

BM 305 Biçimsel Diller ve Otomatlar (Formal Languages and Automata)

Hazırlayan: M.Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

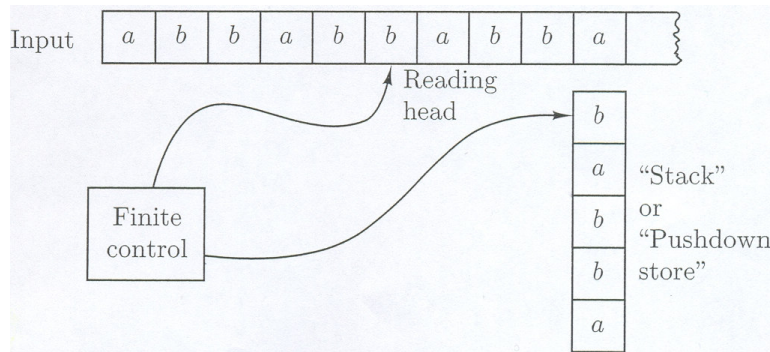
Konular

- Pushdown Automata (PDA)
- Pushdown Automata and Context-Free Grammars

Pushdown Automata

- Regular olmayan context-free dillerin tanınması finite automata ile yapılamaz.
- $\{ww^R : w \in \{a, b\}^*\}$ dilini tanımak için ekstra ne gerekir?
- Bu dil $S \rightarrow aSa$, $S \rightarrow bSb$ ve $S \rightarrow \epsilon$ kurallarına sahip bir grammar tarafından üretilebilir.
- Böyle bir dili tanıyan cihazın string'in yarısına geldiğinde ilk yarısını hafızada tutması ve ikinci yarısını bununla ters sırada karşılaştırması gerekir.
- Bu tür dillerin tanınmasında **pushdown automata** kullanılır.
- Pushdown automata hafıza birimi olarak bir **stack** kullanır.

Pushdown Automata



- Nonregular dillere diğer bir örnek dengelenmiş parantez üreten dildir.
- Pushdown automata bu dili tanırken stack count sıfır ile başlar. Her sol parantez gelişinde 1 artar ve her sağ parantez gelişinde 1 azalır.
- String soldan sağa doğru okunurken negatif değere ulaşılması veya string bittiğinde pozitif değer olması kabul edilmeyen string'i gösterir.
- String bittiğinde stack count sıfır ise string kabul edilir.

Pushdown Automata

Tanım:

- Pushdown automata $M = (K, \Sigma, \Gamma, \Delta, s, F)$ şeklinde bir altılı (6-tuple) ile tanımlanır.

K durumlar

Σ alfabe (giriş sembolleri için)

Γ alfabe (stack sembolleri için)

$s \in K$ başlangıç durumu

$F \subseteq K$ sonuç durumları kümesi

Δ geçiş ilişkisi $(K \times (\Sigma \cup \{e\}) \times \Gamma^*) \times (K \times \Gamma^*)$

Pushdown Automata

$((p, a, \beta), (q, \gamma)) \in \Delta$ ise;

- Pushdown automata p durumundadır.
- input tape'ten a okunmuştur. ($a = e$ ise input tape'e başvurulmaz)
- Stack üzerinde en üstte β okunarak γ ile değiştirilir.
- q durumuna geçilir.
- $\beta = e$ ise stack'tan okuma yapılmaz.
- $\gamma = e$ ise stack'a yazma yapılmaz.

Pushdown Automata

- Bu pushdown automata nondeterministic'tir.
- push stack'in en üstüne sembol/semboller ekler, pop ise en üstteki sembolü/sembolleri alır.
- $((p, u, e), (q, a))$ a'yı push yapar, $((p, u, a), (q, e))$ a'yı pop yapar.
- Okunan string'in soldaki kısmı sonraki işlemler üzerinde etki yapmaz.
- Pushdown automata için configuration $K \times \Sigma^* \times \Gamma^*$ olarak tanımlanır.
- K automata'nın bulunduğu durumu, Σ^* input string'te okunmamış kısmı, Γ^* ise stack'taki string'i gösterir.
- (q, w, abc) için stack'ta en üstte a , en altta c vardır.

Pushdown Automata

- (p, x, α) bir adım sonra (q, y, ζ) 'yı oluşturur ve $(p, x, \alpha) \vdash_M (q, y, \zeta)$ şeklinde gösterilir eğer;
- $((p, a, \beta), (q, \gamma)) \in \Delta$ şeklinde bir ilişki varsa,
 - ve $x = ay$, $\alpha = \beta\eta$, ve $\zeta = \gamma\eta$, $\eta \in \Gamma^*$ ise

$\vdash_M$ 'in reflexive, transitive, closure'u $\vdash_M^*$ şeklinde gösterilir.

