DİL TEORİSİ ⟨Language Theory⟩

Kayıt⟨Note⟩: Aşağıdakiler, noksan bir dokümandan aktarılmış olup, sadece içerdiği örneklerden yararlanılsın diyedir.

Alfabe: **<u>Boş olmayan</u>**, **<u>sonlu</u>** bir simge kümesi. Ör. Σ= {**0**, **1**}, **T**= {**a**, **b**, **c**}.

Dizgi: Yan yana bir sıraya dizili simgelerden ibaret bir oluşuma bir **dizgi** denir.

Aksi belirtilmedikçe, söz konusu edilen dizgiler hep sonlu olarak kabul edilecektir.

Ör.

0100	**aabab**

Bir **u** dizgisinin uzunluğu |**u**|, onu oluşturan simgelerin, tekrarlar dâhil, sayısıdır.

Ör. u_1= 0100 için $|u_1|$= |0100|= 4; u_2= aabab için $|u_2|$= |aabab|= 5

ε: boş dizgi, $|\varepsilon|$= 0

Örneğin: u = aaba, v = bba

u·v = uv = aababba (bu işleme bitişim⟨**concatenation**⟩ işlemi denir.) ve bu haliyle tek bir dizgi haline gelir.

Bitişim işleminin etkisiz elemanı; $\varepsilon \cdot u = u \cdot \varepsilon = u$

u_1, u_2, u_3 dizgileri için w= $u_1 \cdot u_2 \cdot u_3$ ise u_2, w -nun bir **alt-dizgi**sidir.

Bir dizginin en solunda bulunan bir alt-dizgiye o dizginin **ön-takı**sı, en sağında bulunan bir alt-dizgiye de o dizginin **art-takı**sı denir.

Her dizgi kendisinin hem alt-dizgisi, hem ön-takısı, hem de art-takısıdır.

Bir dizginin alt-dizgisi, ön-takısı veya art-takısı kendisinden farklı ise ona sırasıyla, özalt-dizgisi, özön-takısı veya özart-takısı denir.

Σ^*: Sadece Σ -nın simgelerinden oluşan bütün sonlu dizgiler kümesi.

Örnek: Σ = {0, 1} burada 0 ve 1 simgedir, rakam değildir.

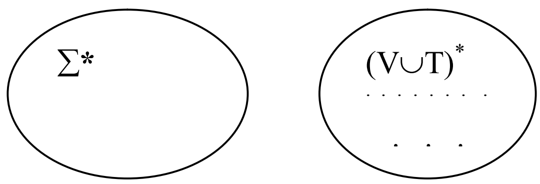
Σ^* = {ε, 0, 1, 00, 01, 10, 11, 000, 001,........,}

burada 0 uzunlukta 1 tane, dizgi vardır.

1 uzunlukta 2 tane, dizgi vardır.

2 uzunlukta 4 tane, dizgi vardır.

... vs.



Herhangi bir dizgi, Σ^* -ın bir elemanı ise onun uzunluğu bir doğal sayıdır.

Eğer $\alpha \in (V \cup T)^*$ ise, o şekilde sonlu bir dizgidir ki, içinde $(V \cup T)$ dışında hiçbir simge yoktur. ($\alpha = \varepsilon$ olabilir). Burada α -nın uzunluğu sıfır ise $\alpha = \varepsilon$ -dur. Uzunluğu 1 ise α ya V -nin bir elemanı ya da T -nin bir elemanıdır. Uzunluğu daha fazla ise o zaman da $(V \cup T)^*$ -ın diğer bir elemanıdır.

L: $\sum$ Alfabesi üzerinde bir dil demek, $L \subseteq \sum^*$ demektir.

← (bitişim) **Simge** (kümeleştirim)→

Dizgi **Alfabe**

(kümeleştirim) → **Dil** ← (Yıldızının altkümesi)

Dilleri temsil için bazen küme gösterimi gibi jenerik ifadeler kullanılabilir; fakat esas üzerinde duracağımız başlıca iki türlü mekanizma vardır:

Dilleri temsil eden mekanizmalar

Tanıyıcılar	**Üreticiler**
«Recognizers»	**«Generators»**
(Hareketliler)	**(Gramerler)**
(«Automata»)	**(«Grammars»)**
(Hesap Mekanizmaları)	

Tanıyıcılar: Belli bir dilin tanıyıcısı, o dildeki ve yalnız o dildeki dizgileri kabul eder.

Üreticiler: Dizgi üretir. Belli bir dilin üreticisi o dildeki dizgileri ve yalnız o dildeki dizgileri üretir.

Gösterim: Herhangi bir simge veya dizgi **x** için,

$x^0 = \varepsilon$

$x^1 = x$

$x^2 = xx$

$x^{n+1} = x^n \cdot x$

$L_1 . L_2 = \{u_1 . u_2 \mid u_1 \in L_1, u_2 \in L_2\}$
Örnek: $L_1 = \{ aa, aab, ababb\}$, $L_2 = \{b, bb\}$
$L_1 . L_2 = \{aab, aabb, \cancel{aabb,}\ aabbb, ababbb, ababbbb\}$
$= \{aab, aabb, aabbb, ababbb, ababbbb\}$
Örnek: $L_1 = \{aa, aab, ababb\}$, $L_2 = \{\varepsilon, b\}$
$L_1 . L_2 = \{ aa, aab, \cancel{aab,}\ aabb, ababb, ababbb\}$
$= \{ aa, aab, \cancel{aab,}\ aabb, ababb, ababbb\}$
$L_2 . L_1 = \{ aa, aab, ababb, baa, baab, bababb\}$
Örnek: $L_1 = \{\varepsilon, aa, ababb\}$, $L_2 = \{aab, b\}$
$L_1 . L_2 = \{aab, b, aaaab, aab, ababbaab, ababbb\}$
$L_1 . L_2 = \{aaaab, aab, ababbaab, ababbb, b\}$

$|L_1 . L_2| \leq |L_1|\ |L_2|$
Her hangi bir L dili için
$L^0 = \{\varepsilon\}$ (yani boş dizgi birlisi)
$L^k = L^{k-1}.L$ ($k \geq 1$ için)
$L^1 = L^0.L = \{\varepsilon\}.L = \{\varepsilon.u \mid u\in L\} = \{u \mid u \in L\} = L$
$L^2 = L^1.L = L.L$
$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \ldots$

Σ : Her alfabe bir dildir.alfabenin her bir elemanı bir simgedir, $a\in\Sigma$, buradaki a 1 uzunluğunda olan bir dizgidir.

