Biçimsel Diller ve Otomata Teorisi

Dr. Öğr. Üyesi Hayri Volkan Agun

Bilgisayar Mühendisliği Bölümü

Bursa Teknik Üniversitesi

Kaynaklar

Ders Kitabı

- An Introduction to Formal Languages and Autotamata, Peter Linz, 6th Edition, 2017.
- An Introduction to Computer Theory, Daniel Isaac Aryeh Cohen, 2nd Edition, 1996.

İçerik

- %100 Teorik
- Klasik sınav
- Vize %40, Final %60

Sadeleştirme

Bağlam Bağımsız Gramerler ve Sadeleştirme

- Bağımsız dil de belirsizlik sorunları ve ayrıştırma konularını işledik. Her zaman ayrıştırma için tüm olası adımların denenmesi mümkün olsa da, bu çok verimsizdir ve bu nedenle pratik değildir.
- Derleyiciler gibi gerçek uygulamalar için daha verimli ayrıştırma yöntemlere ihtiyacımız vardır. Bu, bağımsız dil üretiminin sağ tarafının sınırsız biçimi tarafından zorlaştırılmaktadır, bu nedenle sağ tarafı kısıtlayıp gramerin gücünü azaltmadan yapılabilir mi?
- Bu haftaki konumuzda bu ayrıştırma işleminde norm kavramını göreceğiz. Norma uygun bir gramer hem güçlü olacak hem de ayrıştırmada kolaylık sağlayacaktır.
- Eğer bir bağımsız dil gramerinde sağ tarafta λ bulunan üretimler bulunuyorsa, böyle λ-üretimleri içermeyen eşdeğer bir gramer bulabiliriz. Benzer şekilde, yalnızca sağ tarafta tek bir değişkeni olan birim üretimleri ve hiçbir zaman bir dize türetiminde meydana gelmeyen gereksiz üretimleri kaldırabiliriz.

Gramer Dönüşümleri

- Boş dize birçok teorem ve kanıtlamada oldukça özgün bir rol oynar ve genellikle ona özel önem vermek gereklidir.
- λ içermeyen dilleri gördüğümüz üzere genellikten ödün vermiyoruz. Herhangi bir bağımsız dil L olsun ve G = (V, T, S, P), L - {λ} için bağımsız bir dil grameri olsun. Sonra, S0 olarak yeni bir değişkeni V'ye ekleyerek, S0'u başlangıç değişkeni haline getirerek ve P'ye üretimleri ekleyerek elde ettiğimiz gramer:

$$S_0 \rightarrow S|\lambda$$

- Bu durumda herhangi bir bağımsız dil grameri G verildiğinde, L(GL) = L(G) - {λ} olan GL'yi elde etme yöntemi vardır.
- Benzer bir şekilde λ olsun yada olmasın iki bağlam bağımsız gramer arasında bir fark yoktur.

Yer değiştirme kuralı

- G = (V, T,S,P) bağlam bağımsız grameri için $A \to x_1 B x_2.$ A ve B nin biribirinden farklı olduğu bir kural tanımı olsun. B nin açılımı : $B \to y_1 \,|y_2| \cdots |y_n$ olsun.
- G^2 = (V, T,S,P) bağlam bağımsız gramerinde B açılımını kaldıralım. Bu açılımın kaldırılması durumudan A nın tanımını daha da genişletebiliriz ki kural bozulmasın.
- Aşağıda yeni A açlımı verilmiştir. G gramerinde A açlımınıda ara forma dönüştürerek açarsak aşağıdaki ifade elde edilir.

$$A \to x_1 B x_2$$

$$A \to x_1 y_1 x_2 \,|x_1 y_2 x_2| \cdots |x_1 y_n x_2.$$

- Bu durumda L(G) = L(G^2).

Örnek

- G = ({A,B} , {a, b, c}, A, P) grameri için kurallar aşağıdaki gibidir.

$$A \rightarrow a\,|aaA|\,abBc,$$
$$B \rightarrow abbA|b.$$

- B kuralını açarak A'da uygularsak yeni kurallar aşağıdaki gibi olacaktır.

$$A \rightarrow a\,|aaA|\,ababbAc|abbc,$$
$$B \rightarrow abbA|b.$$

- Örneğin G için aşadaki açılımı uygularsak bu yukarıda A açılımı ile tanımlanmış olacaktır.

$$A \Rightarrow aaA \Rightarrow aaabBc \Rightarrow aaabbc$$

- Bu aynı zamanda B kuralının olmadığı bir G^2 grameri ile de uyumludur.

