CENG 306 Biçimsel Diller ve Otomatlar
Formal Languages and Automata

PUSH DOWN AUTOMATA
PUMPING LEMMA for PDA

Hazırlayan: M.Ali Akçayol - Gazi Üniversitesi
Bilgisayar Mühendisligi Bölümü

Konular

- Context-Free and Non-Context-Free Languages

Context-Free and Non-Context Free Languages

- Context-free dillerin (CFL) üretilmesi için context-free grammar (CFG) kullanılmaktadır.
- CFL tanınması için PDA makineleri kullanılmaktadır.
- Bir CFG tarafından üretilen dili tanıyan PDA oluşturulabilir.
- Bir dilin CFL veya non-CFL olduğunu belirlemek için yöntemler vardır.
- Aynı Regular dillerde (RL) olduğu gibi
  - **Closure properties** ve
  - **Pumping Lemma for CFL**

  iki farklı yöntem olarak kullanılabilir.

Context-Free and Non-Context Free Languages

Theorem: *Context-free diller* ***union, concatenation ve Kleene star*** *işlemleri altında kapalıdır.*

Proof: *$G_1 = (V_1, \sum_1, R_1, S_1)$ ve $G_2 = (V_2, \sum_2, R_2, S_2)$ iki farklı grammar olsun. Bu iki grammar için nonterminal kümeleri disjoint (ayrışık) olsun. $(V_1 - \sum_1) \cap (V_2 - \sum_2) = \emptyset$*

Union

S yeni bir sembol ve ***$G = (V, \sum, R, S)$ olsun öyle ki***

$\boldsymbol{V = V_1 \cup V_2 \cup \{S\}}$ $\boldsymbol{\sum = \sum_1 \cup \sum_2}$ $\boldsymbol{R = R_1 \cup R_2 \cup \{S \to S_1, S \to S_2\}}$ *olsun. Amaç*

$$L(G) = L(G_1) \cup L(G_2)$$

olduğunu göstermektir. Herhangi bir w string'i için ($S \to S_1$, $S \to S_2$ olduğundan) $S \Rightarrow^*_G w$ olur eger sadece ve sadece $S_1 \Rightarrow^*_G w$ veya $S_2 \Rightarrow^*_G w$ ise

Nonterminaller kümeleri disjoint oldugu için ilk kuralla S_1 veya S_2'ye geçildikten sonra diğerine tekrar dönülmez.

Context-Free and Non-Context Free Languages

Proof: (devam)

Concatenation

S *yeni bir sembol ve* ***G = (V, L, R, S) olsun öyle ki***

$V = V_1 \cup V_2 \cup \{S\}$, $L = L_1 \cup L_2$, $R = R_1 \cup R_2 \cup \{S \rightarrow S_1 S_2\}$ ***olsun.***

Bu *şekilde tanımlanan bir grammar ile* $L(G_1)L(G_2)$ *dili oluşturulabilir.*

Birinci grammar'deki non-terminaller (S_1 içindeki) terminallere dönüştürüldükten sonra ikinci grammar'deki non-terminaller (S_2 içindeki) terminallere dönüştürülür.

Context-Free and Non-Context Free Languages

Proof: (devam)

Kleene star

S *yeni bir sembol ve* ***G = (V, L, R, S) olsun öyle ki***

$V = (V_1 \cup \{S\}$, $L = L_1$, $R = R_1 \cup \{S \rightarrow e, S \rightarrow SS_1\}$ *olsun.*

Bu şekilde tanımlanan bir grammar ile $L(G_1)^*$ *dili oluşturulabilir.*

$S \rightarrow SS_1$ kuralının tekrarı ile dildeki kuralın ($S \rightarrow S_1$) tekrarı istenen sayıda yapılabilir.

Context-Free and Non-Context Free Languages

Tanımlar:

G = (V, ∑, R, S) bir context-free grammar olsun.

G'nin fanout değeri: *∅(G) olarak gösterilir ve R kurallar kümesinde sağ kısmı en uzun olan kuralın sağ kısmındaki sembol sayısıdır.*

Context-Free and Non-Context Free Languages

Tanımlar:

G = (V, ∑, R, S) bir context-free grammar olsun.

G'nin fanout değeri: *∅(G) olarak gösterilir ve R kurallar kümesinde sağ kısmı en uzun olan kuralın sağ kısmındaki sembol sayısıdır.*

Bir parse tree üzerinde path (yol): *root node ile yaprak node arasında farklı node'lardan geçilerek elde edilen sıradır.*

Context-Free and Non-Context Free Languages

Tanımlar:

G = (V, ∑, R, S) bir context-free grammar olsun.

