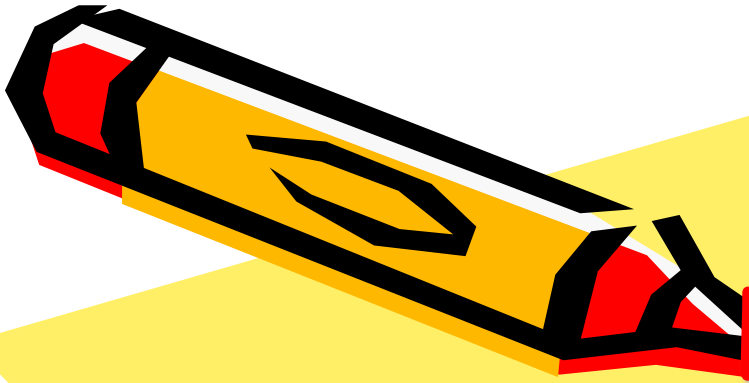


KARAR VERİLEBİLİRLİK (DECIDABILITY)

Otomata Teorisi

Konya Teknik Üniversitesi
Bilgisayar Mühendisliği Bölümü

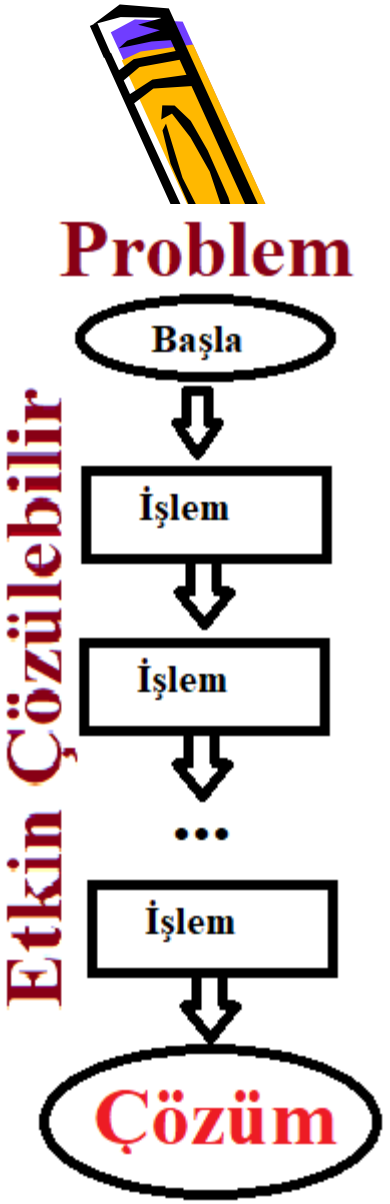
Karar Verilebilirlik

- Sonlu Otomata teorisi kapsamında
  - Verilen iki düzgün ifadenin aynı dili tanımlayıp tanımlamadığına
  - Verilen iki sonlu otomata (FA) modelinin denk (eş) olup olmadığına
  - Verilen bir sonlu otomata (FA) tarafından tanınan dilin sonlu veya sonsuz sayıda kelime içerip içermediğine

  nasıl karar verebiliriz ?

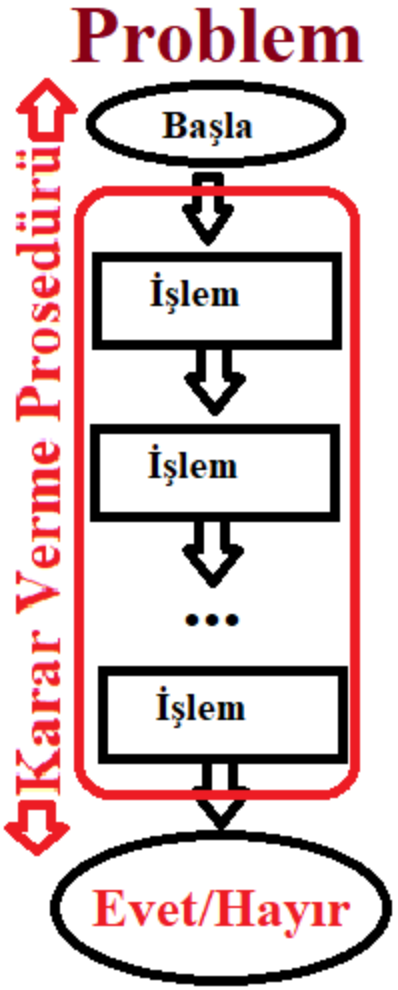
Tanım

- Etkin Çözülebilir (Effectively Solvable) Problem:
  - Matematiksel Mantıkta; Sonlu sayıda adım sonucunda cevabı elde edebileceğimiz bir algoritma varsa, söz konusu problem etkin çözülebilir olarak adlandırılmaktadır.
    - Örneğin, 2. derece denklem çözümü algoritması...