Gereksiz Kurallar

- Örneğin aşağıdaki gramer kuralında A açılımı tamamen gereksizdir.

$$S \rightarrow aSb|\lambda|A,$$
$$A \rightarrow aA,$$

- Bu açılımda A hiçbir zaman sonlanmamaktadır. Kısaca A terminal bir sembol barındırmaz ve iç içe döngü oluşturur.
- Tanım gereği eğer bir kural örneğin A, eğer L dilinde herhangi bir w terminal sembolüne sahip ise bu kural gereklidir.

$$S \stackrel{*}{\Rightarrow} xAy \stackrel{*}{\Rightarrow} w,$$

Gereksiz Kurallar

- Eğer bir kural başlangıçtan ulaşılabilir değil ise gereksizdir. Örneğin aşağıda S başlangıç ise,

$$S \rightarrow A,$$
$$A \rightarrow aA|\lambda,$$
$$B \rightarrow bA,$$

- Burada B kuralı bir terminal sembol barındırsa da hiçbir zaman türetilmeyeceğinden B gereksiz bir kuraldır.

Örnek

- Tanımlı G(V, T, S, P), S={S, A, B, C} ve T={a, b} gramerinden gereksiz kural ve sembolleri çıkarınız.

$$
\begin{aligned}
S &\rightarrow aS\,|A|\,C,\\
A &\rightarrow a,\\
B &\rightarrow aa,\\
C &\rightarrow aCb.
\end{aligned}
$$

- Burada öncelikle hiçbir zaman sonlu terminal bir sembole erişemeyen kurallara bakalım. Görülceği üzere C kuralı tekrar ederek hiçbir zaman sonlu bir duruma ulaşamıyor. O zaman C çıkarılabilir.
- İkinci olarak başlangıçtan ulaşılamayan kurallara bakalım burada S'den B hiçbir zaman ulaşılamıyor. O zaman B gereksizdir.
- Son olarak yeni kurallarda hiç geçmeyen b sembolü de gereksiz olacaktır.

Örnek

- Yeni gramer kuralları aşağıdaki gibi tanımlanacaktır.

$$S \rightarrow aS|A,$$
$$A \rightarrow a.$$

- Kurallar ve alfabe ise aşağıdaki gibi olacaktır.

$$\hat{V} = \{S, A\}, \hat{T} = \{a\},$$

Boş Sembolün Kaldırılması (λ)

- Kurallar içerisinde isteyenmeyen boş karakter katarı (λ) gramerden kaldırıldığında gramerin ifade gücünde hiçbir değişiklik olmaz.
- Herhangi bir bağlam bağımsız gramer için $A \rightarrow \lambda$ ifadesine λ-açılımı denir. Herhangi bir kural açılımı ile erişilemeyen açılım $A \overset{*}{\Rightarrow} \lambda$ ifdesi ile gösterilir ve geçersiz kılınabilen açılım olarak adlandırılır.

Boş Sembolün Kaldırılması (λ)

- Aşağıdaki gramer için

$$S \rightarrow aS_1b,$$
$$S_1 \rightarrow aS_1b|\lambda,$$

- λ açılımını kaldırmak istersek yeni kurallar eklememiz gerekecektir. Sonuçta S_1 kuralı sonlanabilir olmalıdır. Bunun için hem S_1 hem de S kuralı değiştirilmelidir. Çünkü S kuralı S_1 i barındırmaktadır. Boş kural yerine direk erişilebilen son terminal sembolü yazabiliriz. O zaman aşağıdaki kurallar elde edilir.

$$S \rightarrow aS_1b|ab,$$
$$S_1 \rightarrow aS_1b|ab.$$

Örnek

- Aşağıda tanımlı gramer içinde λ – türetimlerini barındırmayan yeni eşlenik bir gramer oluşturunuz.

$$
\begin{aligned}
S &\rightarrow ABaC,\\
A &\rightarrow BC,\\
B &\rightarrow b|\lambda,\\
C &\rightarrow D|\lambda,\\
D &\rightarrow d.
\end{aligned}
$$

Örnek

- Yandaki kuralları inceleyecek olursak B ve C kurallarında geçersiz kılınabilir açılım mevcuttur. Bu kuralların sonlu durumlarını hesaplayarak kullanıldığı kurallara ekleyerek yeni kuralları oluşturabiliriz.
- Bu durumda B yerine ya hiçbir şey yada b ve C yerinde ya hiçbirşey yada d yazabiliriz.
- Bu durumda S'deki A da geçersiz kılınabilir. Çünkü A yerine hiçbirşey yazabiliriz.

$$
\begin{aligned}
S &\rightarrow ABaC,\\
A &\rightarrow BC,\\
B &\rightarrow b|\lambda,\\
C &\rightarrow D|\lambda,\\
D &\rightarrow d.
\end{aligned}
$$

$$
\begin{aligned}
S &\rightarrow ABaC\,|BaC|\,AaC\,|ABa|\,aC\,|Aa|\,Ba|a,\\
A &\rightarrow B\,|C|\,BC,\\
B &\rightarrow b,\\
C &\rightarrow D,\\
D &\rightarrow d.
\end{aligned}
$$

