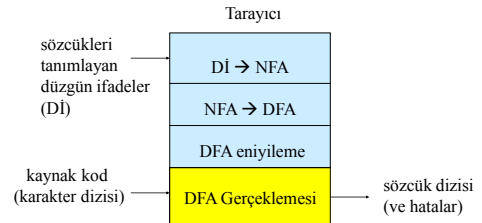


Sözcük Çözümleyici (2)

1

Tarayıcı Nasıl Geliştirilir?

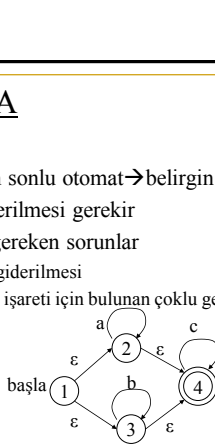


2

NFA → DFA

- belirgin olmayan sonlu otomat → belirgin sonlu otomat
- Belirsizliğin giderilmesi gerekir
- Çözümlemesi gereken sorunlar
 - ϵ geçişlerinin giderilmesi
 - belirli bir giriş işareti için bulunan çoklu geçişlerin giderilmesi

$(a^* | b^*) c^*$



3

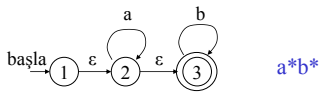
NFA → DFA (2)

- ϵ geçişlerinin giderilmesi
 - Fonksiyon: $\epsilon_closure(s)$
 - s durumunu giriş parametresi olarak alır ve geriye s durumundan “ ϵ ” geçişleriyle ulaşılabilen tüm durumları içeren bir durumlar kümesi getirir
- s durumunu sonuç kümesine ekle ϵ
 - t durumu sonuç kümesinde ise ve $t \rightarrow u$ ise, u durumunu da sonuç kümesine ekle
 - Yukarıdaki iki adımı, sonuç kümesine yeni durum eklenemez olana kadar tekrarla

4

Fonksiyon: $\epsilon_closure(s)$

örnek



$\epsilon_closure(1) = \{1, 2, 3\}$
 $\epsilon_closure(2) = \{2, 3\}$

5

NFA \rightarrow DFA Geçiş Algoritması

Kabul:

- s_0 NFA'nın başlangıç durumu ise, $d_0 = \epsilon_closure(s_0)$ DFA'nın başlangıç durumudur
- D oluşturulacak olan DFA'nın durumlar kümesidir ve başlangıçta $D = \{d_0\}$ ve d_0 işaretlenmemiştir
- Her d_i , NFA durumlarından (s_i) oluşan bir durumlar kümesidir

6

NFA \rightarrow DFA Geçiş Algoritması(2)

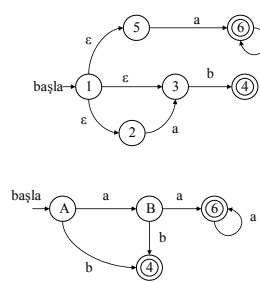
```

while ( D'nin işaretlenmemiş bir "di" durumu var) do
{
  di 'i' işaretle;
  for ( her "a" giriş işareti) do
  {
    T  $\leftarrow$  di'nin içerdiği her sj durumundan a işaretiyle
    geçişin mümkün olduğu durumlar kümesi;
    x =  $\epsilon\_closure(T)$ ;
    if ( x D'nin elemanı değil)
    {
      x 'i' D kümesine ekle;
      x'in işaretli olmadığını göster;
      eğer yoksa, di  $\rightarrow$  x "a" geçişini oluştur;
    }
  }
}

```

7

NFA \rightarrow DFA –Örnek1



D'nin ilk durumu:

- $\epsilon_closure(1) = \{1, 2, 3, 5\} = A$

A'dan "a" ve "b" üzerinden geçişleri incele – A işaretli

- geçiş (A, a) = {3, 6}
- $\epsilon_closure(3, 6) = \{3, 6\} = B$
- A \rightarrow B "a" kenarlı bir geçiş ekle
- geçiş (A, b) = {4}, $\epsilon_closure(4) = \{4\}$
- A \rightarrow 4 "b" kenarlı bir geçiş ekle

B'den "a" ve "b" üzerinden geçişleri incele – B işaretli

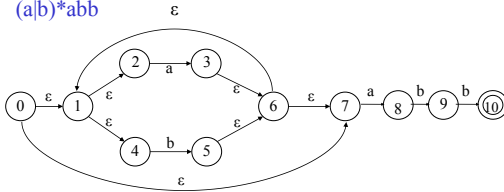
- geçiş (B, a) = {6}, $\epsilon_closure(6) = \{6\}$
- B \rightarrow 6 "a" kenarlı bir geçiş ekle
- geçiş (B, b) = {4}, $\epsilon_closure(4) = \{4\}$
- B \rightarrow 4 "b" kenarlı bir geçiş ekle