Tanım

- Karar Verme Prosedürü (Decision Procedure):
  - Evet yada Hayır cevabına sahip bir problemin etkin çözümü karar verme prosedürü olarak adlandırılmaktadır.
  - Karar verme prosedürüne sahip bir problem karar verilebilir (decidable) bir problem olarak adlandırılır.



Etkin Olmayan Bir Yöntem
(İki düzgün ifade aynı dil mi?)

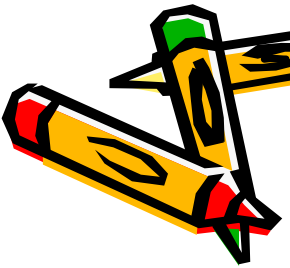
- İki düzgün ifadenin aynı dili tanımlayıp, tanımlamadığına karar vermek için aşağıdaki yöntem etkin bir çözüm müdür?

  Önerilen Yöntem:

  - Bir dile ait düzgün ifadeden sistemli bir şekilde en küçük uzunluktaki kelimelerden başlayarak diğer dilde olmayan bir kelime bulana kadar kelime türetmek ???

Etkin Olmayan Bir Yöntem

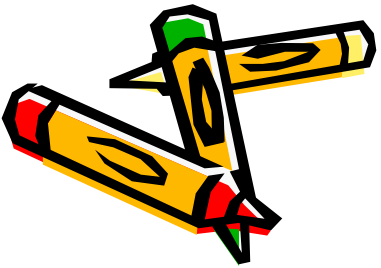
- Önceki slaytta verilen yöntem, bazı düzgün ifadeler için etkin bir çözüm gibi görülse de bazıları için ne kadar adım ve sürede karar verilebileceğinin bir garantisi yoktur.
  - a(a+b)* ile (b+Λ)(baa+ba*)* ?
  - (aa+ab+ba+bb)* ile ((ba+ab)*(aa+bb)*)* ?
  - ((b*a)*ab*)* ile Λ+a(a+b)*+(a+b)*aa(a+b)* ?

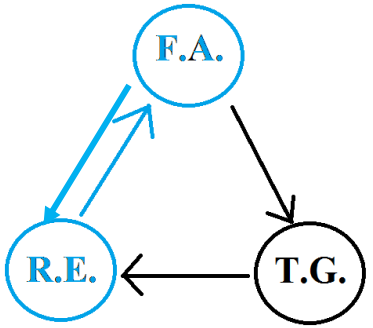


96 harf → 2 milyon civarı kelime

Yorumlar

- Eğer iki düzgün ifadenin denkliği için karar verme prosedürü bulunabilirse, Kleene teoremine göre FA modellerini etkin bir şekilde düzgün ifadelere çevirebilmek mümkün olduğu için, FA modellerinin de denkliği bulunmuş olacaktır. Bunun tersi de doğrudur.





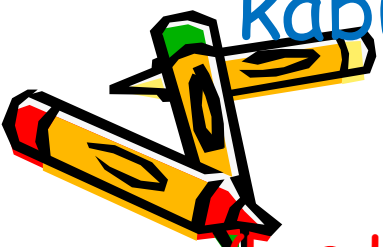
Karar Verme Algoritması (iki FA denk mi?)

- Ya düzenli ifade yada FA'larla tanımlanmış iki L_2 ve L_1 dili verilmiş olsun.
- L_1', L_2', $L_1 \cap L_2'$ ve $L_2 \cap L_1'$ dilleri için sonlu otomata üretmede gerekli olan prosedürleri Bölüm 9 (düzenli diller) da üretmiştik. Buna göre;
- Aşağıdaki dili tanıyan bir FA modeli oluşturulabilir:
  - $(L_1 \cap L_2') + (L_2 \cap L_1')$
  - Eğer L_1 ve L_2 aynı dil ise, bu FA hiçbir kelime kabul edemez. Buna NULL kelime de dahildir.
  - Verilen bir FA modelinin hiçbir kelime kabul etmediğini bulan bir karar verme prosedürü tanımlanmalıdır.

1.Başlangıçlar tek'e,
2.Finaller tek'e,
3.R.E. Formatına uyarla
4.By-Pass

Metot-1 (iki FA denk mi?)

- Önceki slaytta verilen formüle karşılık gelen dilin FA modelinden düzgün ifade elde etme adımlarında (Bakınız: Kleene Teoremi) sona gelirken başlangıç durumundan sonuç durumuna geçme olanağı kalmadığında, düzgün ifade elde edilememiş, dolayısı ile FA hiçbir kelime kabul etmiyor anlamı çıkmaktadır.



