

BIL 463 Derleyici Tasarımı
2025 Güz Dönemi
Vize Sınavı

- Fotoğraflı üniversite kimliği olmayan öğrenciler sınava giremez.
- Her türlü kağıt, kitap, not, hesap makinesi, telefon, saat vb. elektronik aygıtları kullanımı **kopya** sayılır.
- **TOPLAM SÜRE 90 DAKİKADIR.**
- Toplam 100 puan üzerinden değerlendirme yapılacaktır.
- Cevaplarınızı okunaklı bir şekilde ayrılan kısımlara yazınız.

Question	Points	Score
1	5	
2	10	
3	5	
4	5	
5	5	
6	5	
7	8	
8	15	
9	5	
10	6	
11	5	
12	5	
13	5	
14	5	
15	5	
16	5	
17	5	
18	5	
Total:	109	

İsim ve Soyisim : _____

Öğrenci No : _____

İmza : _____

1. **[5 puan] (İngilizce)** Which regular expression matches one or more digits?
(Türkçe) Aşağıdaki regular expression'lardan hangisi bir veya daha fazla rakamı eşleştirir?
- [0-9]
 - [0-9]*
 - [0-9]+
 - [0-9]?
2. **[10 puan] (İngilizce)** Write a regular expression and draw an NFA that recognizes the language: strings over $\{a,b\}$ that end with "abb"
(Türkçe) Aşağıdaki dili tanıyan bir Regular Expression yazınız ve NFA çiziniz: $\{a,b\}$ alfabesi üzerinde "abb" ile biten string'ler
3. **[5 puan] (İngilizce)** Every NFA can be converted into an equivalent DFA.
(Türkçe) Her NFA eşdeğer bir DFA'ya dönüştürülebilir.
- True (Doğru)**
 - False (Yanlış)**
4. **[5 puan] (İngilizce)** Convert the following NFA to DFA:
(Türkçe) Aşağıdaki NFA'yı DFA'ya dönüştürünüz:
-
- ```

graph LR
 start((start)) --> q0((q0))
 q0 -- a --> q1((q1))
 q1 -- b --> q2(((q2)))
 q2 -- "a,b" --> q2

```
5. **[5 puan] (İngilizce)** What is the "maximal munch" rule in lexical analysis?  
**(Türkçe)** Lexical analizde "en büyük lokma" kuralı nedir?
- İlk bulunan token seçilir
  - En kısa eşleşen token seçilir
  - Mümkün olan en uzun token seçilir

- D. Boşluk karakterleri atlanır
6. [5 puan] (İngilizce) What does LL(1) stand for in parsing? (Türkçe) LL(1) parsing'teki LL(1) ne anlama gelir?

6. \_\_\_\_\_ **Left-to-right scan, Leftmost derivation, 1 token lookahead** \_\_\_\_\_

7. (İngilizce) Compare Top-Down vs Bottom-Up parsing in terms of:  
(Türkçe) Top-Down ve Bottom-Up parsing'i aşağıdaki açılardan karşılaştırınız:
- (a) [2 puan] (İngilizce) Derivation order (Türkçe) Türetim sırası  
(a) \_\_\_\_\_ **Top-Down: Leftmost; Bottom-Up: Rightmost in reverse** \_\_\_\_\_
- (b) [2 puan] (İngilizce) Left recursion handling (Türkçe) Left recursion yönetimi  
(b) \_\_\_\_\_ **Top-Down: Must eliminate; Bottom-Up: Can handle directly** \_\_\_\_\_
- (c) [2 puan] (İngilizce) Grammar restrictions (Türkçe) Grammar kısıtlamaları  
(c) \_\_\_\_\_ **Top-Down: More restrictive (LL); Bottom-Up: Less restrictive (LR)** \_\_\_\_\_
- (d) [2 puan] (İngilizce) Implementation approach (Türkçe) Implementasyon yaklaşımı  
(d) \_\_\_\_\_ **Top-Down: Recursive descent; Bottom-Up: Shift-reduce with stack** \_\_\_\_\_
8. (İngilizce) Given the grammar:  $S \rightarrow aSb|\epsilon$   
(Türkçe) Verilen grammar:  $S \rightarrow aSb|\epsilon$
- (a) [5 puan] What language does this generate? (Bu grammar hangi dili üretir?)  
$$\overline{a^n b^n}$$
- (b) [5 puan] Give a leftmost derivation for "aabb" ("aabb" için leftmost türetim veriniz)  
$$S \rightarrow aSb \rightarrow aaSbb \rightarrow aaebb \rightarrow aabb$$
- (c) [5 puan] Give the parse tree for "aabb" ("aabb" için parse tree çiziniz)

