(q, ε) dışında

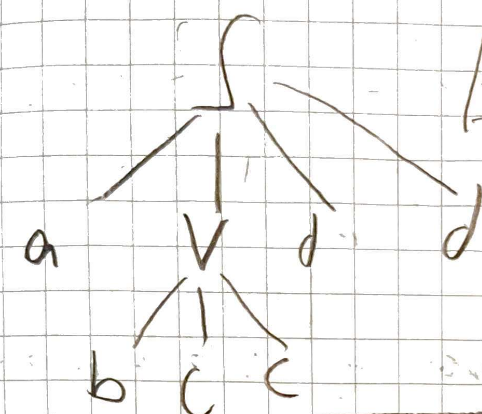
1; oku, stack

...ul edilmez

input gelişi

empty string olamaz ($S \to \epsilon$, $V \to \epsilon$) çünkü $n, m > 0$ diyor.

b)



$\boxed{S \Rightarrow aVdd \Rightarrow abccdd = abc^2d^2}$

$\boxed{\text{Hw6}}$ 1) Consider the CFG $G = (V, \Sigma, R, S)$ where $V = \{S, A, B, C, D, E\}$, $\Sigma = \{a, b, c\}$ and R is given below:

R: $S \to AE \mid EB \mid C$ $\quad A \to aA \mid a$ $\quad B \to Bb \mid b$ $\quad C \to Cc$

$D \to aCb \mid a \mid b \mid c$ $\quad E \to aEb \mid e$

Korhan Erdoğdu

30838

a) Remove all null productions of G, if any, and call the result G_1.

we have $E \to e$ we gonna remove E

$S \to AE \Rightarrow S \to A$ → Just for me to understand.

$S \to EB \Rightarrow S \to B$

$S \to C \Rightarrow S \to C$

New productions: R_1:

$S \to AE|EB|C|A|B$ ✓

$A \to aA|a \quad B \to Bb|b$ ✓

$C \to Cc \quad D \to aCb|a|b|c$

$E \to \cancel{a}E\cancel{b} \Rightarrow ab$

$E \to \epsilon \cancel{\Rightarrow \epsilon}$

↳ we removed the epsilon

$E \to aEb|ab$ ✓

$G_1 = (V, \Sigma, R_1, S)$

b) Remove all the unitary productions of G_1, if any, call the result G_2.

Unit productions: $S \to C$, $S \to A$, $S \to B$

after removing $S \to C$:

① $S \to AE|EB|Cc|A|B$

$A \to aA|a \quad B \to Bb|b$

$C \to Cc \quad D \to aCb|a|b|c$

$E \to aEb|ab$

After removing $S \to A$

② $S \to AE|EB|Cc|aA|a|B$

$A \to aA|a \quad B \to Bb|b$

$C \to Cc \quad D \to aCb|a|b|c$

$E \to aEb|ab$

→

③ After removing $S \to B$

R_2: $S \to AE \mid EB \mid Cc \mid aA \mid a \mid Bb \mid b \qquad A \to aA \mid a$

$B \to Bb \mid b \qquad C \to Cc \qquad D \to aCb \mid a \mid b \mid c \qquad E \to aEb \mid ab$

$$G_2 = (V, \Sigma, R_2, S)$$

c) Remove all non-generative and non-reachable symbols of this grammar if any, call the result G_3. (Non-generative: cannot produce only terminal string) $C \to Cc$) non generative $C \to Cc \mid c$) generative

In this grammar C is non-generating and D is non-reachable. $G_3 = (\{S, A, B, E\}, \{a, b\}, R_3, S)$

R_3: $S \to AE \mid aA \mid a \mid EB \mid Bb \mid b$

$A \to aA \mid a \qquad B \to Bb \mid b \qquad E \to aEb \mid ab$

d) Compute the Chomsky Normal Form of G_3 using your results above.

R_4: $S \to AE \mid XA \mid a \mid EB \mid BY \mid b$

$A \to XA \mid a \qquad B \to BY \mid b \qquad E \to ZY \mid XY$

$X \to a$

$Y \to b$

$Z \to XE$

e) State in the simpliest possible way the language generated by G.