$(L_1 \cap L_2')$ + $(L_2 \cap L_1')$ İşleminden sonra yapılması gerektiğini unutmayınız

Metot-1 – Zıddı
F.A. En az bir kelime kabul ediyor mu?

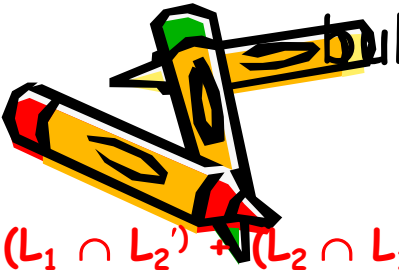
- İki dilin eşdeğerliğine karar verirken bir başlangıç çıkış noktası olarak şöyle bir adım da izlenebilir.
- Bu iki dilden birinin içerdiği kelime bulunmaya çalışılır. İnsan mantığı ile bu kolaydır. Ama bilgisayara bunu yaptırmak için şu adımlar izlenir.
  - FA önce düzenli ifadeye (RE) çevrilir. Her iki dil için her bir düzenli ifade aynı kelimeleri tanımlar.
  - (a+Λ) (ab*+ba*)* (Λ+b*)*
  - Tüm star (*) lar silinir.
  - (a+Λ) (ab+ba) (Λ+b)
  - + işaretinin kendisi ve işaretin sağ tarafındaki her bir ifade göz ardı edilir.
  - (a) (ab) (Λ)
  - Hiçbir * + işareti kalmadığı zaman parantezler kaldırılır ve semboller birleştirilir.
  - a ab Λ
  - Bu bir kelime biçimler.
  - aab
- Burada verilen adımlarla FA en az bir kelime kabul ediyor anlamı çıkartılmaktadır.

$(L_1 \cap L_2')$ + $(L_2 \cap L_1')$ ile FA elde etme İşleminden sonra yapılması gerektiğini unutmayınız

Binlerce state ve milyonlarca yönlendirilmiş kenardan oluşan geniş bir FA ile yüzleşilebilir!

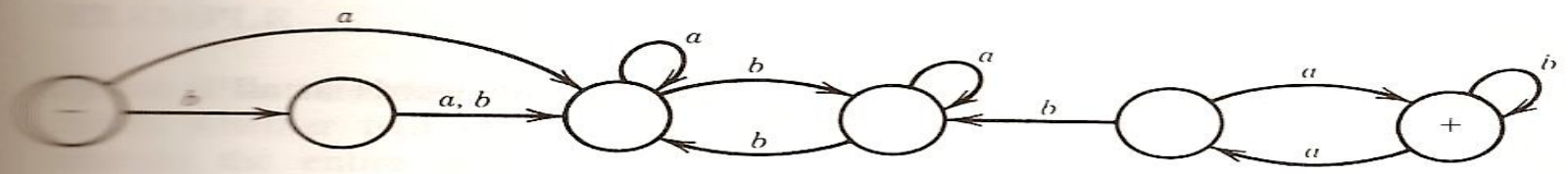
Metot-2 (iki FA denk mi?) Maviye Boyama

- Başlangıçtan sonuç durumuna giden bir yol bulma algoritması:
  - Başlangıç durumunu maviye boya
  - Her mavi durumdan çıkan kenarları izle, vardığı durumları maviye boya ve ilgili kenarları sil.
  - Yukarıdaki adımı yeni bir maviye boyanan durum kalmayana kadar tekrarla.
  - Prosedür bittiğinde, herhangi bir sonuç durumu maviye boyanmış ise FA modeli en az bir kelime kabul ediyor demektir. Bir sonraki slaytta örnek bulunmaktadır.

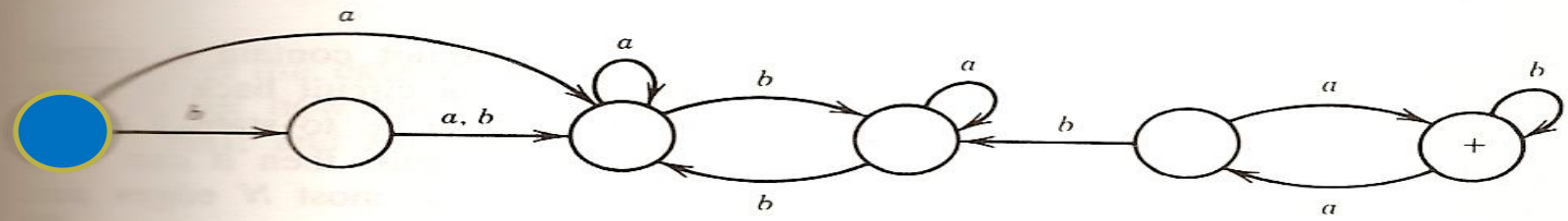


$(L_1 \cap L_2') + (L_2 \cap L_1')$ ile FA elde etme İşleminden sonra yapılması gerektiğini unutmayınız

