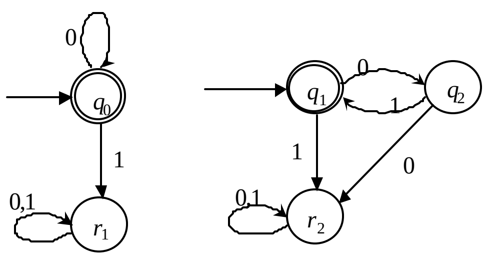
MODUL 5: NONDETERMISNISTIC FINITE STATE AUTOMATA DENGAN TRANSISI-Λ

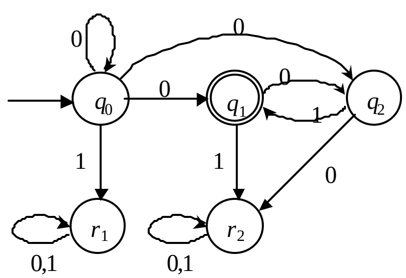
TRANSISI-Λ

Dengan konsep nondeterministisme dari suatu ekspresi regular suatu NFA yang dapat menerima bahasa ybs dapat langsung dilakukan. Namun, masih terdapat kasus-kasus dimana hal ini tidak dapat dilakukan secara sederhana.

Contoh: $0^*(01)^*$ dapat dipandang sebagai sebagai konkatenasi dua ekspresi regular 0^* dan $(01)^*$. Bahasa-bahasa dari ekspresi-ekspresi tsb diterima oleh FA-FA sebagai berikut.



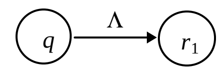
Bahasa dari ekspresi regular $0^*(01)^*$ dapat dikenal oleh NFA yang dibentuk dari kedua FA di atas sebagai berikut ini.



Transisi dari q_0 ke q_1 diperlukan untuk menerima 0^* sementara transisi dari q_0 ke q_2 diperlukan untuk menerima $(01)^*$.

Jika diperhatikan maka aturan penambahan transisi yang melengkapi "konkatenasi" kedua FA semula tidaklah bisa dengan sederhana dijelaskan. Dalam konkatenasi lain bahkan bisa lebih rumit apabila status menerima FA pertama berjumlah cukup banyak.

Seandainya dalam NFA bisa didefinisikan transisi tanpa simbol masukan (kita sebut **transisi-L**) dituliskan sebagai $d(q, \Lambda) = p$, atau dengan diagram



maka kontatenasi tersebut dapat dilakukan hanya seperti pada diagram berikut ini.

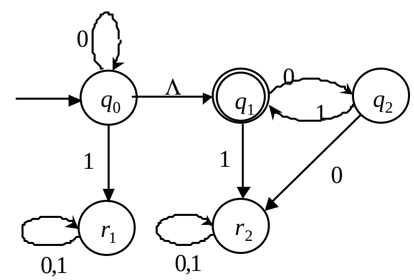


Diagram tsb. bukanlah NFA seperti yang telah didefinisikan sebelumnya, tetapi suatu NFA dengan transisi Λ (ditulis NFA-Λ). NFA-Λ ini dapat mengenali bahasa yang dikenali oleh NFA sebelumnya namun dengan struktur yang lebih sederhana serta lebih mencerminkan konkatenasi kedua FA asal melalui transisi Λ. Selain konkatenasi nanti akan dilihat dapat mencerminkan operasi Kleene-* dan gabungan yang juga dapat lebih sederhana.

DEFINISI NFA-Λ

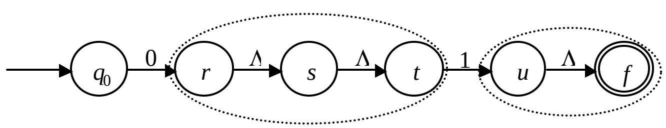
Definisi: suatu NFA dengan transisi-Λ (disingkat NFA-Λ) merupakan 5-tuple (Q, Σ, q_0, A, d) dimana Q dan Σ merupakan himpunan-himpunana terbatas, $q_o \in Q$ dan $A \subseteq Q$, serta d
terdefinisi sebagai pemetaan sbb.
$d: Q \times (\Sigma \cup \Lambda) \rightarrow 2^Q$

Perhatikan disini perbedaannya adalah domain dari d selain himpunan simbol Σ juga meliputi Λ. Dalam NFA $d: Q \times \Sigma \rightarrow 2^Q$ sementara dalam FA $d: Q \times \Sigma \rightarrow Q$

Fungsi perluasan transisi d^*

Demikian pula perluasan fungsi d^* berikut ini didefinisikan untuk memahami penerimaan suatu string oleh suatu NFA-Λ. Hal yang berbeda dengan sebelumnya adalah karena adanya transisi-Λ maka jumlah transisi bisa lebih banyak dari jumlah simbol dalam string sementara pada NFA jumlah simbol sama dengan jumlah transisi yang diaplikasikan pada d^*.