Örneğin: $\Sigma = \{a, b\}$
$\Sigma.\Sigma = \{aa, ab, ba, bb\}$
$L \subseteq \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \ldots \cup \Sigma^n$ n bir sabit olduğuna göre, $\Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \ldots\ldots\ldots \cup \Sigma^n$, L sonludur.

$S \rightarrow A$ ($S, A \in V$) sonlandıran bir üretim değildir.
① $S \rightarrow w_1$ $S \Rightarrow^* w_1$
② $S \rightarrow w_2$ (S den içinde A olan bir şey üretmek mümkün değil)

<u>Örnek</u>: $L_1 = \{a^n b^m \mid n, m \geq 1\}$ bu kümenin tipik elemanı: $\underbrace{aa...a}_{n\ adet} \cdot \underbrace{bb...b}_{m\ adet}$

Bu dili kabul eden bir tanıyıcı, aa geldiğinde **<u>ret</u>** durumunda, aab geldiğinde **<u>kabul</u>** durumunda, aabb geldiğinde yine **<u>kabul</u>** durumunda olur. Fakat aabba geldiğinde **<u>ret</u>** durumuna geçer ve bunun sağına ne gelse **<u>kabul</u>**e geçmez.

<u>Örnek</u>: $L_2 = \{a^n b^n \mid n \geq 1\}$ bu kümenin tipik elemanı: $\underbrace{aa...a}_{n\ adet} \cdot \underbrace{bb...b}_{n\ adet}$

Bu dili kabul eden bir tanıyıcı, aa da gelse, aab de gelse **<u>ret</u>** durumunda olur. Eğer aabb gelirse **<u>kabul</u>** durumuna geçer. Fakat aabb -nin sağına daha fazla simge gelirse **<u>ret</u>** durumuna geçer.

ÜRETİCİLER

<u>Gramerler</u>

Tanım: Bir gramer, G = (V, T, S, P) şeklinde bir sıralı 4-lü olarak verilir. Burada;
V: Değişken simgeleri kümesi «set of **V**ariable symbols» (sonlu bir simge kümesi).
T: Uç simgeleri kümesi «set of **T**erminal symbols» (bir alfabe).
S: Başlangıç simgesi «**S**tart symbol» (değişkenler kümesinde özel bir eleman **S**$\in$ **V**) olmaktadır.
Değişken simgeleri ve Uç simgeleri yerine sırasıyla, kısaca **Değişkenler** ve **Uçlar** denmektedir.

V∩T = ϕ (V ile T ayrık kümeler, ortak eleman yok) olup,

V∪T Gramerin bütün simgelerinden ibaret küme olmaktadır.

(V∪T)*, bu gramerin, evrensel dizgi kümesidir. (Bu gramerin, dizgi evreni) Buna göre,

P: Üretim kuralları kümesi «set of **P**roduction rules»; **(V∪T)*.V.(V∪T)*×** **(V∪T)*** çapraz çarpımının boş olmayan bir sonlu altkümesidir.

Kayıt: (V∪T)*.V.(V∪T)*⊂ (V∪T)* **(altküme değil, özaltküme).**
Yani, (V∪T)*⊆ (V∪T)*.V.(V∪T)* **değil!**

Üretim kuralları yerine kısaca **Üretimler** denmektedir.

Alıştırım: (V∪T)*.V.(V∪T)* = (V∪T)* - T* olduğunu ispatlayınız.

P -nin her bir elemanı, (α, β) şeklinde bir sıralı ikilidir;

α∈ (V∪T)*.V.(V∪T)* yani, Gramerin simgelerinden ibaret olan, ve en az bir değişken bulunan bir dizgi.

β∈ (V∪T)* yani, Gramerin simgelerinden ibaret olan bir dizgi.

P -nin böyle bir öğesi (yani üretimi) α→β şeklinde gösterilir. Bu takdirde α -ya bu üretimin "**sol tarafı**" (kısaca "**solu**"), β -ya da bu üretimin "**sağ tarafı**" (kısaca "**sağı**") denir.

→ işareti: "**yeniden yazılır**" veya daha kısa olarak, "**yerine**" şeklinde okunur.

Örnek: $G_1 = (V_1, T, S_1, P_1)$ öyle ki, $V_1=\{ S_1, A\}$, $T=\{0,1\}$, $P_1=\{ S_1\rightarrow 0AA1, 0A1\rightarrow \varepsilon, A\rightarrow 10\}$.

"**Doğrudan sürer**" ve "**Sürer**" bağıntıları:

Bir G= (V, T, S, P) grameri verildiğinde, P -deki üretimlere dayalı olarak (V∪T)* (bu gramerin simgelerinden ibaret dizgiler kümesi) üzerinde tanımlanan ve özel ismi "**G de doğrudan sürer**" olan bir temel bağıntı vardır. $\boxed{\Rightarrow_G}$ işaretiyle gösterilen bu bağıntı şöyle tanımlanır:

$\alpha_0, \alpha_1, \beta_0, \beta_1 \in$ (V∪T)* **«alfa-0, alfa-1, beta-0, beta-1 ait V bileşim T -nin yıldızı»** için

$(\alpha_0\beta_0\alpha_1, \alpha_0\beta_1\alpha_1) \in \boxed{\Rightarrow_G}$ **«alfa-0, beta-0, alfa-1 virgül alfa-0, beta-1, alfa-1 sıralı ikilisi ait "G de doğrudan sürer" bağıntısı»** -dır, şayet $\beta_0\rightarrow \beta_1 \in P$ ise.

Yani herhangi $\gamma_0, \gamma_1\in$ (V∪T)* için $(\gamma_0, \gamma_1) \in \boxed{\Rightarrow_G}$ -dır, şayet γ_0 -ın içinde herhangi bir üretimin solu bir alt-dizgi olarak yer alır da, onun yerine o üretimin sağı yazıldığında γ_1 elde edilirse.

$(\alpha_0\beta_0\alpha_1, \alpha_0\beta_1\alpha_1) \in \boxed{\Rightarrow_G}$ yerine daha elverişli olan

$\alpha_0\beta_0\alpha_1\Rightarrow_G \alpha_0\beta_1\alpha_1$ «$\boldsymbol{\alpha_0\beta_0\alpha_1}$ **G de doğrudan sürer** $\boldsymbol{\alpha_0\beta_1\alpha_1}$» tarzı kullanılır.