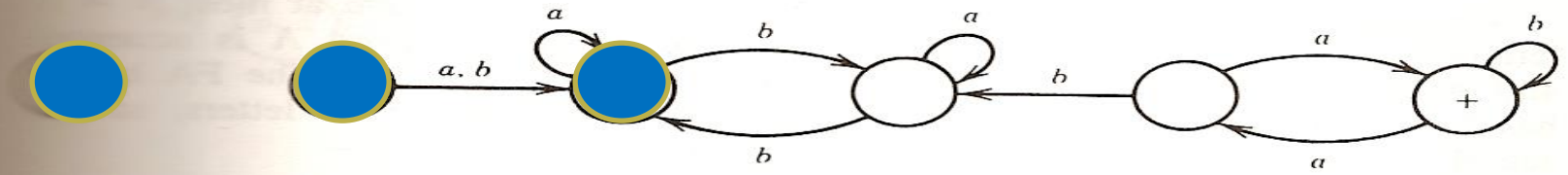
$(L_1 \cap L_2') + (L_2 \cap L_1')$ FA si,



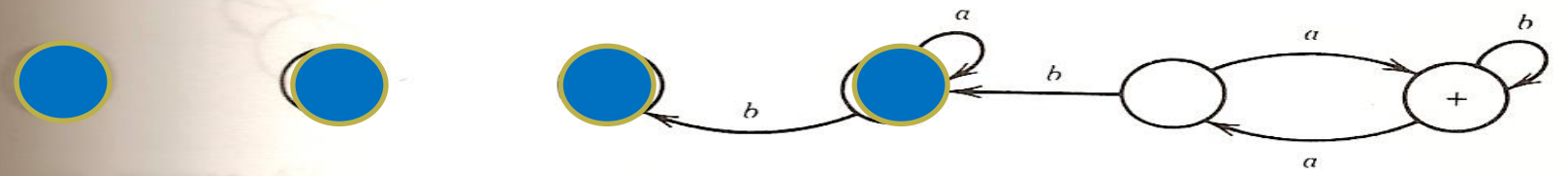
after Step 1:



after Step 2:



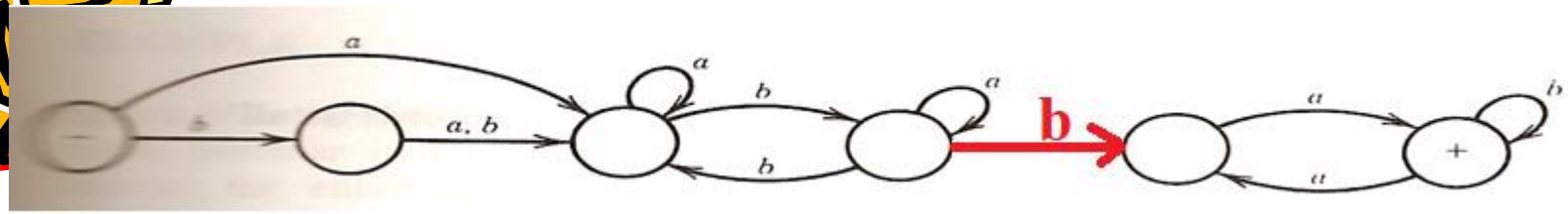
after Step 2 again:



after Step 2 again:

Metot-2
(iki FA denk mi?)
Maviye Boyama

- Metodu yorumlarsak;
  - Eğer N durumlu bir FA modeli varsa ve bu FA herhangi bir kelime kabul ediyor ise N veya daha az harfli kelime kabul edecektir.
    - Başlangıçtan sonuca giden en kısa yolda döngü (circuit/loop) olmaz.
    - En kısa uzunluktaki kelime en çok N-1 tane sembolden oluşabilir.



Metot-3
(İki FA denk mi?)

N durumlu bir FA modeli varsa;

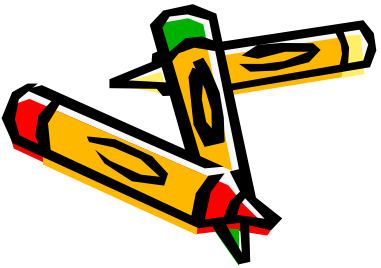
- N veya daha az uzunluktaki kelimelerin tümü denenir.
- Eğer FA modeli bu kelimelerden hiçbirini kabul etmiyor ise , başka hiçbir kelimeyi de kabul etmez.

N sayısından daha az uzunluğa sahip kelimelerin sayısı sonlu olduğu için çözüm karar verilebilir (decidable) bir çözümdür.