G'nin fanout değeri: *∅(G) olarak gösterilir ve R kurallar kümesinde sağ kısmı en uzun olan kuralın sağ kısmındaki sembol sayısıdır.*

Bir parse tree üzerinde path (yol): *root node ile yaprak node arasında farklı node'lardan geçilerek elde edilen sıradır.*

Yolun length (uzunluk) değeri: *Yol üzerindeki düğümler arası çizgi sayısıdır.*

Context-Free and Non-Context Free Languages

Tanımlar:

G = (V, ∑, R, S) bir context-free grammar olsun.

G'nin fanout değeri: *∅(G) olarak gösterilir ve R kurallar kümesinde* ***sağ kısmı en uzun*** *olan kuralın sağ kısmındaki sembol sayısıdır.*

Bir parse tree üzerinde path (yol): *root node ile yaprak node arasında farklı node'lardan geçilerek elde edilen sıradır.*

Yolun length (uzunluk) değeri: *Yol üzerindeki düğümler arası çizgi sayısıdır.*

Bir parse tree için height : *en uzun path (yol) için length değeridir.*

Context-Free and Non-Context Free Languages

Lemma: ***G*** *grammar'ine ait* ***$\emptyset(G)$ fanout değerine*** *ve* ***h height*** *degerine sahip bir parse tree'nin ürettigi string'in length değeri (uzunluk) en çok* ***$\emptyset(G)^h$*** *olabilir.*

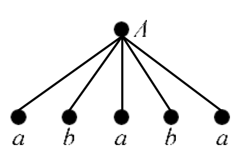
Proof: *h = 1 için parse tree grammar içinde bir kuraldır (2.durum). Ençok $\emptyset(G)^h = \emptyset(G)$ uzunlugunda string üretilir. ($S \rightarrow abc$ (fanout=3) , $S \rightarrow abcabcabc$ (fanout=9))*

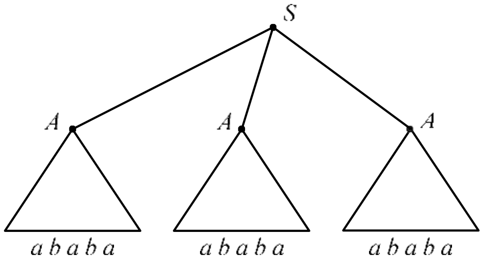
- *h >= 1 olan her h degeri için yeni bir root oluşur ve h-1 yüksekliğindeki parse tree'leri birbirine bağlar.*
- *h+1 için yüksekliği en çok h olan en fazla $\emptyset(G)$ adet parse tree birbirine bağlanır (3.durum). Her parse tree, $\emptyset(G)^h$ uzunluğunda string oluşturur ve toplam en çok $\varphi(G)^{h+1}$ uzunluğunda string oluşur.*

$R_1 = (A \rightarrow ababa, A \rightarrow aba, \ldots)$,

$R_2 = (S \rightarrow AAA, A \rightarrow ababa, \ldots)$

h= 1





Context-Free and Non-Context Free Languages

Pumping Theorem: ***$G = (V, \sum, R, S)$ bir CFG olsun.*** *Uzunluğu $\emptyset(G)^{|V-\Sigma|}$ den büyük her $w \in L(G)$ string'i **$w = uvxyz$** şeklinde yazılabilir. Tüm $n >= 0$ değerleri için **v** veya **y** den birisi boş olmamak kaydıyla **$uv^nxy^nz \in L(G)$** olur. Bunu sağlamayan non-context-free dildir.*

***Örnek**: $L = \{a^nb^nc^n : n >= 0\}$ dili non-context-free'dir. Bir CFG $G = (V, \sum, R, S)$ için $L = L(G)$ oldugunu düşünelim. $w = a^nb^nc^n$ dile ait olmalıdır ve $w = uvxyz$ şeklinde gösterilebilmelidir.*

Burada v veya y' den en az birisi boş olamaz ve tüm $n >= 0$ için $uv^nxy^nz \in L(G)$ olmalıdır:

- *Eger vy string'i a, b ve c'lerin üçünü de içerirse v ve y'den birisi en az ikisini (ab, bc) içerir. uv^2xy^2z string'i a,b,c 'lerin sırasını bozar. b'lerden sonra a veya c'lerden sonra b gelir.*
- *Eger vy string'i a, b ve c'lerin bir kısmını içerirse uv^2xy^2z string'i eşit olmayan sayıda a, b ve c'ler üretir.*

Context-Free and Non-Context Free Languages

Theorem: *Context-free diller* ***complementation*** *ve* ***intersection*** *için kapalı degildir.*

Proof: $\{a^n b^n c^m : m,n \geq 0\}$ *ile* $\{a^m b^n c^n : m,n \geq 0\}$ *dilleri context-free'dir.*

Bu iki dilin kesişimi ise

$\{a^n b^n c^n : n >= 0\}$ *olur. Bu dil non-context-free'dir.*

Öyle ise Intersection için kapalı değildir.

