pemetaan $d^*: Q \times \Sigma^* \rightarrow 2^Q$. Secara rekursif bisa juga dinyatakan bahwa $d^*(p, xa). = d\,(d^*(p, x), a)$ seperti halnya pada DFA, kecuali bahwa dalam NFA $d^*(p, \Lambda). = \{p\}$.

Definisi: Pada setiap NFA $M = (Q, \Sigma, q_0, A, d)$ fungsi $d^*: Q \times \Sigma^* \rightarrow 2^Q$ didefinisikan sbb.

- Untuk setiap $q \in Q$ maka $d^*(q, \Lambda) = \{q\}$.
- Untuk setiap $y \in \Sigma^*, a \in \Sigma$, dan $q \in Q$, maka

$$d^*(q, ya) = \bigcup_{p \in d^*(q,y)} d(p, a)$$

Dalam bentuk diagram dapat digambarkan sbb.



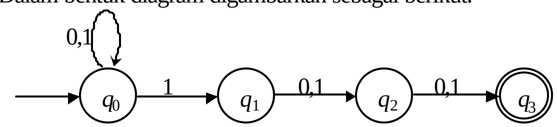
NFA Sebagai Mesin Pengenal Bahasa Regular

Definisi: Diberikan $M = (Q, \Sigma, q_0, A, d)$ suatu NFA, String x dikenal oleh mesin M bila $d^*(q_0, x) \cap A \neq \emptyset$. Bahasa L diterima atau dikenali oleh M adalah $L(M)$ himpunan seluruh string yang diterima oleh M. Untuk bahasa $L \subseteq \Sigma^*$, L dikenali oleh M jika $L = L(M)$.

Contoh $M = (Q, \Sigma, q_0, A, d)$ suatu NFA, dengan $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{0, 1\}$, $A = \{q_3\}$, dan d dinyatakan dalam tabel berikut.

q	***d*(*q*, 0)**	***d*(*q*, 1)**
q_0	$\{q_0\}$	$\{q_0, q_1\}$
q_1	$\{q_2\}$	$\{q_2\}$
q_2	$\{q_3\}$	$\{q_3\}$
q_3	$\emptyset$	$\emptyset$

Dalam bentuk diagram digambarkan sebagai berikut.



Dari diagram nampak jelas bahwa ekspresi regular bahasa yang dikenali ybs adalah $(0+1)^*1(0+1)^2$. Bahasa regular ybs. sebelumnya telah dikenal sebagai bahasa $L_n = \{x$ Î Σ^* | simbol ke n dari kanan adalah 1} dengan $n = 3$.

Namun apakah NFA lebih powerful (powerful dalam hal kemampuan penerimaannya terhadap suatu bahasa) daripada FA? Ternyata tidak! Suatu bahasa yang dikenali oleh NFA akan dapat dikenali pula oleh suatu FA (walaupun dengan diagram transisi yang lebih rumit).

Teorema Konversi NFA ke FA

Teorema: untuk setiap NFA $M = (Q, \Sigma, q_0, A, d)$ yang menerima bahasa $L \subseteq \Sigma$, akan terdapat FA $M_1 = (Q_1, \Sigma, q_1, A_1, d_1)$ yang juga menerima L.

Bukti : Pembuktian dilakukan dengan adanya algoritma untuk mengkonbersi NFA ke FA sbb. $Q_1 = 2^Q$, $q_1 = \{q_0\}$, untuk $q \in Q_1$ dan $a \in \Sigma$,

maka $d_1(q, a) = \bigcup_{p \in q} d(p, a)$ dan $A_1 = \{q \in Q_1 \mid q \cap A \neq \emptyset\}$

Note: Status-status $q \in Q_1$ (dari M_1) mrpk status yang dinamai sesuai himpunan status $\subseteq 2^Q$.

Contoh: Dari $Q = \{A, B, C\}$ dibentuk $Q_1 = \{\emptyset, \{A\}, \{B\}, \{C\}, \{A, B\}, \{A, C\}, \{B, C\}, \{A, B, C\}\}$. Dalam hal ini $\{A\}$ atau $\{A, B, C\}$ merupakan **nama-nama** status anggota Q_1.

Untuk menyederhanakan kadang-kadang ditulis tanpa notasi kurung dan koma.
Contoh: $Q_1 = \{\emptyset, A, B, C, AB, AC, BC, ABC\}$

Dari teorema ini maka dapat pula disimpulkan bahwa $d_1^*(q_1, x) = d^*(q_0, x)$.
(Ingat: d_1^* dan d^* berasal dari dua definisi yang berbeda, yang pertama dari FA dan yang kedua dari NFA).