$(L_1 \cap L_2') + (L_2 \cap L_1')$ ile FA elde etme İşleminden sonra yapılması gerektiğini unutmayınız

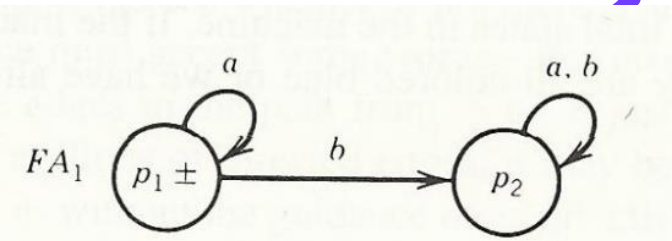
Örnek

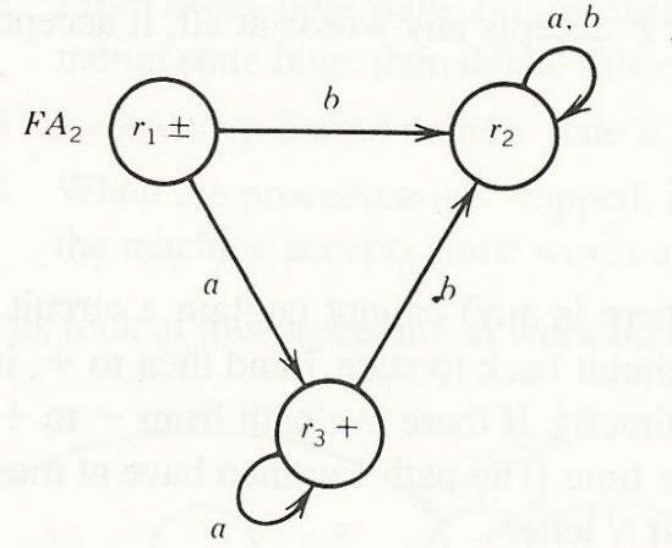
- Aşağıda 2 tane düzenli ifade veriliyor:
  - r_1: a^* r_2: $\Lambda + aa^*$
  - Yukarıda verilen düzgün ifadelerin aynı dili tanımlayıp tanımlamadığını karar verme prosedürü ile ispatlayınız.
    - Örnek sorunun daha kolay anlaşılabilmesi için, Kleene Teoremi ve düzgün dillerin kesişiminin bulunması konuları ile ilgili bilgiler hatırlanmalıdır. (Çözüm ders kitabından anlatılacaktır.)



Örnek devam

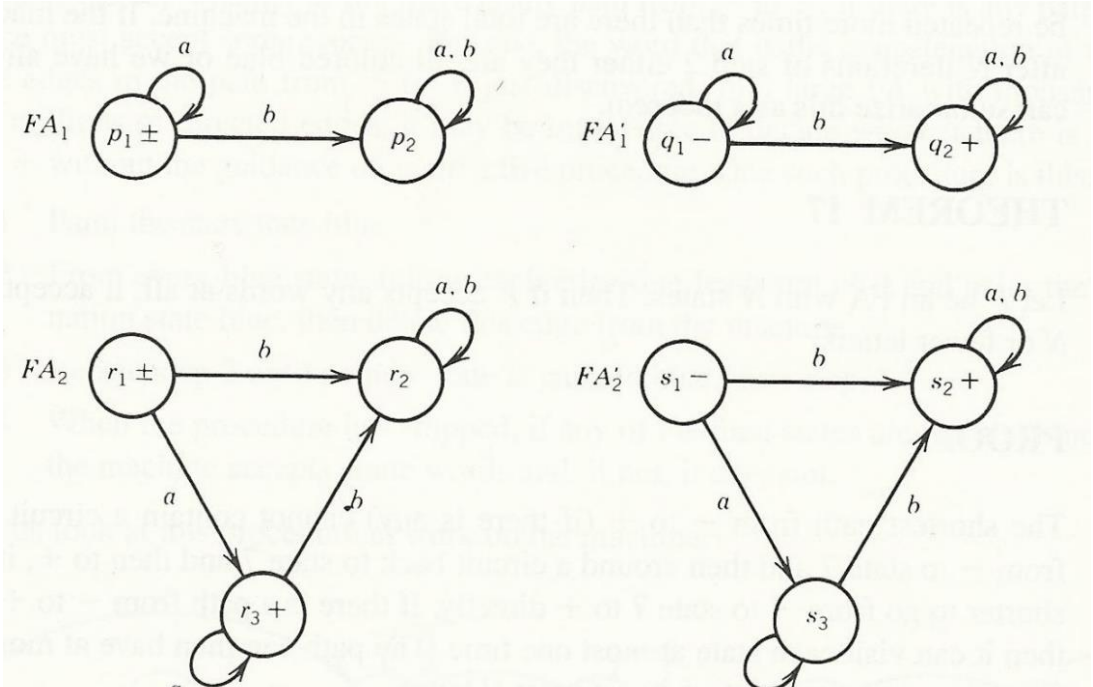
- **r_1: a* dili L_1, r_2: Λ + aa* dili L_2**
- Yukarıda verilen düzgün ifadelerin aynı dili tanımlayıp tanımlamadığını karar verme prosedürü adımları:
- L_1 dili için FA_1 ve L_2 dili için FA_2 çizilir.