Contoh: diagram berikut dapat mengenali string 01 sebagai 0ΛΛ1Λ seolah sejumlah “simbol palsu” Λ disisipkan pada string (kita katakan “simbol palsu” karena Λ bukanlah simbol).



Dalam pendefinisiannya pun kita bayangkan adanya simbol-simbol palsu Λ tersebut tersisipkan pada string x dalam menentukan $d^*(q, x)$.

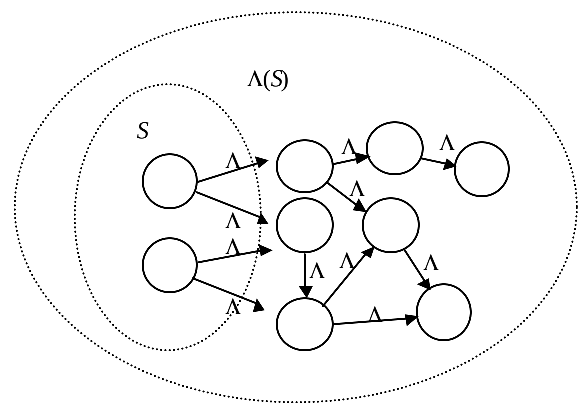
Untuk memahami definisi rekursif dari d^* maka akan diperkenalkan konsep Λ-closure (atau kita terjemahkan lingkupan-Λ) sbb.

L-closure dari Himpunan Status

Secara naratif maka Λ-closure dari suatu himpunan status S adalah seluruh status dalam S tersebut beserta status-status lain yang dapat tercapai oleh masing-masing status dalam S melalui transisi Λ. Jadi Λ-closure dari suatu himpunan S (ditulis Λ(S)) adalah merupakan himpunan juga yang di dalamnya terdapat S sebagai subset.

Definisi: Pada suatu NFA-Λ $M = (Q, \Sigma, q_0, A, d)$ dan $S \subseteq Q$, suatu fungsi $\Lambda(S)$ didefinisikan sebagai berikut:

- Setiap anggota S merupakan anggota $\Lambda(S)$.
- Untuk setiap $q \in \Lambda(S)$, setiap anggota $d(q, \Lambda)$ adalah anggota $\Lambda(S)$.
- Tidak ada anggota lain dari Q yang anggota $\Lambda(S)$ kecuali berdasar kedua pernyataan di atas.



Algoritma Pencarian **L**(*S*)

Dalam bentuk algoritma $\Lambda(S)$ dapat dicari sebagai berikut.

- Inisialisasi $T_1 = S$.
- Lakukan iterasi dari i =1, 2, 3,… , dan pada iterasi ke-i:
  - ➢ $T_{i+1} = T_i$
  - ➢ Untuk setiap $q \in T_i$, gabungkan $d(q, \Lambda)$ ke dalam T_{i+1}.
- Iterasi berhenti saat $T_{i+1} = T_i$ (tidak ada status baru yang bergabung dalam T).

Dengan definsi Λ(S) di atas maka definisi rekursif dari NFA-Λ sekarang dapat dituliskan sebagai berikut.

Definisi d^* untuk NFA-L

Pada setiap NFA-Λ $M = (Q, \Sigma, q_0, A, d)$ fungsi d^*: $Q \times (\Sigma^* \cup \Lambda) \rightarrow 2^Q$ didefinisikan sbb.
Untuk setiap $q \in Q$ maka

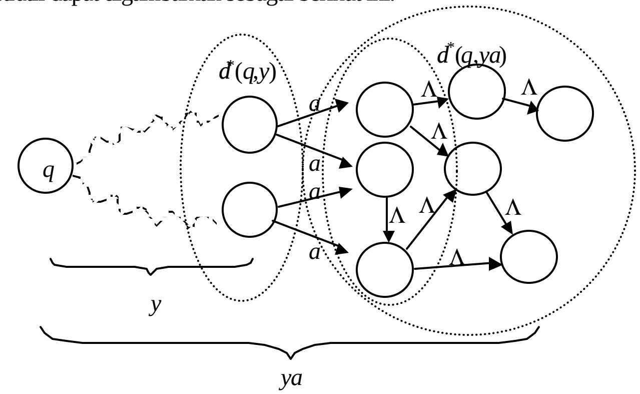
$d^*(q, \Lambda) = \Lambda(\{q\})$.

