Eastern Mediterranean University

Department of Computer Engineering

**CMxE 318: In-Class Practice**

**March 27th, 2025**

**Instructor:**   **Behnam Bojnordi Arbab**

***Name/Surname | Student #***

**Student 1:**

**Student 2:**

**Student 3:**

**Student 4:**

**Student 5:**

**Student 6:**

# *Instructions to Students::*

* ***Discuss the tasks within your groups.***
* ***Use*** [***slides***](https://staff.emu.edu.tr/behnamarbab/en/teaching/cmpe318/lecture-notes) ***provided, textbooks, credible internet sources, or any other resources (such as*** [***ChatGPT***](https://chatgpt.com/g/g-67e4682cabfc819195163c67ba85dae9-cmxe-mentor)***) to assist your answers.***
* ***Write down your group's answers clearly.***
* ***100 Minutes to submit your answers on Teams.***
* ***You may submit both the Digital and/or On-paper (scanned) answers together as a .zip file.***
* ***Any other questions? Do it anyhow you want!***

## PART I : Questions from previous Midterms (2018 – 2019 Spring Semester)

1. Give the internal representation of the Lisp list “**(A (B C) D)**”.
2. We are given the following grammar.

**E → T** **T F**

**T → T a | b**

**F → c F | d**

Give a **rightmost** derivation for the string “**b a b d**”.

1. Show that the following grammar is ambiguous.

**E → E \* T | T**

**T → 2 | 3 | E**

1. Give an equivalent grammar in **BNF** to the following grammar in **EBNF**.

**S →** **{ (b |** **e )** **g } [h]**

1. Eliminate left recursion from the grammar given below (give an equivalent grammar without left recursion).

**X → X a b | c | X d | X e**

1. We are given a context free grammar and its **LR** parsing tables below:

**1. E → E + T**

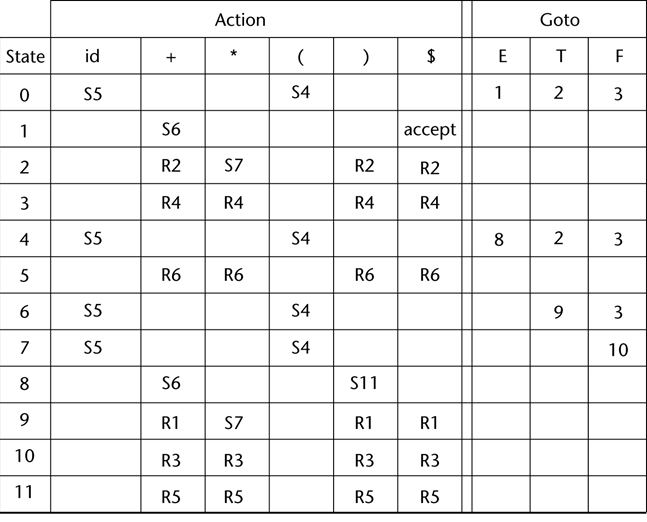
**2. E → T**

**3. T → T \* F**

**4. T → F**

**5. F →** **( E )**

**6. F → id**



Give the configurations of the **LR parser** for the input “**( (** **id ) ) $**”.

## PART II: Essential questions

### Question 7: Programming Paradigms

* **What are the main differences between imperative, functional, and logic programming paradigms?**

* Provide a clear definition and one example language for each paradigm.

* Explain briefly how each of these languages would approach solving the same simple task:

"Calculating the factorial of a number."

### Question 8: Parsing Techniques

* Explain clearly the difference between **Top-Down** (Recursive Descent) Parsing and **Bottom-Up** (LR) Parsing techniques.
* For the given grammar, demonstrate briefly how each parser type would start analyzing the input:

**Grammar:**

**S → aSb | ab**

**Input:**

**aaabbb**

* Which parsing technique do you think is easier to implement by hand and why?

### Question 9: Language Implementation

* What are the major differences between a **Compiler**, an **Interpreter**, and a **Just-In-Time** (JIT) compilation system?

* For each type, name one popular programming language implementation that uses this approach.

* Briefly discuss one advantage and one disadvantage of each implementation method.