Kararverilebilirlik

Karar prosedürü olan bir probleme karar verilebilir denir.

Karar Prosedürü → cevabı evet veya hayır olan bir probleme getirilen algoritma

İki RE tam olarak aynı dili belirleyebilir mi?

ÖRN $a(a+b)^*$ ve $(b+\Lambda)(baa+ba^*)^*$ | ikisinde de a ile başlar ikinci ifadenin kelimesi de b ile başlar

Fakat bu ifadelerin ortak kelimesi yoktur.

ÖRN $(aa+ab+ba+bb)^*$ ve $((ba+ab)^*(aa+bb^*))^*$

→ Her ikisi de $\Sigma=\{a,b\}$'de tanımlı çift sayıda harfe sahip tüm kelimelerin dilini tanımlar.

Equivalence (Eşdeğerlik)

İki RE'nin eşdeğer olup olmadığını belirlemek için bir karar prosedürümüz olsaydı, bunu iki FA'nın eşdeğer olup olmadığını belirlemek için kullanabilirdik.

→ FA'ları RE'lere dönüştürdük sonra karar verirdik.

Benzer şekilde, iki FA'nın eşdeğer olup olmadığını belirlemek için etkili bir prosedürümüz olsaydı, bunu RE'leri FA'lara dönüştürerek sonra aşağıdaki için kullanabilirdik.

RE'ler ve FA'lar tarafından tanımlanmış L_1 ve L_2 dilleri verildiğinde şu diller için FA üretmek için gerekli prosedür geliştirdik: L_1', L_2', $L_1 \cap L_2'$, $L_2 \cap L_1'$

$$(L_1 \cap L_2') + (L_2 \cap L_1')$$

⇒ Eğer L_1 ve L_2 aynı dillerse bu makine hiçbir kelimeyi kabul edemez

⇒ Bu makine tek bir kelimeyi bile kabul ederse o zaman L_1, L_2'ye eşit değildir, bu bir kelime olsa bile (bu Λ bile olsa)

Bir FA'nın herhangi bir kelimeyi kabul edip etmediği nasıl belirlenir?

① FA'yı RE'ye dönüştürün. Her RE bazı kelimeleri tanımlar. Önce tüm yıldızları silin. Sonra her + için, toplamın sağ tarafını ve +'yı atın. Hiç * veya + kalmadığında parantezleri kaldırın. Artık a'lar, b'ler ve null'ın bir birleşimine sahibiz.

ÖRN $(a+\Lambda)(ab^*+ba^*)^*(\Lambda+b^*)^* \Rightarrow (a+\Lambda)(ab+ba)(\Lambda+b)$

$(a)(ab)(\Lambda)$

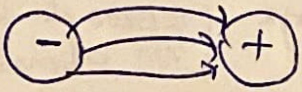
aab

Eğer RE en az bir kelime tanımlıyorsa, ilk bakışta her FA'nın en az bir kelimeyi kabul etmesi gerekir gibi görünüyor.

Bu mantıktaki boşluk bu FA'yı RE'ye dönüştürme sürecinin bozulmasıdır.

Bir FA'yı RE'ye dönüştürürken son adımda bu noktaya geliriz.

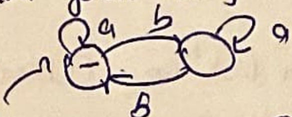
-'den +'ya uzanan kenarlar olmalıdır.



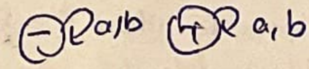
Ancak son adımda -'den +'ya giden hiçbir yolun olmadığını fark edebiliriz.

Bu durum 3 farklı şekilde olabilir:

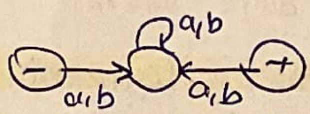
① Makinenin final stateti yoktur



② Final state, start stateten kopuktur. →



③ Final state start stateten erişilemiyordur

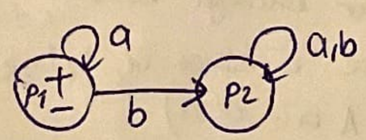


② FA'yı inceleyerek -'den +'ya giden bir yol olup olmadığını kontrol edin. Herhangi bir yol varsa, makine bazı kelimeleri kabul etmelidir.

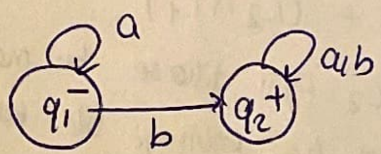
③ N makinedeki state sayısı o.ü. N'den az uzunluktaki tüm kelimeleri FA üzerinde çalıştırarak test edin. Eğer FA hiçbirini kabul etmezse, o zaman hiçbir kelime kabul etmez.

ÖRN $r_1 = a^*$, $r_2 = \Lambda + aa^*$ Bu ikisinin aynı dili tanımladığını ispatlayalım.

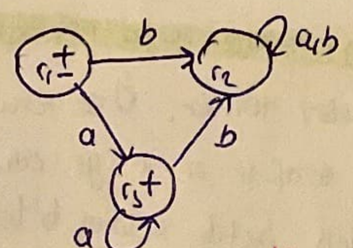
FA_1



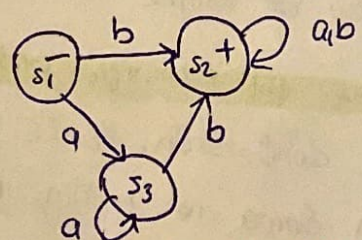
FA_1'



FA_2

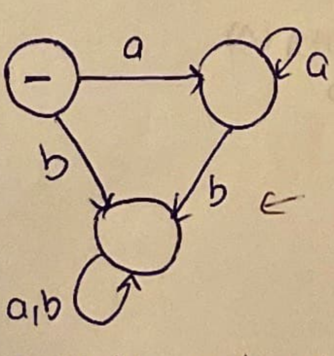


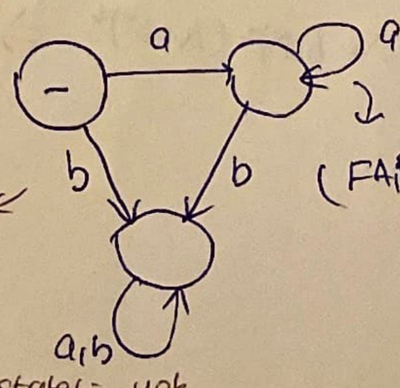
FA_2'



→ $(L_1 \cap L_2') + (L_2 \cap L_1')$ mantıksal formülünü kullanmak yerine makinemizi eşdeğer küme teorisi formülünü → $(L_1' + L_2)' + (L_2' + L_1)'$ kullanarak oluşturuyoruz.

$(FA_1' + FA_2)'$





$(FA_1' + FA_2')'$

⇒ Hiçbir makinenin final stateti yok.

⇒ İkisinden biri bir kelimeyi kabul ederse $L_1 \neq L_2$

→ RE * içeriyorsa o zaman dil sonsuzdur.
Tek istisna Λ^*'dır o da Λ eşittir

Örn

$(\Lambda + a\Lambda^*)(\Lambda^* + \Lambda)^*$ ve $(\Lambda + a\Lambda)^* (\Lambda^* + \Lambda)^*$

→ Sadece ikincisi sonsuz bir dil tanımlar.

RE bir * içermiyorsa dil mecburen sonludur. Çünkü bir RE düşünmenin diğer kuralları sonlu bir kümeden sonsuz bir kümeye ulaşamaz.

Bu soruyu bir FA için çözmek istiyorsak önce onu bir RE'ye dönüştürmeliyiz.