- Durum 6 ve durum 4 için aynı adımları tekrarla

8

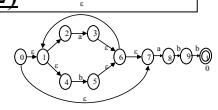
NFA → DFA – Örnek2

$(a|b)^*abb$



9

NFA → DFA – Örnek2 (2)



■ D'nin ilk durumu:

• $\epsilon\text{-closure}(0) = \{0, 1, 2, 4, 7\} = A$

■ A'dan "a" ve "b" üzerinden geçişleri incele – A işaretletle

• geçiş $(A, a) = \{3, 8\}$, $\epsilon\text{-closure}(3, 8) = \{1, 2, 3, 4, 6, 7, 8\} = B$

• $A \rightarrow B$ "a" kenarlı bir geçiş ekle

• geçiş $(A, b) = \{5\}$, $\epsilon\text{-closure}(5) = \{1, 2, 4, 5, 6, 7\} = C$

• $A \rightarrow C$ "b" kenarlı bir geçiş ekle

■ B'den "a" ve "b" üzerinden geçişleri incele – B işaretletle

• geçiş $(B, a) = \{3, 8\}$, $\epsilon\text{-closure}(3, 8) = \{1, 2, 3, 4, 6, 7, 8\} = B$

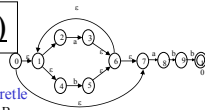
• $B \rightarrow B$ "a" kenarlı bir geçiş ekle

• geçiş $(B, b) = \{5, 9\}$, $\epsilon\text{-closure}(5, 9) = \{1, 2, 4, 5, 6, 7, 9\} = D$

• $B \rightarrow D$ "b" kenarlı bir geçiş ekle

10

NFA → DFA – Örnek2 (3)



■ C'den "a" ve "b" üzerinden geçişleri incele – C işaretletle

• geçiş $(C, a) = \{3, 8\}$, $\epsilon\text{-closure}(3, 8) = \{1, 2, 3, 4, 6, 7, 8\} = B$

• $C \rightarrow B$ "a" kenarlı bir geçiş ekle

• geçiş $(C, b) = \{5\}$, $\epsilon\text{-closure}(5) = \{1, 2, 4, 5, 6, 7\} = C$

• $C \rightarrow C$ "b" kenarlı bir geçiş ekle

■ D'den "a" ve "b" üzerinden geçişleri incele – D işaretletle

• geçiş $(D, a) = \{3, 8\}$, $\epsilon\text{-closure}(3, 8) = \{1, 2, 3, 4, 6, 7, 8\} = B$

• $D \rightarrow B$ "a" kenarlı bir geçiş ekle

• geçiş $(D, b) = \{5, 10\}$, $\epsilon\text{-closure}(5, 10) = \{1, 2, 4, 5, 6, 7, 10\} = E$

• $D \rightarrow E$ "b" kenarlı bir geçiş ekle

■ E'den "a" ve "b" üzerinden geçişleri incele – E işaretletle

• geçiş $(E, a) = \{3, 8\}$, $\epsilon\text{-closure}(3, 8) = \{1, 2, 3, 4, 6, 7, 8\} = B$

• $E \rightarrow B$ "a" kenarlı bir geçiş ekle

• geçiş $(E, b) = \{5\}$, $\epsilon\text{-closure}(5) = \{1, 2, 4, 5, 6, 7\} = C$

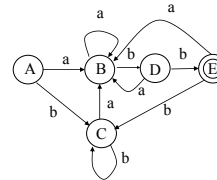
• $E \rightarrow C$ "b" kenarlı bir geçiş ekle

11

Oluşan DFA

$(a|b)^*abb$

DFA



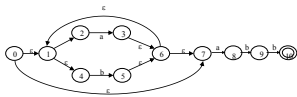
"E" sonlanma durumu

çünkü NFA'nın sonlanma durumu olan "durum 10" u içeriyor

12

DFA Eniyileme

- **DFA Eniyileme:** durum sayısını en aza indirme, (gereksiz veya birleştirilebilir durumlar bulunabilir)
- **Önemli Durum:** Bir DFA durumunu oluşturan NFA durumlarından “ε” çıkışlarının bulunmaması hali.