Yerel bağıntılarda genelde olduğu gibi, bir $k\geq 0$ için $\Rightarrow_G$ bağıntısının k -ıncı kuvveti $\Rightarrow_G^k$ olarak gösterilir ve şayet γ_0, $\gamma_k \in (V\cup T)^*$ için $\gamma_0 \Rightarrow_G^k \gamma_k$ ise, özelde (gramerlere mahsus olarak) "**γ_0 G de k adımda sürer γ_k**" denir.

Her yerel bağıntının olduğu gibi, $\Rightarrow_G$ bağıntısının da, $\Rightarrow_G^*$ olarak gösterilen bir **yansıyıcı ve geçişli kapatanı** vardır. $(V\cup T)^*$ üzerindeki bu bağıntıya "**G de sürer**" denmektedir.

G anlaşıldığı takdirde açıkça belirtilmez; $\Rightarrow_G$ yerine sadece $\Rightarrow$ yazılarak **«doğrudan sürer»** veya $\Rightarrow_G^k$ yerine sadece $\Rightarrow^k$ yazılarak "**k adımda sürer**" veya $\Rightarrow_G^*$ yerine sadece $\Rightarrow^*$ yazılarak "**sürer**" denir.

Kayıt: Herhangi $\gamma_1, \gamma_2 \in (V\cup T)^*$ için $\gamma_1 \Rightarrow^* \gamma_2$ Æ bir $k\geq 0$ için $\gamma_1 \Rightarrow^k \gamma_2$ ise.

"**Sürer**" bağıntısı özyineleyişli olarak da tanımlanabilir:

γ_0 **sürer** γ_2 şayet,

$\gamma_2 = \gamma_0$ ise veya herhangi bir $\gamma_1 \in (V\cup T)^*$ için $\gamma_0 \Rightarrow^* \gamma_1$ ve $\gamma_1 \Rightarrow_{\cancel{G}} \gamma_2$ ise.

Kayıt: Herhangi $\gamma \in (V\cup T)^*$ için "**γ sıfır adımda sürer γ**" olmakla beraber, bir $k>0$ için "**γ k adımda sürer γ**" da olabilir. Ancak ikinci hal, verilen gramere bağlıdır.

Kayıt: $\gamma \Rightarrow^* \gamma$ her zaman geçerlidir. Fakat $\gamma \Rightarrow \gamma$ her zaman geçerli değildir. Genelde, bir dizginin diğerini "**doğrudan sürer**" oluşu **sürer** oluşunu gerektirir. Ancak "**sürer**" oluşu, **doğrudan sürer** oluşunu gerektirmez. Örneğin, $A\in V$ olsun; $A\Rightarrow^0 A$ (A sıfır adımda sürer A) dolayısıyla, $A\Rightarrow^* A$ -dır. Fakat $A\Rightarrow A$ dolayısıyla $A\Rightarrow^1 A$ (A bir adımda sürer A) olur, Æ $A\rightarrow A\in P$ ise.

<u>Bir Gramerin Ürettiği Dil</u>:

Tanım: G= (V, T, S, P) bir gramer olsun; herhangi $\alpha\in (V\cup T)^*$ bu gramer itibariyle bir "**cümlesel kalıp**" «**sentential form**» olur, şayet $S\Rightarrow_G^* \alpha$ ise. G gramerindeki bütün cümlesel kalıplar kümesi:

$$C(G) = \{\alpha\in (V\cup T)^* \mid S\Rightarrow_G^* \alpha\}$$

Verilen bir G= (V, T, S, P) gramerinin **ürettiği dil**, değişken simgesi bulunmayan cümlesel kalıplarının kümesidir:

$$L(G) = \{w\in T^* \mid S\Rightarrow_G^* w\};\ L(G) = C(G) \cap T^*$$

Kayıt: Farklı isimlenenler hariç, mümkün olan en küçük gramer, G_k= ({S}, {1}, S, P_k) öyle ki, bir $\alpha \in (V \cup T)^*$ için P_k= {S→ α} olmaktadır. Eğer bir $w \in T^*$ için α= w ise $L(G_k)$= {w}, aksi halde $L(G_k)$= {}; yani her halükârda $|L(G_k)| \leq 1$ olur.

Örnek: Yukarıda verilen $G_1 = (V_1, T, S_1, P_1)$ öyle ki, $V_1=\{S_1, A\}$, T={0,1},

P_1: 1. $S_1 \rightarrow 0AA1$ 2. $0A1 \rightarrow \varepsilon$ 3. $A \rightarrow 10$ için,

"**Doğrudan sürer**" bağıntısında kullanılan üretimlerin sayılarını da alt-indis olarak gösterelim,

$S_1 \Rightarrow_1 0AA1$,	$0\mathbf{A}A1 \Rightarrow_3 \mathbf{010}A1$,	$010\mathbf{A}1 \Rightarrow_3 010\mathbf{101}$
$S_1 \Rightarrow_1 0AA1$,	$0A\mathbf{A}1 \Rightarrow_3 0A\mathbf{101}$,	$0\mathbf{A}101 \Rightarrow_3 0\mathbf{10}101$
$S_1 \Rightarrow_1 0AA1$,	$0\mathbf{A}A1 \Rightarrow_3 0\mathbf{10}A1$,	$01\mathbf{0A1} \Rightarrow_2 01$
$S_1 \Rightarrow_1 0AA1$,	$0A\mathbf{A}1 \Rightarrow_3 0A\mathbf{101}$,	$\mathbf{0A1}01 \Rightarrow_2 01$

Demek ki, $L(G_1)$ = {010101, 01}

Bazı özel şekillerdeki Üretimler:

Bir G= (V, T, S, P) gramerinde, herhangi u, v ∈T*; A, B∈V için,

A→ u şeklindeki bir üretime **bitiren üretim «terminating production»** denir.

(Bazı hatalara yol açtığından, solu birden uzun hiçbir **üretim**e **bitiren üretim** denmeyecek!)

u= ε olduğu takdirde bu aynı zamanda bir (A -yı) "**silen üretim**" «**erasing production**» olur.

A→ uB şeklindeki bir üretime **sağ-doğrusal üretim «right-linear production»**,

A→ Bu şeklindeki bir üretime **sol-doğrusal üretim «left-linear production»**,

A→ uBv şeklindeki bir üretime **doğrusal üretim «linear production»** denir.