Örnek devam

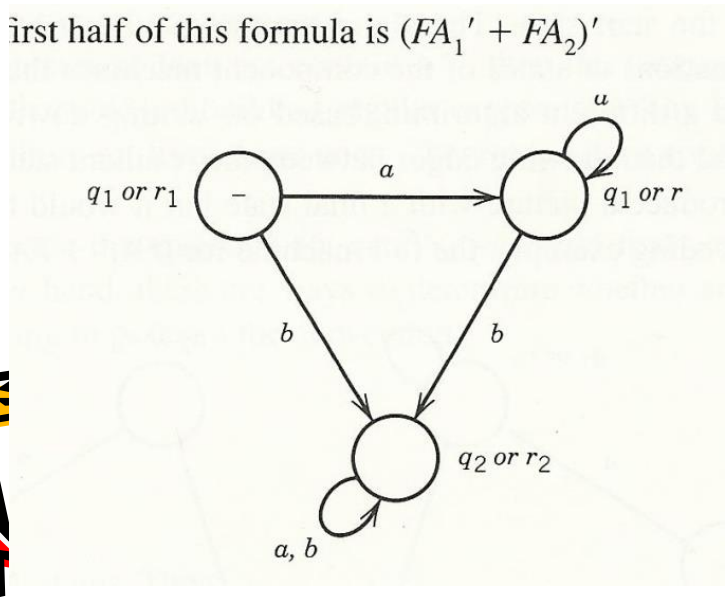
- **r_1: a* dili L_1, r_2: Λ + aa* dili L_2**
- Yukarıda verilen düzgün ifadelerin aynı dili tanımlayıp tanımlamadığını karar verme prosedürü adımları:
- L_1 dili için FA_1 ve L_2 dili için FA_2 çizilir.
- $(L_1 \cap L_2')+(L_2 \cap L_1')$ formülü yerine küme teorisine göre $(L_1'+L_2)'+(L_2'+L_1)'$ formülü kullanılır.
  - FA_1'e göre FA_1' ve FA_2'ye göre FA_2' otomatları çizilir.

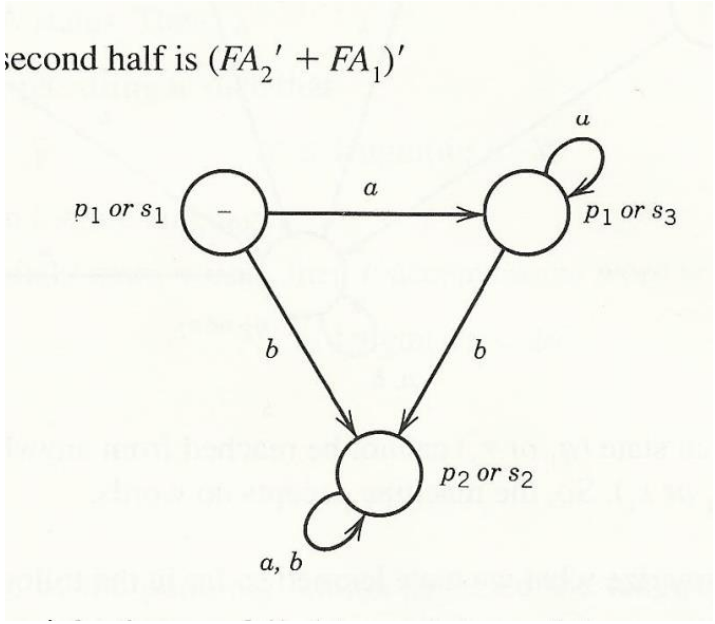


(FA_2'+ FA_1) otomatı final olmayan state içermediği için (FA_2'+ FA_1)' otomatı da final state'e sahip olmayacaktır.