NFA'daki önemli durumlar = {2,4,7,8,9}

13

DFA Eniyileme

- İki DFA durumunun bire indirgenebilmesi için:
 - Her ikisinin de aynı önemli durumları içermesi
 - Her ikisinin de “kabul” durumunu içermesi/içermemesi gerekir

14

DFA Eniyileme (2)

- **Örnek:** $(a|b)^*abb$ için üretilen DFA'yı incelersek
 - NFA'daki önemli durumlar = {2,4,7,8,9}
 - DFA durumu $A = \{0, 1, 2, 4, 7\}$
 - DFA durumu $C = \{1, 2, 4, 5, 6, 7\}$
- **A ve C tek duruma indirgenebilir**, çünkü
 - - her ikisi de aynı önemli durumları içeriyor, ve
 - - her ikisi de kabul durumunu içermiyor
- Durum tablosunda “C” durumunu “A” ile yer değiştir

15

DFA Eniyileme (3)

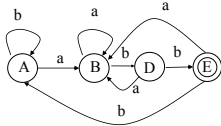
- DFA durum tablosu Eniyilenmiş DFA durum tablosu

durum	simge		durum	simge	
	a	b		a	b
A	B	C	A	B	A
B	B	D	B	B	D
C	B	C			
D	B	E	D	B	E
E	B	C	E	B	A

16

İndirgenmiş DFA

$(a|b)^*abb$

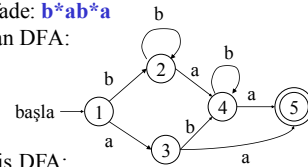


durum	a	b
A	B	A
B	B	D
D	B	E
E	B	A

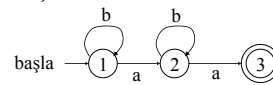
17

Örnek

- Düzgün ifade: b^*ab^*a
- Oluşturulan DFA:



- İndirgenmiş DFA:



18

DFA'nın Gerçeklenmesi

- DFA'nın bir C programına dönüştürülmesi
 - Sonlu otomati gerçekleyen bir durum tablosuna ve
 - Tablo, her durum ve giriş simgesi için geçişleri tanımlar
 - Tabloyu simüle eden bir program parçasına gerek duyulur

```

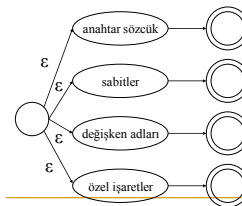
gecis_tablo[durum_sayisi][giris_sayisi]; // sonlu otomat tablosu
kabal_durumu[durum_sayisi]; // durumun kabul durumu olup olmadığını gösteren lojik dizi
durum = ilk_durum;
while (durum != HATA) { // yeni bir duruma geçiş mümkün olduğu sürece
    c = input.read(); // kaynaktan bir karakter oku-giriş bilgisi
    if (c == EOF) break;
    durum = gecis_tablo[durum][c]; // giriş bilgisi ile bir sonraki duruma geç
}
return kabal_durumu[durum]; // ulaşılan son durum kabul durumu ise "TRUE" döndürür

```

19

Genel Amaçlı Bir Tarayıcının Tasarlanması

- Dilin kabul ettiği sözcükleri tanımlayan her düzgün ifade için bir NFA üret (düzgün ifadeler tanım sıralarına göre önceliklidirler)
- NFA'ların tümünü bir başlangıç durumu altında birleştir
- Yeni NFA'dan bir DFA üret



20

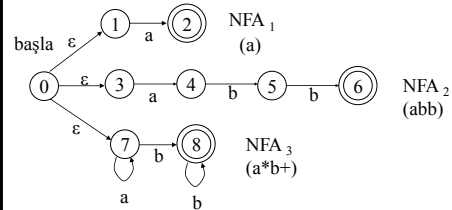
Genel Amaçlı Bir Tarayıcının Tasarlanması

- Birleştirilmiş NFA içinde birden çok kabul durumu bulunacaktır.
- **DFA'nın bir kabul durumu birden fazla NFA kabul durumu içerebilir.**
- **AMAÇ: uygun olan en uzun katar ile eşleştirme**
- Tarayıcının çalışma yöntemi:
 - Bir kabul durumu ile karşılaşıncı, çalışmayı sonlandırmayıp bir sonraki durum tanımlı olmayana kadar işlemi sürdür
 - Daha sonra geri dönüp, kabul durumu içeren ilk DFA durumunu geri getir

21

Tarayıcı Tasarımı İçin Örnek

- Düzgün ifadeler: a , abb , a^*b^+



22

Tarayıcı Tasarımı İçin Örnek (2)

- NFA → DFA: Geçiş Tablosu - {2,6,8} kabul durumları

durum	a	b	sözcük
A={0,1,3,7}	{2,4,7}	{8}	---
B={2,4,7}	{7}	{5,8}	a
C={8}	-	{8}	a^*b^+
D={7}	{7}	{8}	---
E={5,8}	-	{6,8}	a^*b^+
F={6,8}	-	{8}	abb

- DFA'nın sonlanma durumları: {B, C, E, F}
 - F iki kabul durumu içeriyor, tanım sırası dikkate alınarak 6 (abb) seçilir

23

DFA Üzerinde Örnek Uygulama