Dan untuk setiap $y \in \Sigma^*$, $a \in \Sigma$, dan $q \in Q$, maka

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

Formulasi rekusif di atas secara proseduralmenyatakan bahwa $d^*(q, ya)$ diperoleh pertama dengan menentukan $d^*(q, y)$, kemudian mendapatkan himpunan gabungan dari setiap $d(p, a)$ untuk setiap $p \in d^*(q, y)$, terakhir menentukan Λ( ) dari himpunan gabungan tersebut. Demikian juga jika $y = xb$ maka $d^*(q, y) = d^*(q, xb)$ diperoleh dengan menentukan $d^*(q, x)$, dan seterusnya mengulangi hal di atas. Apabila $y = \Lambda$ maka $d^*(q, y) = d^*(q, \Lambda) = \Lambda(\{q\})$.

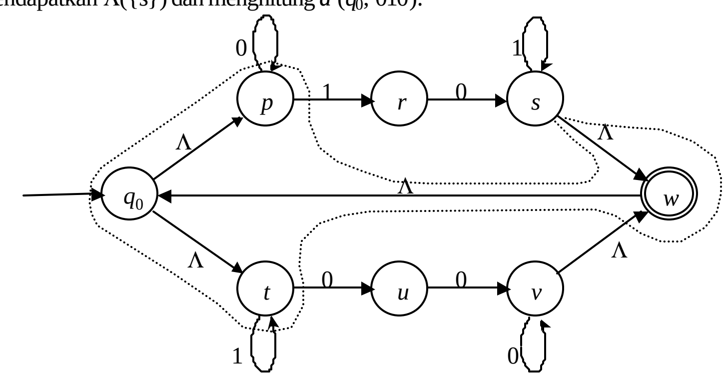
Secara ilustratif dapat digambarkan sebagai berikut ini.



Dengan terdefinisinya d^* maka suatu NFA-Λ M sebagai mesin pengenal bahasa dapat didefinisikan sebagai berikut.

Definisi: Untuk NFA-Λ $M = (Q, \Sigma, q_0, A, d)$ string x diterima oleh M bila $d^*(q_0, x) \cap A \neq \emptyset$ (ingat q_0 adalah initial state dan A adalah himpunan accepting state). Bahasa yang dikenali oleh M adalah $L(M)$ yaitu semua string yang diterima oleh M.

Contoh: suatu NFA-Λ ditunjukkan pada gambar berikut ini. Kita akan mendapatkan $\Lambda(\{s\})$ dan menghitung $d^*(q_0, 010)$.



Mula-mula $T = \{s\}$. Setelah iterasi pertama, $T = \{s, w\}$. Setelah iterasi kedua menjadi $\{s, w, q_0\}$ dan setelah iterasi ketiga menjadi $\{s, w, q_0, p, t\}$

Berdasarkan definisi rekursif kita dapatkan $d^*(q_0, 010)$ sbb.

$$
\begin{aligned}
d^*(q_0, \Lambda) &= \Lambda(\{q_0\}) = \{q_0, p, t\} \\
d^*(q_0, 0) &= \Lambda(d(q_0,0) \cup d(p,0) \cup d(t,0)) \\
&= \Lambda(\emptyset \cup \{p\} \cup \{u\}) = \Lambda(\{p, u\}) = \{p, u\} \\
d^*(q_0, 01) &= \Lambda(d(p,1) \cup d(u,1)) = \Lambda(\{r\}) = \{r\} \\
d^*(q_0, 010) &= \Lambda(d(r,0)) = \Lambda(\{s\}) = \{s, w, q_0, p, t\}
\end{aligned}
$$

Karena $d^*(q_0, 010)$ berisi w yang adalah status terima maka 010 diterima. Dalam hal ini transisi yang terjadi akibat string Λ010Λ adalah

$$q_0 \xrightarrow{\Lambda} p \xrightarrow{0} p \xrightarrow{1} r \xrightarrow{0} s \xrightarrow{\Lambda} w$$

KOMPATIBILITAS NFA-Λ DENGAN NFA

Sebelumnya telah ditunjukkan bahwa NFA tidak lebih powerful dari FA dalam kemampuannya mengenali bahasa. Setiap bahasa yang dapat dikenali oleh suatu NFA juga dapat dikenali oleh suatu FA. Dalam pembahasan hal ini dibuktikan dengan kenyataan bahwa daro setiap NFA dapat diturunkan suatu FA.