A→ B şeklindeki bir üretime "**birim üretim**" denir. Böyle bir üretim, hem **sağ-doğrusal**, hem **sol-doğrusal**, hem de **doğrusal** üretim tanımına uyar.

Gramerler, üretimlerine göre sınıflanırlar. Buna göre, bazı gramer sınıfları:

<u>Bitiren Gramerler</u>: Üretimleri bitiren üretimlerden ibaret gramerler. Bunların ürettiği diller, **sonlu diller**dir.

<u>Sağ-Doğrusal Gramerler</u>: Üretimleri sağ doğrusal ve bitiren üretimlerden ibaret gramerler. Bunların ürettiği dillere, <u>Sağ-Doğrusal Diller</u> denir.

<u>Sol-Doğrusal Gramerler</u>: Üretimleri sol doğrusal ve bitiren üretimlerden ibaret gramerler. Bunların ürettiği dillere, <u>Sol-Doğrusal Diller</u> denir.

<u>Doğrusal Gramerler</u>: Üretimleri doğrusal ve bitiren üretimlerden ibaret gramerler. Bunların ürettiği dillere, <u>Doğrusal Diller</u> denir.

Aynı dili üreten iki gramere "**eşdeğer**" denir.

Bir Doğrusal Gramer	Eşdeğer bir Doğrusal Gramer
	$S \rightarrow aA$
$S \rightarrow aSb$	$A \rightarrow Sb$
$S \rightarrow \varepsilon$	$S \rightarrow \varepsilon$

Üretilen dil: $\{a^n b^n \mid n \geq 0\}$

Ayrıca, herhangi $A \in V$; $\gamma_0, \gamma_1, \gamma_2 \in (V \cup T)^*$ için,

$A \rightarrow \gamma_0$ şeklindeki bir üretime **bağlama-duyarsız üretim**,

$\gamma_1 \rightarrow \gamma_2$ şeklinde ve $|\gamma_1| \leq |\gamma_2|$ olan bir üretime **kısaltmayan üretim** denir.

Bunlara göre de,

Bağlama-Duyarsız Gramerler ⟨**Context-Free Grammars**⟩: Üretimleri **bağlama-duyarsız** üretimlerden ibaret gramerlerdir. Bunların ürettiği dillere, Bağlama-Duyarsız Diller denir.

Bağlama-Duyarlı Gramerler ⟨**Context-Sensitive Grammars**⟩: Üretimleri **kısaltmayan** üretimler ile başlangıç simgesini silen ($S \rightarrow \varepsilon$ şeklindeki) üretimden ibaret gramerlerdir. Şu var ki, başlangıç simgesini silen üretim bulunduğu takdirde, başlangıç simgesi hiçbir üretimin sağında bulunmamalıdır. Bunların ürettiği dillere, Bağlama-Duyarlı Diller denir.

Kayıt: Silen üretimler dışındaki her bağlama-duyarsız üretim, aynı zamanda bir kısaltmayan üretim olmaktadır.

==

Gramer sınıfı	**Üretimlerin olabileceği şekiller**	**Üretilen diller sınıfı**
Bitiren Gramerler	Bitiren	Sonlu diller
Sağ-Doğrusal Gramerler (Cins-3)	Bitiren, Sağ-doğrusal	Sağ-Doğrusal Diller (Cins-3)
Sol-Doğrusal Gramerler (Cins-3)	Bitiren, Sol-doğrusal	Sol-Doğrusal Diller (Cins-3)
Doğrusal Gramerler	Bitiren, Doğrusal	Doğrusal Diller
Bağlama-Duyarsız Gramerler (Cins-2)	Bağlama-duyarsız üretimler	Bağlama-Duyarsız Diller (Cins-2)
Bağlama-Duyarlı Gramerler (Cins-1)	**Kısaltmayan,** başlangıç simgesini silen. Başlangıç simgesini silen üretim varsa, başlangıç simgesi hiçbir üretimin sağında bulunmaz.	Bağlama-Duyarlı Diller (Cins-1)
Gramerler (Kısıtsız) (Cins-0)	Kısıtsız	Sonlu Tanımı Olan Diller (Cins-0) (Turing Tezinden)

Bir gramerin cinsi belirtilirken, onu barındıran sınıfın bu cetvele göre en üstte olanı söylenir.

Örnek: $G_2 = (V_2, T, S_2, P_2)$ öyle ki, $V_2=\{S_2, B, C\}$, $T=\{a, b, c\}$.

<table>
<tr>
<td>PP_2:
1. $S_2 \to aS_2BC$
2. $S_2 \to aBC$
3. $CB \to BC$
4. $aB \to ab$
5. $bB \to bb$
6. $bC \to bc$
7. $cC \to cc$</td>
<td>Bu bir bağlama-duyarlı gramerdir.

$L(G_2) = \{a^n b^n c^n \mid n \geq 1\}$ -dir.

$\mathbf{S_2 \Rightarrow^*}$ aaabbbccc olduğu, sağ tarafta “⇒” bağıntısı cinsinden gösterilmektedir.</td>
<td rowspan="2">$\mathbf{S_2 \Rightarrow_1 aS_2BC}$
$aS_2BC \Rightarrow_1 a\mathbf{aS_2BC}BC$
$aa\mathbf{S_2}BCBC \Rightarrow_2 aa\mathbf{aBC}BCBC$
$aaaB\mathbf{CB}CBC \Rightarrow_3 aaaB\mathbf{BC}CBC$
$aaaBBC\mathbf{CB}C \Rightarrow_3 aaaBBC\mathbf{BC}C$
$aaaBB\mathbf{CB}CC \Rightarrow_3 aaaBB\mathbf{BC}CC$
$aa\mathbf{aB}BBCCC \Rightarrow_4 aa\mathbf{ab}BBCCC$
$aaa\mathbf{bB}BCCC \Rightarrow_5 aaa\mathbf{bb}BCCC$
$aaab\mathbf{bB}CCC \Rightarrow_5 aaab\mathbf{bb}CCC$
$aaabb\mathbf{bC}CC \Rightarrow_6 aaabb\mathbf{bc}CC$
$aaabbb\mathbf{cC}C \Rightarrow_7 aaabbb\mathbf{cc}C$
$aaabbbc\mathbf{cC} \Rightarrow_7 aaabbbc\mathbf{cc}$</td>
</tr>
<tr>
<td colspan="2">Üretim-3 -ün bütün C -ler en sağa gelene kadar uygulandığını, aksi halde cümlesel kalıpta aaabbccBC dizgisinde olduğu gibi c -lerin sağında B kalacağını, ve böyle bir cümlesel kalıbın doğrudan sürer olduğu hiçbir dizginin bulunmadığını, dolayısıyla a*b*c* dizilimine uymayan bir uçlar dizgisinin elde edilemeyeceğini kaydediniz.</td>
</tr>
</table>

$L(G_2) = \{a^n b^n c^n \mid n \geq 1\}$ olduğunun ispatı için resmi bir yaklaşım:

Herhangi $w \in \{a^n b^n c^n \mid n \geq 1\}$ Æ $w \in L(G_2)$ olduğu, yani

hem $\{a^n b^n c^n \mid n \geq 1\} \subseteq L(G_2)$ hem de $L(G_2) \subseteq \{a^n b^n c^n \mid n \geq 1\}$ olduğu gösterilir.