M pushdown automata'ı $w \in \Sigma^*$ string'ini kabul eder eğer $(s, w, e) \vdash_M^* (p, e, e)$, $p \in F$ ise

Konfigürasyonlar $C_0 \vdash_M C_1 \vdash_M \dots \vdash_M C_n$ şeklinde gösterilir.

Eğer $C_0 = (s, w, e)$ ve $C_n = (p, e, e)$ ve $p \in F$ ise w string'i kabul edilir.

Pushdown Automata

Örnek: $L = \{wcw^R : w \in \{a, b\}^*\}$ dilini kabul eden bir pushdown automata oluşturalım. ($ababcbaba \in L$, $abcbab \notin L$, $cbcb \notin L$)

$M = (K, \Sigma, \Gamma, \Delta, s, F)$, $K = \{s, f\}$, $\Sigma = \{a, b, c\}$, $\Gamma = \{a, b\}$, $F = \{f\}$
 Δ toplam 5 adet geçiş ilişkisine sahip olsun;

1. $((s, a, e), (s, a))$
2. $((s, b, e), (s, b))$
3. $((s, c, e), (f, e))$
4. $((f, a, a), (f, e))$
5. $((f, b, b), (f, e))$

Otomat string'in ilk yarısını okurken (c'ye kadar) başlangıç durumunu korur ve input tape'ten okuduğunu push eder, c okuduktan sonra final state'e geçer ve input tape'ten okuduğuyla stack'tan okuduğunu karşılaştırır. Nondeterministic pushdown automata'dır.

Pushdown Automata

Örnek: (devam) $L = \{wcw^R : w \in \{a, b\}^*\}$ dilini kabul eden bir pushdown automata oluşturalım.
 $abbcbbba$ için geçişler aşağıdaki tabloda verilmiştir.

State	Unread Input	Stack	Transition Used
s	abbcbbba	e	-
s	bcbba	a	1
s	cbba	ba	2
s	cbba	bba	2
f	bba	bba	3
f	ba	ba	5
f	a	a	5
f	e	e	4

Giriş string'i bittiğinde stack boş değilse, giriş string'i ile stack arasında farklı karakter okuma yapılırsa, giriş string'i bittiğinde ve/veya stack'ta okunacak sembol olmadığında sonuç durumunda (f) değilse string kabul edilmez.

Pushdown Automata

Örnek: $L = \{w \in \{a, b\}^* : w \text{ aynı sayıda } a \text{ ve } b\text{'ye sahiptir.}\}$ dilini kabul eden bir pushdown automata oluşturalım.

$M = (K, \Sigma, \Gamma, \Delta, s, F)$, $K = \{s, q, f\}$, $\Sigma = \{a, b\}$, $\Gamma = \{a, b, c\}$, $F = \{f\}$

Δ toplam 8 adet geçiş ilişkisine sahip olsun;

1. $((s, e, e), (q, c))$ c stack'in sonunu gösterir
2. $((q, a, c), (q, ac))$
3. $((q, a, a), (q, aa))$
4. $((q, a, b), (q, e))$
5. $((q, b, c), (q, bc))$
6. $((q, b, b), (q, bb))$
7. $((q, b, a), (q, e))$
8. $((q, e, c), (f, e))$

Otomat ilk önce stack'a c yazar ve ara duruma (q) geçer. Her a 'ya karşılık b veya b 'ye karşılık a geldiğinde stack'tan pop yapılır diğer durumlarda input ile stack'tan pop edilen concatenate edilerek stack'a push edilir.

Pushdown Automata

Örnek: (devam) $L = \{w \in \{a, b\}^* : w \text{ aynı sayıda } a \text{ ve } b\text{'ye sahiptir.}\}$ dilini kabul eden bir pushdown automata oluşturalım.

abbbabaa için geçişler aşağıdaki tabloda verilmiştir.

State	Unread Input	Stack	Transition	Comments
s	$abbbabaa$	e	-	Initial configuration.
q	$abbbabaa$	c	1	Bottom marker.
q	$bbbabaa$	ac	2	Start a stack of a 's.
q	$bbabaa$	c	7	Remove one a .
q	$babaa$	bc	5	Start a stack of b 's.
q	$abaa$	bbc	6	
q	baa	bc	4	
q	aa	bbc	6	
q	a	bc	4	
q	e	c	4	
f	e	e	8	Accepts.



Pushdown Automata

Tanım:

Her finite automata basit bir pushdown automata olarak görülebilir.