Bagaimana dengan NFA-Λ terhadap NFA atau FA? Apakah juga tidak lebih powerful? Kesamaan NFA-Λ dengan NFA ini dijelaskan dengan teorema dan pembuktiannya sebagai berikut yang berarti juga secara tidak langsung menjelaskan kesamaan NFA-Λ dengan FA.

Teorema: Bila $L \subseteq \Sigma^*$ suatu bahasa yang diterima NFA-Λ $M = (Q, \Sigma, q_0, A, d)$, maka akan terdapat suatu NFA $M_1 = (Q_1, \Sigma, q_1, A_1, d_1)$ yang juga menerima L.

Bukti : Bahwa NFA ekivalen dengan FA telah ditunjukkan dengan mensubstitusi semua pencabangan nondeterminisme dengan status.

Suatu transisi-Λ adalah jenis lain dari nondeterminisme. Misalnya dalam M terdapat transisi

$$p \xrightarrow{0} q \xrightarrow{\Lambda} r$$

Maka, dari p masukan simbol 0 akan membawa M baik ke q maupun ke r. Maka suatu NFA M_1 dapat dibuat dari M tsb. dengan menghapus setiap transisi-Λ dan menggantikannya dengan menambahkan transisi dari p ke r dengan masukan simbol 0. Dalam konversi ini himpunan status Q tidak berubah, demikian pula status inisialnya. Berikut ini adalah algoritma untuk mendapatkan A_1, dan d_1 dari NFA yang akan didapatka dari NFA-Λ dengan A, dan d.

Menentukan d_1 dari d

Perubahan yang terjadi pada fungsi transisi d menjadi d_1 adalah karena adanya <u>tambahan transisi-transisi yang menggantikan transisi-Λ</u> tsb.

Untuk setiap $q \in Q$ dan $a \in \Sigma$,

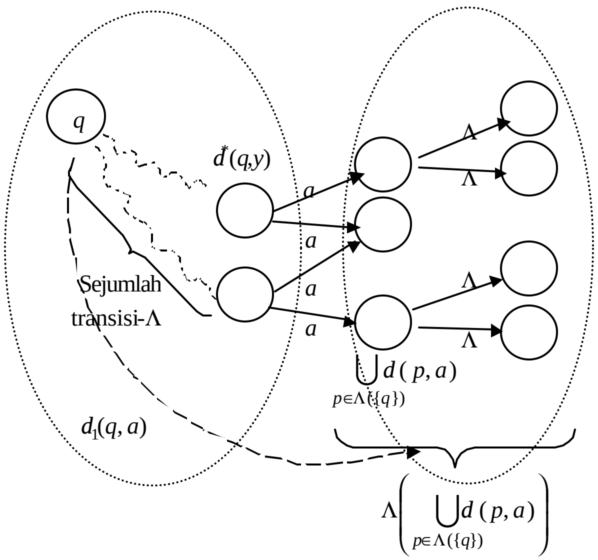
$$d_1(q, a) = d^*(q, a)$$

Menurut definisi d^* pada NFA -Λ diperoleh sbb.

$$d^*(q, a) \;= d^*(q, \Lambda a) = \Lambda\left(\bigcup_{p \in d^*(q,\Lambda)} d(p, a) \right) = \Lambda\left(\bigcup_{p \in \Lambda(\{q\})} d(p, a) \right)$$

Operasi penggabungan di dalam tanda kurung besar menyatakan himpunan status yang tercapai dari q dengan simbol masukan a termasuk kemungkinan adanya sejumlah transisi-Λ sebelum transisi karena simbol a tsb.

Kemudian operasi closure-Λ terluar menghasilkan himpunan status yang dalam hal ini akan menjadi $d_1(q, a)$ di NFA M_1.