Non-empty string in the form
any number of a's followed by any number of b's

$G = \{ a^n b^m \mid n+m > 0 \}$

---

2) Consider the alphabet T of terminals consisting of 3 pairs of matching left and right parentheses of three types, namely: { , } , [,] , (,)

a) Describe a CFG, $G = (V, T, R, S)$ such that L_G has the following properties:

(i) every left parenthesis is balanced by a distinct right parenthesis somewhere on its right side and of its own type

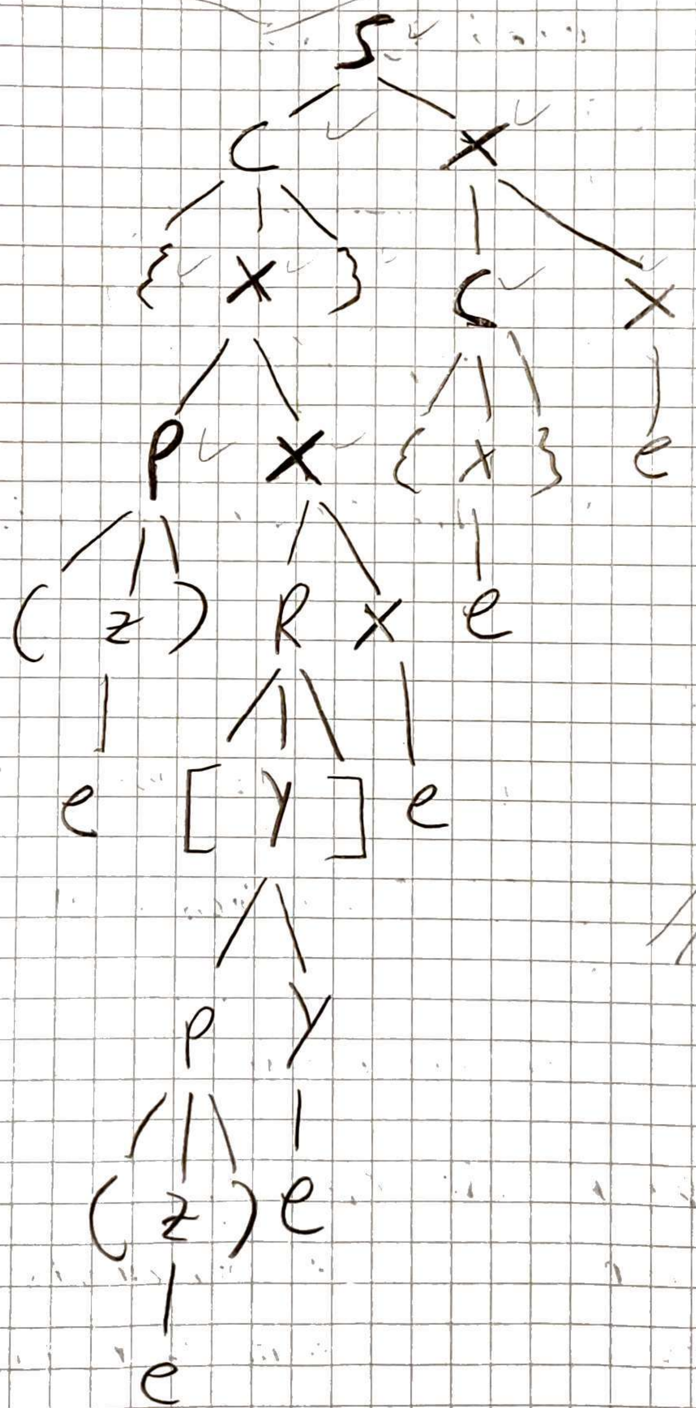
(ii) no curly parenthesis i.e { } is contained within a rectangular pair, i.e [ ] or a plain pair i.e ( ). And no rectangular parenthesis is contained within a plain pair.

(iii) empty string is not a member of L_G.

R: $S \to PX \mid RX \mid CX$ | P → düz parantez ( )
$X \to e \mid PX \mid RX \mid CX$ | R → köşeli parantez [ ]
$Y \to e \mid PY \mid RY$ | C → süslü parantez { }
$Z \to e \mid PZ$ | $G = (V, T, R, S)$
$C \to \{X\}$ | $T = \{ \{,\}, [,], (,) \}$
$R \to [Y]$ $P \to (Z)$ | $V = \{ S, X, Y, Z, C, R, P \}$

2 farklı parse tree → ambiguous

b) Using your grammar find a parse tree that derives the String: {()[()]}{}



$\boxed{6.4.2}$ Give DPDA to accept the following languages:

a) $\{0^n 1^m \mid n \leq m\}$ Sıfırların sayısı 1'lerden fazla olamaz.