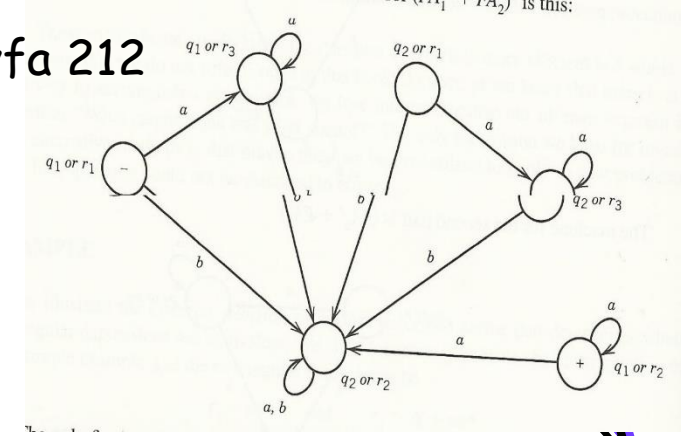
Örnek devam

- **r_1: a* dili L_1, r_2: Λ + aa* dili L_2**
- Yukarıda verilen düzgün ifadelerin aynı dili tanımlayıp tanımlamadığını karar verme prosedürü adımları:
- L_1 dili için FA_1 ve L_2 dili için FA_2 çizilir.
- $(L_1 \cap L_2')+(L_2 \cap L_1')$ formülü yerine küme teorisine göre $(L_1'+L_2)'+(L_2'+L_1)'$ formülü kullanılır.
  - FA_1'e göre FA_1' ve FA_2 'ye göre FA_2' otomatları çizilir.
  - (FA_1'+ FA_2)' ve (FA_2'+ FA_1)' otomatları Kleene teoremine göre çizilir.





Sayfa 212



Örnek devam

- **r_1: a* dili L_1, r_2: Λ + aa* dili L_2**
- Yukarıda verilen düzgün ifadelerin aynı dili tanımlayıp tanımlamadığını karar verme prosedürü adımları:
- L_1 dili için FA_1 ve L_2 dili için FA_2 çizilir.
- $(L_1 \cap L_2')+(L_2 \cap L_1')$ formülü yerine küme teorisine göre $(L_1'+L_2)'+(L_2'+L_1)'$ formülü kullanılır.
  - FA_1'e göre FA_1' ve FA_2 'ye göre FA_2' otomatları çizilir.
  - $(FA_1'+FA_2)'$ ve $(FA_2'+FA_1)'$ otomatları Kleene teoremine göre çizilir.
  - $(FA_1'+FA_2)$ otomatı final olmayan state içermediği için complementi olan $(FA_1'+FA_2)'$ otomatı final state'e sahip olmayacaktır.
  - $(FA_2'+FA_1)$ otomatı final olmayan state içermediği için $(FA_2'+FA_1)'$ otomatı da final state'e sahip olmayacaktır.
  - $(L_1'+L_2)'+(L_2'+L_1)'$ makinası üretildiğinde **9 state içerecektir ama state'lerin hiçbiri final state değildir**.
- Dolayısıyla $(L_1'+L_2)'+(L_2'+L_1)'$ dili hiçbir kelime içermez

Karar prosedürünün sonu

- L_1 dili ve L_2 dili eş değerdir. İki düzenli ifade aynı dili ifade etmektedir.

Bir Düzenli Dilin Sonlu veya Sonsuz Bir Dil Tanımladığına Karar Verme

R.E.

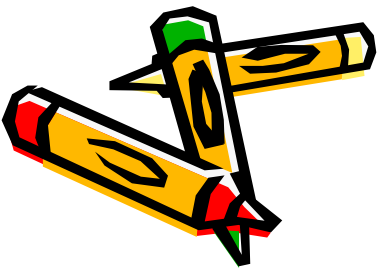
Düzgün ifade eğer * operatörü içeriyor ise sonsuz bir dil tanımlamaktadır.

- Λ^* istisnadır.

F.A.

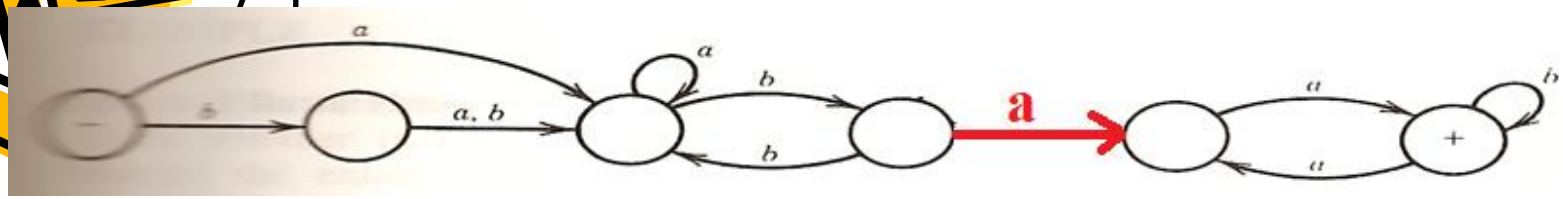
Eğer bir FA için bu soruya karar vermek istiyorsak ilk önce FA'yı düzenli ifadeye dönüştürmek faydalı olacaktır.