Pendefinisian d_1 seperti itu dimaksudkan agar $d_1^*(q, a)$ dari NFA M_1 adalah semua status yang dapat tercapai oleh NFA-Λ M dari status q berdasarkan masukan string tak-kosong x (termasuk setiap kemungkinan transisi-Λ). Dpl., untuk membuktikan bahwa

$d_1^*(q, x) = d^*(q, x)$

Secara induksi struktural matematis akan dibuktikan sbb. Untuk $x = a \in \Sigma$, maka $d_1^*(q, a) = d_1(q, a) = d^*(q, a) = d^*(q, x)$ Dengan hipotesis bahwa $d_1^*(q, x) = d^*(q, x)$, untuk $x = ya$, dengan $y \in \Sigma^*$, $|y| \geq 1$ dan $a \in \Sigma$, maka

$$\begin{aligned} d_1^*(q, ya) &= \bigcup_{p\in d_1^*(q,y)} d_1(p, ya) \\ &= \bigcup_{p\in d^*(q,y)} d_1(p, a) \\ &= \bigcup_{p\in d^*(q,y)} d^*(p, a) \end{aligned}$$

Karena setiap $d^*(p, a)$ adalah Λ-closure maka setelah digabungkan juga Λ-closure. Jadi Λ-closure dari himpunan yang dihasilkan ruas terakhir tersebut sama dengan himpunan itu sendiri, sbb.

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

Menentukan A_1 dari A

A_1 bisa berbeda dari A. Suatu status inisial q_0 yang bukan status menerima akan menjadi status menerima apabila di dalam $\Lambda(\{q_0\})$ terdapat status menerima. Dituliskan secara formal sbb.

$$A_1 = \begin{cases} A \cup \{q_0\} & \text{jika } \Lambda(\{q_0\}) \cap A \neq \emptyset \text{ dalam } M \\ A & \text{lainnya} \end{cases}$$

Teorema di atas menyatakan implikasi NFA-Λ dengan NFA. Berikut ini teorema yang lebih lengkap antara ketiga FA serta implikasi timbal balik pada ketiganya.

Teorema: Untuk suatu alfabet Σ, setiapa bahasa $L \subset \Sigma^*$, ketiga pernyataan berikut ini adalah ekivalen (jika salah satu benar maka kedua lainnya juga benar):

1. L dapat dikenali oleh suatu FA.
2. L dapat dikenali oleh suatu NFA.
3. L dapat dikenali oleh suatu NFA-Λ.

Bukti : Teorema-teorema sebelumnya menyatakan bahwa pernyataan ketiga berimplikasi pernyataan kedua dan pernyataan kedua berimplikasi pernyataan pertama. Jadi untuk membuktikan teorema di atas cukup dengan membuktikan bahwa pernyataan pertama berimplikasi pernyataan ketiga.

Jika suatu bahasa L diterima oleh FA $M = \{Q, \Sigma, q_0, A, d\}$. Suatu NFA-Λ $M_1 = \{Q, \Sigma, q_0, A, d_1\}$ dapat dibentuk untuk menerima L sebagai berikut.

d_1: $Q \times (\Sigma \cup \{\Lambda\}) \rightarrow 2^Q$

didefinisikan dengan formula (untuk setiap $q \in Q$, dan setiap $a \in \Sigma$).

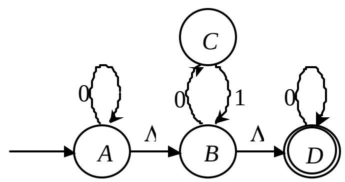
$d_1(q, \Lambda) = \varnothing$

$d_1(q, a) = \{d(q,a)\}$

Formula pertama menyatakan dalam M_1 tidak ada transisi-Λ, dan yang kedua menyatakan fungsi transisi d_1 dan d adalah identik kecuali jika d memetakan ke suatu status, d_1 memetakan ke himpunan yang hanya berisi status tersebut. Jadi, dalam hal ini, NFA-Λ M_1 tidak ada nondeterminisme.

Berikut ini beberapa contoh mengeliminasi transisi-Λ dari NFA -Λ menjadi NFA, dan mengkonversinya menjadi FA.

Contoh: Disamping ini, *M* adalah NFA-Λ yang menerima bahasa $\{0\}^*\{01\}^*\{0\}^*$.