Bukti : dibuktikan dengan induksi matematis sbb.
Untuk $x = \Lambda$, $d_1^*(q_1, x) = d_1^*(q_1, \Lambda) = q_1 = \{q_0\}$, sementara itu juga $d^*(q_0, x) = d^*(q_0, \Lambda) = \{q_0\}$.
Untuk x yang lain, jika $d_1^*(q_1, x) = d^*(q_0, x)$, akan dibuktikan apakah untuk setiap $a \in \Sigma$ maka $d_1^*(q_1, xa) = d^*(q_0, xa)$. Ternyata,

$$d_1^*(q_1, xa) = d_1(d_1^*(q_1, x), a) = d_1(d^*(q_0, x), a) = \bigcup_{p \in d^*(q_0, x)} d(p, a) = d^*(q_0, xa)$$

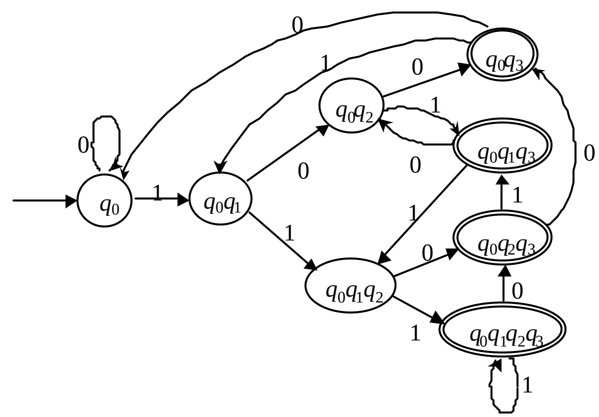
Teorema dan cara pembuktian tsb. menghasilkan suatu konsep algoritma menghilangkan sifat nondeterministisme dari NFA menjadi FA.

Contoh: dari contoh NFA sebelumnya untuk bahasa regular dengan ekspresi regular $(0+1)^*1(0+1)^2$ akan dicari FA ybs. Mulai dari status inisial $\{q_0\}$ diperoleh
$d_1(\{q_0\}, 0) = \{q_0\}$, $d_1(\{q_0\}, 1) = \{q_0, q_1\}$

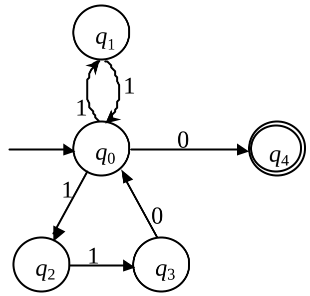
lanjutkan dengan status $\{q_0, q_1\}$ diperoleh
$d_1(\{q_0, q_1\}, 0) = d(\{q_0\}, 0) \cup d(\{q_1\}, 0) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}$
$d_1(\{q_0, q_1\}, 1) = d(\{q_0\}, 1) \cup d(\{q_1\}, 1) = \{q_0, q_1\} \cup \{q_2\}$
$= \{q_0, q_1, q_2\}$

lanjutkan dengan status $\{q_0, q_2\}$ diperoleh
$d_1(\{q_0, q_2\}, 0) = d(\{q_0\}, 0) \cup d(\{q_2\}, 0) = \{q_0\} \cup \{q_3\} = \{q_0, q_3\}$
$d_1(\{q_0, q_2\}, 1) = d(\{q_0\}, 1) \cup d(\{q_2\}, 1) = \{q_0, q_1\} \cup \{q_3\}$
$= \{q_0, q_1, q_3\}$

lanjutkan dengan status $\{q_0, q_1, q_2\}$ diperoleh
$d_1(\{q_0, q_1, q_2\}, 0) = d(\{q_0\}, 0) \cup d(\{q_1\}, 0) \cup d(\{q_2\}, 0)$
$= \{q_0\} \cup \{q_2\} \cup \{q_3\} = \{q_0, q_2, q_3\}$
$d_1(\{q_0, q_1, q_2\}, 1) = d(\{q_0\}, 1) \cup d(\{q_1\}, 1) \cup d(\{q_2\}, 1)$
$= \{q_0, q_1\} \cup \{q_2\} \cup \{q_3\}$
$= \{q_0, q_1, q_2, q_3\}$
dst.
Akan diperoleh total delapan status dan lima status berikutnya adalah $\{\{q_0, q_3\}, \{q_0, q_1, q_2\}, \{q_0, q_1, q_3\}, \{q_0, q_2, q_3\}, \{q_0, q_1, q_2, q_3\}\}$.
Diagram transisinya adalah sbb.



Contoh: NFA di awal pembahasan yaitu NFA untuk mengenali bahasa regular dengan ekspresi regular $(11+110)^*0$.



Jika NFA dikonversi menjadi FA dengan algoritma di atas atas maka akan diperoleh himpunan status $\{\{q_0\}, \{q_4\}, \{q_1, q_2\}, \varnothing, \{q_0, q_3\}, \{q_0, q_4\}\}$ dengan tabel dan diagram transisi sbb. Yang sama dengan diagram transisi pada pembahasan contoh tersebut di awal.

q	***d* (*q*, 0)**	***d* (*q*, 1)**
$\{q_0\}$	$\{q_4\}$	$\{q_1, q_2\}$
$\{q_4\}$	$\varnothing$	$\varnothing$
$\{q_1, q_2\}$	$\varnothing$	$\{q_0, q_3\}$
$\varnothing$	$\varnothing$	$\varnothing$
$\{q_0, q_3\}$	$\{q_0, q_4\}$	$\{q_1, q_2\}$
$\{q_0, q_4\}$	$\{q_4\}$	$\{q_1, q_2\}$