$M = (K, \Sigma, \Delta, s, F)$ bir nondeterministic finite automata ve $M' = (K, \Sigma, \emptyset, \Delta', s, F)$ pushdown automata ve $\Delta' = \{((p, u, e), (q, e)) : (p, u, q) \in \Delta\}$ şeklinde tanımlanır.

M' stack üzerine boş string yazar ve okuma yapmaz böylelikle finite automata'nın geçişlerini simüle eder.



Pushdown Automata and CFGs

- Pushdown automata'lar context-free dilleri tanımak için gerekli olan özelliklere sahiptir.
- PDA'lar özellikle context-free dil olan programlama dillerinin analizinde kullanılmaktadırlar.
- PDA'lar programlama dillerinde syntax analyzer olarak kullanılmaktadır.



Pushdown Automata and CFGs

Theorem: *Her context-free dil bir pushdown otomat tarafından kabul edilir.*

Proof: $G = (V, \Sigma, R, S)$ bir CFG olsun. $L(M) = L(G)$ olacak şekilde bir pushdown otomat oluşturmak zorundayız. Bu otomatın iki durumu (p, q) olsun ve M stack alfabesi olarak terminal ve nonterminalleri (V) kullansın.

$$M = (\{p, q\}, \Sigma, V, \Delta, p, \{q\})$$



Pushdown Automata and CFGs

Proof: (devam)

$$M = (\{p, q\}, \Sigma, V, \Delta, p, \{q\})$$

Δ toplam 3 adet geçiş ilişkisine sahip olsun;

1. $((p, e, e), (q, S))$
2. $((q, e, A), (q, x))$ her bir $A \rightarrow x \in R$ için
3. $((q, a, a), (q, e))$ her $a \in \Sigma$ için

PDA, G 'nin başlangıç sembolü S 'yi stack'a push ederek başlar ve q durumuna geçer. Daha sonraki adımlarda stack'ın en üstündeki A sembolü ile x sembolünü değiştirir ($A \rightarrow x \in R$) veya girişten okunan sembol ile aynı olan stack'ın en üstündeki terminal sembolü pop eder.

Bu PDA leftmost derivation yapar. Nondeterministic çalışır.

Pushdown Automata and CFGs

Örnek: $G = (V, \Sigma, R, S)$, $V = \{S, a, b, c\}$, $\Sigma = \{a, b, c\}$, ve $R = \{S \rightarrow aSa, S \rightarrow bSb, S \rightarrow c\}$ şeklinde tanımlı bir CFG olsun ve $L = \{wcw^R : w \in \{a, b\}^*\}$ dilini oluştursun.

Bu dili tanıyan bir PDA olan $M = (\{p, q\}, \Sigma, V, \Delta, p, \{q\})$ olarak tanımlanabilir.

$$\begin{aligned} \Delta = \{ & ((p, e, e), (q, S)), & (T1) \\ & ((q, e, S), (q, aSa)), & (T2) \\ & ((q, e, S), (q, bSb)), & (T3) \\ & ((q, e, S), (q, c)), & (T4) \\ & ((q, a, a), (q, e)), & (T5) \\ & ((q, b, b), (q, e)), & (T6) \\ & ((q, c, c), (q, e))\} & (T7) \end{aligned}$$

Pushdown Automata and CFGs

Örnek: (devam)

abbcbbba için geçişler aşağıdaki tabloda verilmiştir.

State	Unread Input	Stack	Transition Used
<i>p</i>	<i>abbcbbba</i>	<i>e</i>	-
<i>q</i>	<i>abbcbbba</i>	<i>S</i>	T1
<i>q</i>	<i>abbcbbba</i>	<i>aSa</i>	T2
<i>q</i>	<i>bbcbba</i>	<i>Sa</i>	T5
<i>q</i>	<i>bbcbba</i>	<i>bSba</i>	T3
<i>q</i>	<i>bcbbba</i>	<i>Sba</i>	T6
<i>q</i>	<i>bcbbba</i>	<i>bSbba</i>	T3
<i>q</i>	<i>cbba</i>	<i>Sbba</i>	T6
<i>q</i>	<i>cbba</i>	<i>cbba</i>	T4
<i>q</i>	<i>bba</i>	<i>bba</i>	T7
<i>q</i>	<i>ba</i>	<i>ba</i>	T6
<i>q</i>	<i>a</i>	<i>a</i>	T6
<i>q</i>	<i>e</i>	<i>e</i>	T5



Ödev

- Problemleri çözünüz 3.3.1a, 3.3.1b (sayfa 135)
- Problemleri çözünüz 3.3.2a, 3.3.2b, 3.3.2c, 3.3.2d (sayfa 135)
- Problemleri çözünüz 3.4.1 (sayfa 142)