Birim üretimleri

- Birim değişimleri tekrar eden değişimlerdir. Yeni bir bilgi barındırmayan kurallardır. Örneğin aşağıdaki değişimler birim üretimdir.
- $A \to B,$ veya $\begin{array}{l} A \to B, \\ B \to A, \end{array}$
- Görüldüğü gibi birim değişimler A yerine B nin sonlu durumlarını barındıran açılımı yazarak kolayca kaldırılabilirler. Örneğin B nin tüm sonlu durum açılımlarını barındıran kural aşağıdaki gibi ise

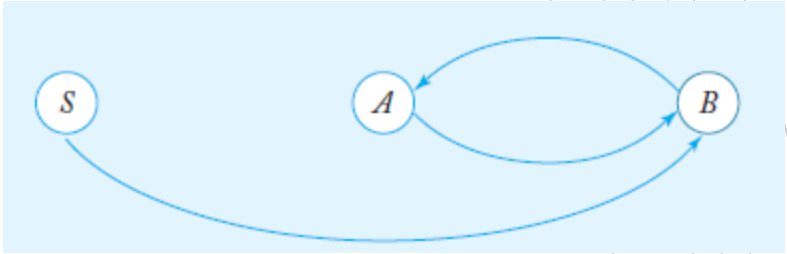
$$B \to y_1 \,|y_2| \cdots y_n$$

- Ancak o zaman $A \to y_1 \,|y_2| \cdots |y_n,$ yazılabilir.

Örnek

- Aşağıdaki gramerden tüm birim üretimleri çıkarınız?

$$S \rightarrow Aa|B,$$
$$B \rightarrow A|bb,$$
$$A \rightarrow a\,|bc|\,B.$$



- Yandaki açılımlar için $S \overset{*}{\Rightarrow} A, S \overset{*}{\Rightarrow} B, B \overset{*}{\Rightarrow} A,$ ve $A \overset{*}{\Rightarrow} B$
- Birim tekrarları kaldıralım ve türetimleri ekleyelim. Birim tekrarlar kaldırıldığında aşağıdaki kurallar elde edilir.

$$S \rightarrow Aa,$$
$$A \rightarrow a|bc,$$
$$B \rightarrow bb,$$

Örnek

- Yeni kurallardan birim türetimleri elde edersek aşağıdaki kurallara ulaşırız.

$$
\begin{aligned}
S &\rightarrow a\,|bc|\,bb,\\
A &\rightarrow bb,\\
B &\rightarrow a|bc,
\end{aligned}
$$

- Bu kurallara önce bulduğumuz birim kurallardan arındırılmış kuralları eklersek eşlenik grameri aşağıdaki gibi bulmuş oluruz.

$$
\begin{aligned}
S &\rightarrow a\,|bc|\,bb|Aa,\\
A &\rightarrow a\,|bb|\,bc,\\
B &\rightarrow a\,|bb|\,bc.
\end{aligned}
$$

Sadeleştirme

- Gramerin sadeleştirilmesi için önce aşağıdaki adımların izlenmesi doğrudur.
  1. Boş üretimlerin (λ) kaldırılması
  2. Birim üretimlerin kaldırılması
  3. Gereksiz üretimlerin kaldırılması

Örnek

- Aşağıdaki gramer kurallarından B açılımını kaldırınız?

$$S \rightarrow aSB|bB$$
$$B \rightarrow aA|b.$$

- Burada S açılımında B gördüğümüz noktada aA yada b yazarak çözüm üretebiliriz. Ancak üreteceğimiz çözümde öncesinde geçen kural kısmı sabit olacaktır. Burada sonuç

S -> aSaA|aSb|baA|bb

Soru

- Aşağıdaki iki farklı gramer kurallarının eşlenik olduğunu gösteriniz?
- G1

$$S \rightarrow abAB|ba,$$
$$A \rightarrow aaa,$$
$$B \rightarrow aA|bb$$

G2

$$S \rightarrow abAaA\ |abAbb|\ ba,$$
$$A \rightarrow aaa$$

Soru

- Aşağıdaki gramer içerisinden gereksiz kuralları kaldırarak grameri sadeleştiriniz?

$$S \rightarrow aS|AB|\lambda,$$
$$A \rightarrow bA,$$
$$B \rightarrow AA.$$

- Aşağıdaki gramerden boş geçişleri kaldırınız.

$$S \rightarrow aSSS,$$
$$S \rightarrow bb|\lambda.$$

Soru

- Aşağıdaki gramerin içerisinden boş geçişleri kaldırınız ve yakaladığı/ürettiği dile ait örnekleri karşılaştırınız?

$$
\begin{aligned}
S &\rightarrow A|B, \\
A &\rightarrow \lambda, \\
B &\rightarrow aBb, \\
B &\rightarrow b.
\end{aligned}
$$