Bunu iki aşamada yapalım:

1- Bir dizgi $\{a^n b^n c^n \mid n \geq 1\}$ -de varsa $L(G_2)$ -de de vardır:

İspat: $\{a^n b^n c^n \mid n \geq 1\}$ -deki dizgilerden herhangi birini temsilen $a^n b^n c^n$ dizgisini alalım.

$$\mathbf{S_2 \Rightarrow_1^{n-1} a^{n-1} S_2 (BC)^{n-1} \Rightarrow_2 a^n (BC)^n \Rightarrow_3^* a^n B^n C^n}$$

$$\Rightarrow_4 \mathbf{a^n} b \mathbf{B^{n-1} C^n} \Rightarrow_5^{\mathbf{n-1}} \mathbf{a^n} b^{\mathbf{n}} \mathbf{C^n} \Rightarrow_6 \mathbf{a^n} b^{\mathbf{n}} c \mathbf{C^{n-1}} \Rightarrow_7^{\mathbf{n-1}} \mathbf{a^n} b^{\mathbf{n}} c^{\mathbf{n}} \in L(G_2)$$ ☺

2- Bir dizgi $L(G_2)$ -de varsa $\{a^n b^n c^n \mid n \geq 1\}$ -de de vardır; veya başka bir deyişle,

bir dizgi $\{a^n b^n c^n \mid n \geq 1\}$ -de yoksa $L(G_2)$ -de de yoktur.

İspat: G_2 -deki (4-7 no.lu olan) bitiren üretimleri incelediğimizde, B, C değişkenlerinin sırasıyla, b ve c uçları için yer tutmak üzere kullanıldığını görüyoruz. Başlangıçta kullanılmak durumunda olan 1 ve 2 no.lu üretimler Cümlesel kalıplarda a, B, C simgelerinin eşit sayılarda olmalarını sağlamaktadır. Dolayısıyla, bir $w \in L(G_2)$ ise, eşit sayıda a -lar, b -ler ve c -lerden oluşur. Böyle bir dizginin $\{a^n b^n c^n \mid n \geq 1\}$ içinde olmaması, ancak a -lar, b -ler ve c -lerin diziliminin a*b*c* dizilimine aykırı düşmesiyle mümkündür. G_2 -deki üretimleri bu açıdan incelediğimizde, 7 no.lu üretimin kullanılabilmesi için cümlesel kalıpta c çıkmış olması gerektiği, bunun için daha önce 6 no.lu üretimin kullanılmış olması gerektiği, bunun için de, daha önce b -lerin elde edilişine 4 no.lu üretimle başlanmış olması gerektiği anlaşılmaktadır. B, C değişkenlerinin uçlara donüşebilmesinde soldan sağa bir sıra takibi zorlanmaktadır. Dolayısıyla bir C -nin sağında kalan bir B hiçbir zaman uç simgesine dönüşemez. ☺

Örnek: $G_3= (V_3, T, S_3, P_3)$ öyle ki,

$V_3= \{ S_3, B\}$, $T= \{a, b, c\}$,

$\mathbf{P_3}$:
1. $S_3 \rightarrow aS_3Bc$
2. $S_3 \rightarrow abc$
3. $cB \rightarrow Bc$
4. $bB \rightarrow bb$

$\mathbf{S_3} \Rightarrow_1 \mathbf{aS_3Bc}$	$aaabBc\mathbf{cB}c \Rightarrow_3 aaabBc\mathbf{B}cc$
$aS_3Bc \Rightarrow_1 a\mathbf{aS_3Bc}Bc$	$aaabB\mathbf{cB}cc \Rightarrow_3 aaabB\mathbf{B}ccc$
$aa\mathbf{S_3}BcBc \Rightarrow_2 aa\mathbf{abc}BcBc$	$aaa\mathbf{bB}Bccc \Rightarrow_4 aaa\mathbf{bb}Bccc$
$aaab\mathbf{cB}cBc \Rightarrow_3 aaab\mathbf{Bc}cBc$	$aaab\mathbf{bB}ccc \Rightarrow_4 aaab\mathbf{bb}ccc$

Alıştırım: G_3 eşdeğer G_2 yani $L(G_3) = \{a^nb^nc^n \mid n \geq 1\}$ olduğunu ispatlayınız.

Örnek: Önceki G_3 gramerinin üretimlerine bir "başlangıç simgesini silen" üretim ilâve ederek şöyle bir grameri elde edelim: $G_4= (V_3, T, S_3, P_4)$, $\mathbf{P_4}= \mathbf{P_3} \cup \{S_3 \rightarrow \varepsilon\}$. Bu başlangıç simgesini silen üretimden ötürü, $\varepsilon \in L(G_4)$ olur. buradan iki soru çıkar:

1. G_4 ile G_3 -ün cinsleri aynı olur mu?
2. $L(G_4)= L(G_3) \cup \{\varepsilon\}$ olur mu?

CK: S_3, aS_3Bc, aBc(5. $S_3 \rightarrow \varepsilon$ olursa), aS_3Bc, ...

Örnek: Yukarıdaki G_3 gramerinden şöyle bir gramer daha elde edelim:

$G_5= (V_5, T, S_5, P_5)$. Öyle ki, $V_5= V_3 \cup \{S_5\}$, $P_5= \mathbf{P_3} \cup \{S_5 \rightarrow \varepsilon, S_5 \rightarrow S_3\}$. Bu takdirde,

1. $L(G_5)= L(G_3) \cup \{\varepsilon\}$ olur mu?
2. G_5 ile G_3 -ün cinsleri aynı mı olur?