$P = (Q, \Sigma, T, \delta, q_0, z_0, F)$ $Q = \{q_0, q_1, q_2, q_3\}$

δ: $(q_0, 0, z_0) \rightarrow (q_1, z_0)$ ✓ $\Sigma = \{0, 1\}$

$(q_0, 1, z_0) \rightarrow (q_3, z_0)$ ✓ → we accept because 1's already higher than 0's. $T = \{0, z_0\}$

$F = \{q_0, q_3\}$

$\delta(q_1, 0, z_0) \rightarrow (q_1, 0z_0)$ ✓
$\delta(q_1, 0, 0) \rightarrow (q_1, 00)$ ✓

For every 0's we push zero to the stack.

$\delta(q_1, 1, 0) \rightarrow (q_2, e)$ ✓ → pop 0's when 1's come.

$\delta(q_1, 1, z_0) \rightarrow (q_3, z_0)$ ✓ → when 1 comes and we see every 0's are popped, we understand there is an extra 1 came. So we accept it.

$\delta(q_2, 1, 0) \rightarrow (q_2, e)$ ✓ → pop 0's when 1's come

$\delta(q_2, 1, z_0) \rightarrow (q_3, z_0)$ ✓

$\delta(q_3, 1, z_0) \rightarrow (q_3, z_0)$ ✓

1'lerin sayısı, sıfırdan fazla olmayacak

b) $\{0^n 1^m \mid n \geq m\}$ $P = (Q, \Sigma, T, \delta, q_0, z_0, F)$

δ: $\delta(q_0, 0, z_0) \rightarrow (q_0, 0z_0)$ $Q = \{q_0, q_1\}$ $T = \{0, z_0\}$

↳ we push '0' to the stack every time we read 0's. We go final state because at that point 0's are higher than 1's, so we accept. $F = \{q_0, q_1\}$

$\delta(q_0, 0, 0) \rightarrow (q_0, 00)$ → for every 0's, we push "0" to the stack.

$\delta(q_0, 1, 0) \rightarrow (q_1, e)$

$\delta(q_1, 1, 0) \rightarrow (q_1, e)$

→ for every 1's we pop 0's.

★ Bir PDA'nın reddetmesi için 2 yol var

1) Tanımlı geçiş, transition yok

2) Tanımlı geçiş var ama red durumuna gidiyor

işte ya reddedici state'e gönderen ya da hiç yazmıyorum.

ikisi de olur fakat DPDA tasarlarken genelde geçiş tanımlamadan reddetme tercih edilebilir. Zorunlu değildir iki yöntem de kullanılabilir. Aynı anda da kullanılabilir.

$\delta(q_0, 1, z_0)$ böyle bir şey tanımlamıyoruz bu yüzden bunu reddetmek gerektir. Çünkü demek ki

$\delta(q_1, 1, z_0)$ tüm 1'ler sıfırları poplamış ki z_0 kalmış ama bizim dilimiz 000011 gibi bir şey. Tüm 1'ler sıfırları poplayamaz.

00111

n ve m herhangi bir sayı ama ilk ve son sıfır sayıları eşit olacak.

(1) $\{0^n 1^m 0^n \mid n \text{ and } m \text{ are arbitrary}\}$

$P = (Q, \Sigma, T, \delta, q_0, z_0, F)$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$

$\Sigma = \{0, 1\}$ $T = \{0, z_0\}$

$F = \{q_0, q_3, q_5\}$

1'den önceki sıfırları stacke pushla, 1 gelince stacki elleme. 1'den sonraki her sıfır ile stackten sıfırları popla.

$\delta(q_0, 0, z_0) \to (q_1, 0z_0)$ ✓

$\delta(q_0, 0, 0) \to (q_1, 00)$ ✓

$\delta(q_0, 1, z_0) \to (q_3, z_0)$ ✓ → $n = 0$ we accept it.

$\delta(q_0, 1, 0) \to (q_2, 0)$ ✓

$\delta(q_1, 0, 0) \to (q_1, 00)$

$\delta(q_1, 1, 0) \to (q_2, 0)$ ✓

$\delta(q_2, 1, 0) \to (q_2, 0)$ ✓

$\delta(q_2, 0, 0) \to (q_4, \epsilon)$ ✓

$\delta(q_3, 1, z_0) \to (q_3, z_0)$ ✓

$\delta(q_4, 0, 0) \to (q_4, \epsilon)$ ✓

$\delta(q_4, \epsilon, z_0) \to (q_5, z_0)$ ✓

Durum	görev
q_0	0'ları mı 1'leri mi okuyacağına karar ver.
q_1	ilk 0'ları stack'e pushet.
q_2	1'leri oku, stack'i sabit tut.
q_3	hiç "0" yok, "1"leri geçir
q_4	son "0"ları stackten pop et.
q_5	kabul durumu.