$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$ *olduğu için eğer complementation için kapalı olsaydı kesişim içinde kapalı olurdu. (Birleşim altında CFL'nin kapalı olduğunu biliyoruz.)*

Örnek Sorular

Aşağıdaki ispatta $a^n b^{2n} a^n$ dilinin bağlamdan bağımsız olduğunun kanıtını yanlış yapan nedir?

(1) Hem $\{a^n b^n : n \geq 0\}$ hem de $\{b^n a^n : n \geq 0\}$ Bağlamdan bağımsızdır.

(2) $a^n b^{2n} a^n = \{a^n b^n\}.\{b^n a^n\}$ yazılabilir.

(3) Bağlamdan bağımsız diller **kaynaştırma** altında kapalılık özelliğine sahip olduğundan, $a^n b^{2n} a^n$ **bağlamdan bağımsızdır**.

-

  iki dili birleştirdiğimizde, hala ayrı değişkenlerle iki ayrı dil tanımımız var. Yani iki n farklıdır.
- $a^n b^{2n} a^n = \{a^n b^n\}.\{b^n a^n\}$ doğru fakat

$L=\{a^n b^{2n} a^n, n \geq 0\}=L1.L2$ öyle ki

$L1=\{a^n b^n, n \geq 0\}$

$L2= \{b^m a^m, m \geq 0\}$ olarak düşünmek lazım. (n sadece bir gösterilim.)

Örnek Sorular

$L = \{a^n b^m a^n : n \geq m\}$ context-free midir? PL ile gösteriniz.

Pumping Theorem for CFG: ***G = (V, ∑, R, S) bir CFG olsun.*** *Uzunluğu $\emptyset(G)^{|V-\Sigma|}$ den büyük her $w \in L(G)$ string'i* ***w = uvxyz*** *şeklinde yazılabilir. Tüm n>= 0 değerleri için* ***v*** *veya* ***y*** *den birisi boş olmamak kaydıyla* $uv^nxy^nz \in L(G)$ *olur. Bunu sağlamayan non-context-free dildir.*

$w = a^k b^k a^k$ seçtiğimizde:
Ne v ne de y'nin a ve b bölgelerini geçemeyeceğini biliyoruz, çünkü eğer bunlardan biri olursa, o zaman pumping ile, a ve b sıraları bozulur. Bu nedenle, her birinin w'nin üç bölgesinden (a'nın ilk grubu, b'ler ve a'nın ikinci grubu) olduğu durumları dikkate almamız gerekir.
(1, 1) a'ların ilk grubu artık ikinci grupla eşleşmeyecektir.
(2, 2) Eğer b'ye pumping yaparsak, bir noktada a'dan daha fazla b olacaktır ve buna izin verilmez.
(3, 3) (1, 1) 'e benzer
(1, 2) a'ları bölge 1'e ya (ya da her ikisini) pompalamalıyız, yani iki bölge eşleşmeyecek ya da, eğer y boş değilse, b'lere pompalayacağız ama sonunda a'dan daha fazla b olacaktır.
(2, 3) (1, 2) 'ye benzer
(1, 3) | vxy | ≤ M, bu yüzden vxy b'nin orta bölgesini kapatamaz.

Pumping Theorem for CFG: ***G = (V, ∑, R, S) bir CFG olsun.*** *Uzunluğu* $\emptyset(G)^{|V-\Sigma|}$ *den büyük her w∈ L(G) string'i* ***w = uvxyz*** *şeklinde yazılabilir. Tüm n>= 0 değerleri için* ***v*** *veya* ***y*** *den birisi boş olmamak kaydıyla* $uv^nxy^nz \in L(G)$ *olur. Bunu sağlamayan non-context-free dildir.*

Örnek Sorular

$L = \{xx^R yy^R zz^R : x, y, z \in \{a, b\}^*\}$ bağlamdan bağımsız mıdır?

- $\{xx^R : x \in \{a, b\}^*\}$ CFL olduğunu biliyoruz.

CFL concatenation altında kapalı.

Öyleyse $L = \{xx^R yy^R zz^R : x, y, z \in \{a, b\}^*\}$ CFL'dir.

Ama bunu doğrudan L için bir dilbilgisi vererek de yapabiliriz:

$$S \rightarrow AAA$$

$$A \rightarrow aAa$$

$$A \rightarrow bAb$$

$$A \rightarrow \varepsilon$$

Ödev

- Problemleri çözünüz 3.5.2c (sayfa 148)
- Problemleri çözünüz 3.5.5a (sayfa 148)
- Problemleri çözünüz 3.5.14a, 3.5.14c (sayfa 149)