Bir bağlama-duyarlı dilde ε bulunabilmesi için, Bağlama-duyarlı gramer tanımına uygun bir $G= (V, T, S, P)$ gramerinde, $S \rightarrow \varepsilon$ yer almasına izin verilir. Ancak, böyle bir üretim bulunduğu takdirde, hiçbir üretimin sağında o başlangıç simgesinin bulunmaması gerekir. ε için böyle bir kuralın olması, $\gamma_1, \gamma_2 \in (V \cup T)^*$ için, $\gamma_1 \Rightarrow^* \gamma_2$ olduğunda, şayet $\gamma_1 \neq S$ ve $\gamma_2 \neq \varepsilon$ ise $|\gamma_2| \geq |\gamma_1|$ olması içindir.

Aidiyet problemi... verilen w ile L(G), problem: $w \in L(G)$?

$\gamma_1, \gamma_2 \in (V \cup T)^*$ için, $\gamma_1 \Rightarrow^* \gamma_2$ olduğunda, şayet $\gamma_1 \neq S$ ve $\gamma_2 \neq \varepsilon$ ise $|\gamma_2| \geq |\gamma_1|$

$S \Rightarrow^* \gamma$; $S \Rightarrow^* w$?

Lema: $G= (V, T, S, P)$ bir bağlama-duyarlı gramer olsun, $\gamma_1, \gamma_2 \in (V \cup T)^*$ için eğer $\gamma_2 \neq \varepsilon$ ve $\gamma_1 \Rightarrow^* \gamma_2$ ise $|\gamma_2| \geq |\gamma_1|$ olur.

<u>Cins-0 Gramerler</u>: Kısıtsız gramerler. Bütün gramerleri kapsayan sınıftır. Bunların ürettiği dillere, <u>Cins-0 Diller</u> denir. Böyle diller, sonlu bir şekilde tanımlanabilen bütün dillerdir.

Kısıtsız gramerler için bazen $\mathbf{P} \subseteq (V\cup T)^* \times (V\cup T)^*$ olarak daha da **gevşetilebilir**. Yani “üretimin solunda hiç değişken olmasa da olur” denebilir. Ne var ki, üretilen diller sınıfında farklılık olmaz.

<u>Açıklayış:</u> Bazen ‹isim› şeklinde açılı tırnaklar içinde, görünümü anlam çağrıştıran simgeler oluşturulur. Böyle olduğunda örneğin ‹isim›, altı simgelik bir dizgiyi değil, “‹” şekilciğinden “›” şekilciğine kadar bir tek simgeyi temsil eder. O zaman, şu uzunluk ifadeleri geçerli olur: |‹isim›|= 1; |‹isim›‹isim›|= 2; |‹isim›a‹isim›b|= 4; |‹isim-a›‹isim-b›|= 2.

<u>Örnek:</u> Bir harf veya bir harfin sağına bitiştirilmiş harf-rakam -lardan oluşan isimler kümesini temsil eden bir gramer: $\mathbf{G_{is} = (V_{is}, T_{is}}$, **‹isim›**, $\mathbf{P_{is})}$

P_{is} :
1. ‹isim› → ‹harf›
2. ‹isim› → ‹isim›‹harf-rakam›
3. ‹harf-rakam› → ‹harf›
4. ‹harf-rakam› → ‹rakam›

5.-8. ‹harf› → a/b/c/d (“/” kullanımı, “veya” yerine) → = ‹harf› → a, ‹harf› → b, ‹harf› → c, ‹harf› → d

9.-18. ‹rakam› → 0/1/.../9

(Üretimler kümesinin elemanlarına, işaret amacıyla sıra sayısı verilmiştir.)

Yukarıdaki gramerde;

$\mathbf{V_{is}}$ **= {‹isim›,‹harf›,‹harf-rakam›,‹rakam›}** : Değişkenler kümesidir.

$\mathbf{T_{is}}$ **= {a, b, c, d, 0, 1, …, 9}** : Alfabedir.

‹isim› : Başlangıç Simgesidir.

$\mathbf{P_{is}}$ **:** Üretimler kümesidir. (18 elemanlı.)

Bu gramerin ürettiği dildeki simge dizgileri, aşağıdaki gibi yöntemlerle elde edilirler.

Başlangıç simgesi

‹isim› $\Rightarrow_2$ **‹isim›**‹harf-rakam› $\Rightarrow_2$ **‹isim›**‹harf-rakam›‹harf-rakam›

$\Rightarrow_1$ **‹harf›**‹harf-rakam›‹harf-rakam› $\Rightarrow_8$ d**‹harf-rakam›**‹harf-rakam›

$\Rightarrow_4$ d**‹rakam›**‹harf-rakam› $\Rightarrow_{12}$ d3**‹harf-rakam›** $\Rightarrow_3$ d3**‹harf›** $\Rightarrow_7$ **d3c** $\in T_{is}^*$

<u>Örnek:</u> Dil örneklerinde sık kullanılan bir tanesi şudur: $\{a^n b^n \mid n \geq 1\}$.

Bunun alfabesindeki simgeler a ile b. Bu öyle bir dil ki bir dizgi, eğer ve ancak, soldan itibaren bir veya daha çok sayıda a ve onu takip eden aynı sayıda b -den oluşuyorsa onun elemanı.

Bunu üreten bir gramer: ({S},{a, b}, S, {S → aSb, S → ab})

Üretimler: **1.** S→ aSb, **2.** S→ ab. Yukarıdaki dil, bu iki üretim kuralı ile tanımlanabilir

.

Bu üretim kurallarından birinci kural uygulanırsa aşağıdaki gibi eşit sayıda a -lar ve b -ler üretilebilir.

$S \underset{1}{\Rightarrow} aSb \underset{1}{\Rightarrow} aaSbb \underset{1}{\Rightarrow} \textbf{...} \underset{1}{\Rightarrow} aa...aSbb...b \underset{2}{\Rightarrow} aa...aabb...bb$

Bir bitiren üretim olan ikinci kuralın uygulamasıyla dile ait dizgi elde edilir.

İkinci bir dil $L_2 = \{a^n b^{2n} \mid n \geq 1\}$

1. S→ aSbb, **2.** S→ abb. Şeklinde bir gramerle her bir adet a -ya karşılık iki adet b üretilir.

DİL TEORİSİ NE İŞE YARAR?