9. [5 puan] (İngilizce) Eliminate left recursion from:  $E \rightarrow E + T \mid T$   
(Türkçe) Aşağıdaki grammar'den left recursion'ı kaldırınız:  
 $E \rightarrow E + T \mid T$

9. \_\_\_\_\_  $E \rightarrow TE'$  \_\_\_\_\_  $E' \rightarrow +TE'|\epsilon$  \_\_\_\_\_

10. [6 puan] (İngilizce) Compute FIRST sets for:(Türkçe) Aşağıdaki grammar için FIRST kümelerini hesaplayınız:  
 $S \rightarrow AB$   
 $A \rightarrow aA|\epsilon$   
 $B \rightarrow bB|c$

FIRST(S): a, b, c    FIRST(A): a, \_\_\_\_\_    FIRST(B): b, c \_\_\_\_\_

11. [5 puan] (**İngilizce**) An ambiguous grammar is one that has more than one parse tree for the same string. (**Türkçe**) Belirsiz bir grammar, aynı string için birden fazla parse tree'ya sahip olmalıdır.  
A. True (Doğru)    B. False (Yanlış)
12. [5 puan] (**İngilizce**) What is the main problem with left recursion in top-down parsing?  
(**İngilizce**) Top-down parsing'te left recursion'in ana problemi nedir?

12. **Causes infinite loop - keeps expanding same non-terminal without consuming input**

13. [5 puan] (**İngilizce**) The FOLLOW set of a non-terminal includes terminals that can appear immediately after the non-terminal in some sentential form.  
(**Türkçe**) Bir non-terminal'in FOLLOW kümesi, bazı sentential form'larda non-terminal'i hemen takip edebilen terminal'leri içerir.  
A. True (Doğru)    B. False (Yanlış)
14. [5 puan] (**İngilizce**) An LL(1) parser uses:  
(**Türkçe**) Bir LL(1) parser şunu kullanır:  
A. A stack and a parsing table (Bir stack ve parsing tablosu)  
B. A queue and a parsing table (Bir kuyruk ve parsing tablosu)  
C. A stack and a binary tree (Bir stack ve ikili ağaç)  
D. A queue and a binary tree (Bir kuyruk ve ikili ağaç)
15. [5 puan] (**İngilizce**) What are the two main actions in a shift-reduce parser?  
(**Türkçe**) Bir shift-reduce parser'da iki ana aksiyon nedir?

15. \_\_\_\_\_ **Shift and Reduce**

16. [5 puan] (**İngilizce**) In an LR(0) automaton, an item of the form  $A \rightarrow \alpha.\beta$  indicates:  
(**Türkçe**) Bir LR(0) otomatında,  $A \rightarrow \alpha.\beta$  formundaki bir item şunu belirtir:  
A. We have already matched  $\beta$  and are looking for  $\alpha$  ( $\beta$  zaten eşleştirildi ve  $\alpha$  aranıyor)  
B. We have matched  $\alpha$  and are looking to match  $\beta$  ( $\alpha$  eşleştirildi ve  $\beta$  eşleştirilmeye çalışılıyor)  
C. We have matched  $\beta$  and are looking to match  $\alpha$  ( $\beta$  eşleştirildi ve  $\alpha$  eşleştirilmeye çalışılıyor)  
D. We have matched  $\alpha$  and  $\beta$  (Hem  $\alpha$  hem de  $\beta$  eşleştirildi)
17. [5 puan] (**İngilizce**) SLR parsing uses FOLLOW sets to resolve conflicts in the LR(0) automaton.  
(**Türkçe**) SLR parsing, LR(0) otomatındaki çakışmaları çözmek için FOLLOW kümelerini kullanır.  
A. True (Doğru)    B. False (Yanlış)
18. [5 puan] (**İngilizce**) What is a handle in bottom-up parsing ( shift-reduce parsing)?  
(**Türkçe**) Bottom-up parsing ( shift-reduce parsing)'te handle nedir?

18. **Substring matching RHS of production whose reduction is a step in rightmost derivation**