Diğer yandan dönüştürme işlemi yapmadan bir FA'nın sonlu bir dil kabul edip etmeyeceğini tanımlamak için yollar da vardır.



Bir FA'nın Sonlu veya Sonsuz Bir Dil Tanımladığına Karar Verme (1.teorem)

- N durumlu bir FA modeli olsun:
  - Eğer FA otomatı; N <= length(w) < 2N aralığındaki bir uzunlukta bir kelime kabul ederse, FA sonsuz bir dil kabul eder.
  - Eğer FA sonsuz bir dil kabul ediyor ise N <= length(w) < 2N aralığında uzunlukta bir kelime de kabul eder.
    - FA modelindeki döngüler düşünülerek ispat yapılabilecektir.

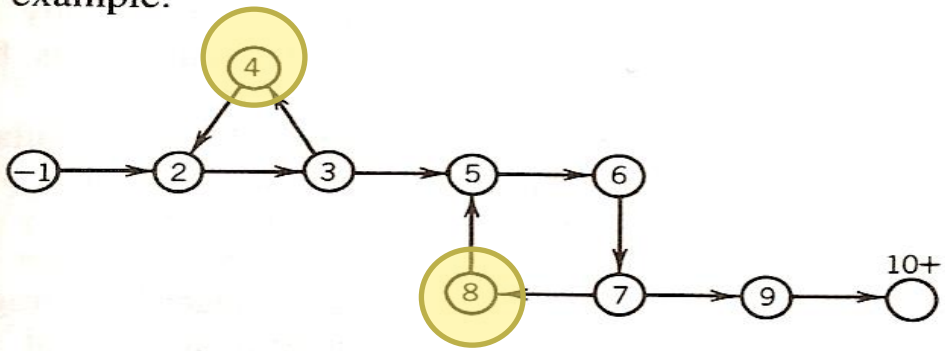


Başlangıçtan finale giden yolda DÖNGÜ varsa SONSUZdu

Bir FA'nın Sonlu veya Sonsuz Bir Dil Tanımladığına Karar Verme

EXAMPLE

Consider this example:



The first circuit is 2-3-4. It stays. The second circuit is 5-6-7-8. It is bypassed to become 5-6-7-9.

The path that used to be

1-2-3-4-2-3-5-6-7-8-5-6-7-8-5-6-7-9-10

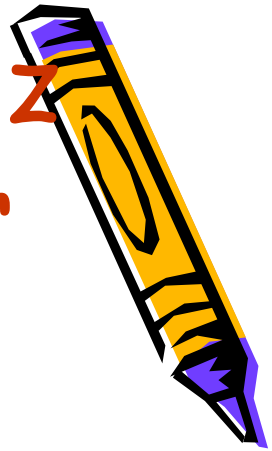
becomes

1-2-3-4-2-3-5-6-7-9-10

This demonstrates the existence of a simple one-circuit path in any FA that accepts infinitely many words. ■

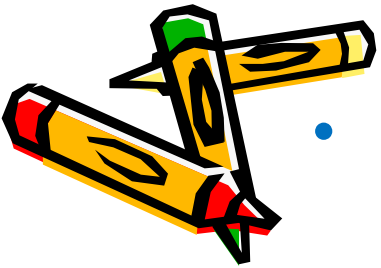
Sayfa 216'yı inceleyiniz.

Bir FA'nın Sonlu veya Sonsuz Bir Dil Tanımladığına Karar Verme (2.teorem)



N ve 2N arası uzunluktaki tüm kelimeler FA modeli üzerinde işletilir.

- Eğer bunlardan en az bir tanesi sonuç duruma ulaşıyorsa, FA modelinin tanımladığı dil sonsuzdur.
- Hiç biri kabul edilmezse dil sonlu olur.



Bir FA'nın Sonlu veya Sonsuz Bir Dil Tanımladığına Karar Verme

- N durumlu bir FA ve m harfli bir alfabe varsa, N ile 2N uzunluk aralığında

$$m^N + m^{N+1} + m^{N+2} + \ldots\ldots\ldots + m^{2N-1}$$

tane kelime vardır.

Buna göre çözüm karar verilebilirdir (decidable).

Bir FA'nın Sonlu veya Sonsuz Bir Dil Tanımladığına Karar Verme

Genelde FA'yı RE'ye çevirmek daha verimli olabilir. Ama niçin?

- Üç durum ve iki harfden oluşan bir alfabeye sahip bir makinayı düşünürsek test edilecek kelime sayısı

  $m^N + m^{N+1} + m^{N+2} + \ldots\ldots + m^{2N-1}$

  $2^3 + 2^4 + 2^5 = 56$ dır.

Bununla beraber üç durumlu bir FA 'yı sadece birkaç adımla bir RE 'ye dönüştürebiliriz.