1. Öncelikle resmi dillerin teorik temellerinin araştırılmasına ve keşfedilmesine yarar.
2. Programlama dilleriyle ilgili konulara özellikle derleyici yapımına, gerçekleştirilmesine temel teşkil eder.
3. Dil teorisindeki dil tanımına göre her bir dil bir boyutlu desenler kümesidir ve dil teorisiyle bir boyutlu desenleri tanıma yöntemleri geliştirilir.
4. Resmi dilleri tanımlama yöntemlerinden bir tanesi gramerler vasıtasıyladır. Gramerler sonlu veya sonsuz bir takım dilleri tanımlamakta ve tanımakta kullanılan sonlu tanımlardır.
5. Gramerlerin uygulamalarından bir tanesi de biyolojik bazı olayların modellenmesidir. Çünkü sonlu bir tanıma sahip olan bir gramer vasıtasıyla bir büyüme silsilesini temsil eden sonsuz bir kümeyi tanımlamak mümkündür. Bu küçük bir çekirdeğin içerdiği formülleriyle bir ağacın büyümesini temsil etmesine benzer.
6. Gramer tanımlarıyla aksiyomatik mantık sistemi tanımları arasında benzerlik vardır. Dolayısıyla verilen herhangi bir simge dizgisinin bir dile ait olup olmadığı probleminin ilkeleriyle verilen herhangi bir cümlenin bir mantık sistemine göre teorem olup olmadığını belirleme yöntemleri geliştirilebilir.
7. Yukarıda sayılanların dışında dil teorisinin başka uygulamaları da vardır. Bunlar teorinin içine girildikçe meydana çıkar.

DERLEYİCİLER :

Derleyicilerde cereyan eden safhaları üç ana başlık altına yerleştirebiliriz.

1. **Lexical Analysis :**

Token = Gramatic Symbol (variable)

Verilen izlencedeki bu token'ların belirlenmesi

Örnek : Identifier bir Token' dır.

Ör. $\boxed{\text{if}}$ $\boxed{\text{hız}}$ $\boxed{=}$ $\boxed{375}$ $\boxed{\text{then}}$ (5 adet uç simgesi)…

Ör. $\boxed{\text{if}}$ $\boxed{\text{hız}}$ $\boxed{=}$ $\boxed{3A75}$ $\boxed{\text{then}}$ (1hata, 4 adet uç simgesi)…

Ör. $\boxed{\text{if}}$ $\boxed{\text{hız}}$ $\boxed{\text{then}}$ $\boxed{=}$ $\boxed{375}$ (5 adet uç simgesi)…

Değişken anlamsal (semantik) bir öğedir, "Identifier" gramatik bir öğedir (sentaktik)

Anlam yazım

Değişkenler için bir değişkenin kendisi, bir de ismi olduğu gibi sabitler için de bir sabitin kendisi (veya değeri) bir de sabitin yazılışı vardır.

Lexical Analysis safhası bittiği zaman izlence sentaktik analize hazır hale gelir.

2. **Syntactic Analysis:**

Burada izlencenin ilgili dilin yazım kurallarına göre çözümlemesi yapılır ve verilen dizginin ilgili dile ait olup olmadığı belirlenir. Eğer ilgili dile aitse sentaktik yapısı ortaya çıkmış olur. (Yazımsal boyut)

3. **Semantic Analysis (Anlamsal karşılığın bulunması safhası):**

Sentaktik analizi geçen dizgide bulunan sentaktik yapılara tekabül eden semantik yapı oluşturulur. Bu semantik yapının makine dilindeki ifadesi amaç izlenceyi oluşturur. Bu suretle verilen bir izlence, derleyici tarafından amaç koda dönüştürülmüş olur. (Anlamsal boyut)

Bir dilin sentaktik yapısı gramerle ifade edilebildiği gibi Lexical yapılar da daha basit düzeyde olmakla birlikte gramerlerle veya eşdeğer tanım yöntemleriyle tanımlanabilirler.

Böylece dil teorisiyle derleyiciler arasındaki bağlantı kurulmuş oldu.

Pratik açıdan sonsuz kümenin anlamı elemanlarının uzunluklarına bir kısıtlama getirilmemiş bir kümedir.

Ödev: $L(G)= \{a^n b^n \mid 100>n\geq 1\}$ ise, G= ?

$S_1 \rightarrow$ **ab/aabb;** $S_2 \rightarrow S_1$/aaS_1bb/aa S_1bb

$S_2 \rightarrow S_1$/aaS_1bb/aa S_1bb

/aa S_1bb S_1/

$S \rightarrow a^{10} S_1 b^{10}$

$S_1 \rightarrow a^{10} S_2 b^{10}$

$S_2 \rightarrow a^{10} S_3 b^{10}$

$S_3 \rightarrow a^{10} S_4 b^{10}$

$S_4 \rightarrow a^{10} b^{10}$/

Program x;

begin

begin end; begin end; ...

end.

Pascal dili sonsuz bir kümedir. Dizgiler sonsuz değildir. Pascal izlenceleri kümesi sonlu değildir.

Aksiyomatik Mantık Sistemi:

Burada bir aksiyomlar kümesi vardır. Bir akıl yürütüm kuralları kümesi vardır.

Aksiyomlara akıl yürütüm kuralları uygulanarak teoremler elde edilir. Daha sonra teorem ve aksiyomlara uygulanan akıl yürütüm kurallarıyla yeni teoremler elde edilir. Böylece akıl yürütüm kurallarıyla aksiyomlar bir teoremler kümesini tanımlamış olurlar. Buradaki teoremler kümesinin dil teorisindeki benzeri(karşılığı) dildir. Gramerler de bir başlangıç simgesi vardır.

Gramerlerden Örnekler

Örnek: Öyle bir bağlama duyarsız gramer bulunuz ki,

$L(G) = \{a^n(bc)^n \mid n \geq 0\} = \{\varepsilon, abc, aabcbc, ...\}$ (yani bu dilin elemanları o şekildedir ki soldan sağa olmak üzere önce bir dizi sıfır veya daha çok sayıda a -lar gelecek, sonra aynı sayıda bc -ler gelecek.

n = 0 için, $a^n(bc)^n = a^0.(bc)^0 = \varepsilon.\varepsilon = \varepsilon$

$G = (\{S\}, \{a, b, c\}, S, P)$

P: ① $S \rightarrow a\ S\ bc$

② $S \rightarrow \varepsilon$ görüldüğü gibi sadece iki tane üretim vardır.