Fungsi transisi digambarkan dalam tabel transisi serta harga-harga $d^*(q, 0)$ dan $d^*(q, 1)$ yang dihitung dengan formula sbb.

$$d^*(A,0) = \Lambda\left(\bigcup_{p\in\Lambda(\{A\})} d(p,0)\right)$$

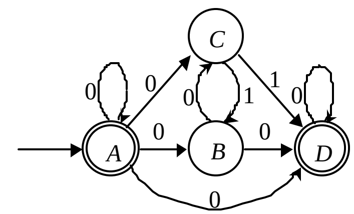
q	*d* (*q*, **L**)	*d* (*q*, 0)	*d* (*q*, 1)	d^*(*q*, 0)	d^* (*q*, 1)
A	{*B*}	{*A*}	∅	{*A*,*B*,*C*,*D*}	∅
B	{*D*}	{*C*}	∅	{*C*,*D*}	∅
C	∅	∅	{*B*}	∅	{*B*,*D*}
D	∅	{*D*}	∅	{*D*}	∅

Dari tabel tsb. Λ({*A*}) dihitung dengan mendapatkan *d* (*p*, 0) untuk setiap *p*, lalu menggabungkannya, kemudian mendapatkan Λ-closure dari hasilnya.

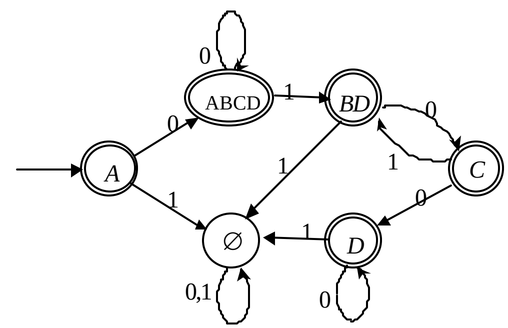
dari *A* dengan masukan 0, bisa tetap di *A* (string 0), atau transisi ke B (string 0Λ), atau ke C (string Λ0), atau ke D (string 0ΛΛ).

NFA M_1 diperoleh dengan fungsi transisi $d_1(p, a) = d^*(p, a)$. Dalam M_1 status *A* menjadi status menerima karena dalam *M* status *D* dapat dicapai dari *A* dengan duakali transisi-Λ.

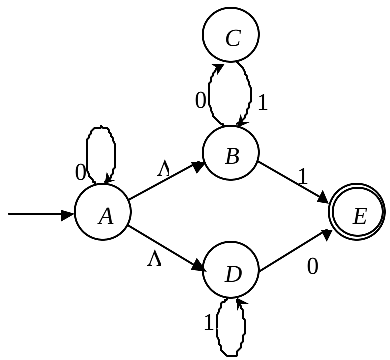
Dengan demikian NFA yang diperoleh adalah sebagai disamping ini.



Selanjutnya dengan konversi NFA ke FA yang telah dibahas pada topik NFA dari M_1 tsb.dapat diperoleh FA M_2 berikut ini.



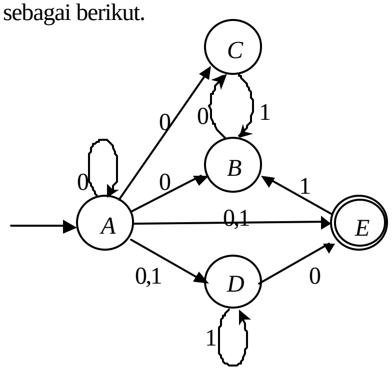
Contoh: NFA berikut ini mengenali bahasa $\{0\}^*(\{01\}^*\{1\} \cup \{1\}^*\{0\})$.



Tabel transisi berikut harga $d^*(q, 0)$ dan $d^*(q, 1)$ adalah.

q	*d*(*q*, L)	*d*(*q*, 0)	*d*(*q*, 1)	*d**(*q*, 0)	*d** (*q*, 1)
A	{*B*,*D*}	{*A*}	∅	{*A*,*B*,*C*,*D*,*E*}	{*D*,*E*}
B	∅	{*C*}	{*E*}	{*C* }	{*E*}
C	∅	∅	{*B*}	∅	{*B*}
D	∅	{*E*}	{*D*}	{*E*}	{*D*}
E	∅	∅	∅	∅	∅

NFA yang diperoleh adalah sebagai berikut.



Dari NFA di atas kemudian diperoleh FA sebagai berikut ini.