SSN College of Engineering

Department of Information Technology

UIT2401 — Automata Theory and Compiler Design

Assignment 2

2024 – 2025

- This Assignment to be submitted on or before 24.03.2025
- Submit Handwritten Notes, Credits will be awarded based on **originality and understanding**

Question 1: With the given CFG,

$$S \rightarrow S + T | T$$
$$T \rightarrow T * F | F$$
$$F \rightarrow (S) | id$$

Construct LL parsing tree and validate the following:

(a) id + id * id

(b) (id)

Question 2: Construct the parsing LL(1) table for the below grammar and parse the string id + id * id

$$G \rightarrow SG'$$
$$G' \rightarrow +SG' | \epsilon$$
$$S \rightarrow FS'$$
$$S' \rightarrow *FS' | \epsilon$$
$$F \rightarrow id | (G)$$

Question 3: Convert CFG to CNF & GNF:

(a)

$$S \rightarrow AX | b$$
$$A \rightarrow aA | a$$
$$X \rightarrow bX | \epsilon$$

(b)

$$S \rightarrow aA|aB$$
$$A \rightarrow aaA|\epsilon$$
$$B \rightarrow bB|bbC$$
$$C \rightarrow B$$

(c)

$$S \rightarrow AB|BB|BA|Z$$
$$A \rightarrow aA|b$$
$$B \rightarrow bB|a$$
$$P \rightarrow ad|D$$

(d)

$$S \rightarrow aSB|aA|AB$$
$$A \rightarrow aA|\epsilon$$
$$B \rightarrow bB|\epsilon$$

Question 4: Recursive Descent Algorithm:

$$E \rightarrow TE^{'}$$
$$E^{'} \rightarrow +TE^{'}|\epsilon$$
$$T \rightarrow FT^{'}$$
$$T^{'} \rightarrow *FT^{'}|\epsilon$$
$$F \rightarrow (E)|id$$

Question 5: Construct an LALR(1) Parsing table for the grammar.

$$S \rightarrow Aa \mid bAc \mid dc \mid bda$$
$$A \rightarrow d$$

Parse the input string "bdc" using the LALR parsing table.

Question 6: Consider the following grammar for conditional statements:

$$S \rightarrow \text{if } E \text{ then } S \text{ else } S$$
$$S \rightarrow \text{id} = \text{num}$$
$$E \rightarrow \text{id} < \text{num}$$
$$E \rightarrow \text{id} > \text{num}$$
$$E \rightarrow \text{id} == \text{num}$$

Perform shift-reduce parsing on the input string:

$$\text{if } x < 10 \text{ then } y = 5 \text{ else } y = 20$$

Question 7: Consider the language of all strings of the language $0^n 1^n | \, where \, m, n \geq 0$
Construct a Turing Machine to validate such strings.

Question 8: Construct a Push down Automata to accept the following languages

(a) Consider the language of all strings of the form

$$\{0^n 1^{m+n} 1^n | \, where \, m, n \geq 0\}$$

(b)

$$\{W = 0, 1^* | n_0 < 2 * n_1\}$$

Question 9: Explain Operator precedence Parser with an Example

Question 10: How do you use SDT for type checking. Explain with example

Question 11: Consider a syntax-directed translation (SDT) scheme for converting an infix arithmetic expression into a postfix expression using the following grammar:

E → E + T print('+')
E → E - T print('-')
E → T
T → T * F print('*')
T → T / F print('/')
T → F
F → (E)
F → id print(id.lexeme)

Perform syntax-directed translation for the input: (a + b) * (c - d) / e

Give your comment the translation

Question 12: Describe about code generation and challenges associated with it. Explain the various code optimization techniques in detail.