$S \Rightarrow_{②} \varepsilon$ elde ediyoruz $\therefore$ $S \Rightarrow^{*} \varepsilon$ $\therefore \varepsilon \in L(G)$.

$S \Rightarrow_{①}$ aSbc diyoruz (çünkü böyle bir üretim var)

$\underbrace{a}_{\gamma_1}\ S\ \underbrace{bc}_{\gamma_2} \Rightarrow_{②} \underbrace{a}_{\gamma_1}\ \varepsilon\ \underbrace{bc}_{\gamma_2} = abc$ (2 nolu üretime göre)

$S \xRightarrow{①} a\ S\ bc \xRightarrow{①} aa\ S\ bcbc \xRightarrow{②} aa\ \varepsilon\ bcbc$

Tanım: Bir w dizgisinin "Aksi" w^A, özyinelemeli olarak şöyle tanımlanır:

$$w^A = \begin{cases} \varepsilon, \text{ şayet } w = \varepsilon \text{ ise;} \\ u^A x, \text{ şayet herhangi bir } x \text{ simgesi ve } u \text{ dizgisi için } w = xu \text{ ise.} \end{cases}$$

Örnek: G: ① $S \rightarrow a\ S\ a$

② $S \rightarrow b\ S\ b$

③ $S \rightarrow \varepsilon$

$L(G) = \{\varepsilon, aa, bb, abba, baab, aaaa, bbbb, ...\} = \{u\,u^A \mid u \in \{a,b\}^*\} = \{u \in \{a,b\}^* \mid u = u^A \text{ ve } |u| \text{ çift}\}$

$u\,u^A = (u\,u^A)^A$

Bu dilin elemanları "Palindrom" lardır. Yani sağdan sola dizilişi ile soldan sağa dizilişi aynı olan dizgilerdir.

Örnek: G:① $S \rightarrow a\ S\ a$
② $S \rightarrow b\ S\ b$
③ $S \rightarrow a$
④ $S \rightarrow b$
⑤ $S \rightarrow \varepsilon$

Örnek: DYCK LANGUAGE (DENGELİ Parantezler DİLİ)

G_{D1}: $S \rightarrow SS$ (Burada alfabe sağ ve sol yuvarlak parantezden oluşur)
$S \rightarrow (S)$ $T = \{(,)\}$
$S \rightarrow (\,)$

Grameri bu üç üretimden oluşan dil acaba nasıl bir dil olur. Dengeli parantezlerden oluşan bütün dizgileri bu üç üretimle elde etmek mümkün mü?

G_{D2}: $S \rightarrow SS$ (Burada alfabe yuvarlak ve köşeli parantezlerden oluşur)
$S \rightarrow (S)$ $T = \{(,),[,]\}$
$S \rightarrow [S]$
$S \rightarrow (\,)$ İki ayrı cins parantezin dengelendiği bütün dizgiler amaçlanıyor?
$S \rightarrow [\,]$

Örnek: $G = \{V, T, S, P\}$, $V = \{A, B, S\}$, $T = \{a, b\}$,
$P = \{ S \rightarrow Aba,$
$A \rightarrow BB,$
$B \rightarrow ab,$
$\cancel{AB \rightarrow b}\}$. L(G) = ?

Örnek: $G = \{V, T, S, P\}$, $V = \{S\}$, $T = \{0, 1\}$,
$P = \{ S \rightarrow 11S,$
$S \rightarrow 0\}$.
L(G) = ?

– $L(G)=\{0, 110, 11110, 1111110, \ldots\}$

Örnek: $G = \{V, T, S, P\}$
$V = \{ A, B, C, S\}$, $T = \{a, b, c\}$,
$cbab \in L(G)$? veya $S \Rightarrow^{*} cbab$? sorusu!

Aidiyet problemi için iki yaklaşım: Üstten-aşağı ayrıştırım, Alttan-yukarı ayrıştırım:

G: $S \to AB$

$A \to Ca$

$B \to Ba$

$B \to Cb$

$B \to b$

$C \to cb$

$C \to b$

$\boxed{cbab \in L(G)?}$

Üstten-aşağı ayrıştırım:

$S \Rightarrow AB$

$S \Rightarrow AB \Rightarrow CaB$

$S \Rightarrow AB \Rightarrow CaB \Rightarrow cbaB$

$S \Rightarrow AB \Rightarrow CaB \Rightarrow cbaB \Rightarrow cbab$

Alttan-yukarı ayrıştırım:

$Cab \Rightarrow cbab$

$Ab \Rightarrow Cab \Rightarrow cbab$

$AB \Rightarrow Ab \Rightarrow Cab \Rightarrow cbab$

$S \Rightarrow AB \Rightarrow Ab \Rightarrow Cab \Rightarrow cbab$

Örnek:

G_{S2}= ({A,B,C,D,S}, {0,1}, S, P) ve	P:	1. $S \to C0A$	4. $A \to 0B$	7. $B \to 1C0$	10. $D \to 0C$
		2. $S \to 0C1$	5. $A \to 0$	8. $C \to 0CS$	11. $D \to 0CS$
		3. $S \to 0B1$	6. $B \to A1$	9. $C \to 0D$	olarak veriliyor

$S \Rightarrow 0B1 \Rightarrow 0A11 \Rightarrow 0011$

Soru:

G_{S1}= ({A,B,S}, {0,1}, S, P) ve	P:	1. $S \to 0A$	4. $A \to 0A$	7. $B \to 1B$	
		2. $S \to 1B$	5. $A \to 0S$	8. $B \to 1$	
		3. $S \to B0$	6. $A \to B1$	9. $B \to 0$	olarak veriliyor

Buna göre şu ifadeleri, altlarına "geçerli" veya "geçersiz" yazarak değerlendiriniz. (1 yanlış 1 doğruyu götürür.)

$S \Rightarrow^* \varepsilon$	$(S, 0S) \in \Rightarrow$	$(S, 0S) \in \Rightarrow^*$	$0S \Rightarrow^* 0S$	$0S \Rightarrow^* 00S$	$0S \Rightarrow 0S$	$0S \Rightarrow 00S$
geçersiz	**geçersiz**	**geçersiz**	**geçerli**	**geçersiz**	**geçersiz**	**geçersiz**

Halbuki…	**$(S, 0A) \in \Rightarrow$**	**$(S, 0A) \in \Rightarrow^*$**
	geçerli	**geçerli**

Alıştırım: L(G)= $\{a^n \mid 100 \geq n \geq 1\}$ ise, G= ?