

The Open University of Sri Lanka
Faculty of Engineering Technology
Department of Electrical and Computer Engineering



Study Programme : Bachelor of Software Engineering Honours
Name of the Examination : Final Examination
Course Code and Title : EEX6363 – Compiler Construction
Academic Year : 2024/25
Date : 14th February 2026
Time : 09:30 – 12:30 h
Duration : 3 hours

General instructions

1. Read all instructions carefully before answering the questions.
 2. This question paper consists of **Three (3)** questions in **Three (3)** pages.
 3. Answer **all** questions.
 4. Clearly state your assumptions, if any.
 5. Answer for each question should commence from a new page.
 6. Answers should be in clear handwriting and do not use Red colour pen.
 7. This is a Closed Book Test (**CBT**).
-

Q1 [60 marks]

An embedded sensor control domain-specific language (DSL) is a specialized programming language to manage industrial sensors in IoT systems. These DSLs bridge the gap between high-level operational requirements and low-level embedded hardware. The context free grammar (CFG) for a DSL is developed as follows:

$$\begin{aligned}
 \text{program} &\rightarrow \text{device id } \{ \text{stmtlist} \} \\
 \text{stmtlist} &\rightarrow \text{stmt stmtlist} \mid \varepsilon \\
 \text{stmt} &\rightarrow \text{set param} = \text{num} ; \mid \text{activate} ; \\
 \text{param} &\rightarrow \text{temp_threshold} \mid \text{sample_rate}
 \end{aligned}$$

where *program* is the start term and ε denotes the empty string.

This DSL supports the following features:

Keywords	<i>device, set, activate</i>
Identifiers (only alphanumeric)	<i>id</i>
Parameters	<i>temp_threshold, sample_rate</i>
Positive integers (except 0)	<i>num</i>
Assignment operator	<i>=</i>
Block braces	<i>{ }</i>
Statement terminator	<i>;</i>

(a) Define token specifications using regular expressions. [06]

(b) Tokenize the following input stream: [08]

```

device Sensor1 {
    set sample_rate = 1000;
    set temp_threshold = 75;
    activate;
}

```

(c) Define the grammar for the given DSL by clearly stating the relevant set of parameters. [08]

(d) Validate the above input stream by using the given CFG. [08]

(e) Draw the complete parser tree that the parser would build for the above input stream. [12]

(f) State whether the grammar is suitable for LL(1) parsing. Justify briefly. [05]

(g) Identify two types of semantic errors that may be occurred in this language. [04]

(h) Generate Intermediate Code (three Address-Code) for: [04]

```

set sample_rate = 1000;

```

(i) Write LEX implementation syntax for the given DSL. [05]

Q2 [20 marks]

Consider the following context-free grammar with x and y are terminals, all others are non-terminals, including the start symbol, S .

$$\begin{aligned} S &\rightarrow AB \\ A &\rightarrow Sy \mid x \\ B &\rightarrow yS \mid y \end{aligned}$$

Answer the following:

- Find the FIRST and FOLLOW sets of the non-terminals. [02]
- Draw the DFA of LR(0)-items (numbering the states for later reference). [14]
- This grammar is not SLR(1). Explain and identify corresponding conflicts in terms of in which state(s) they occur and what conflicting reactions occur under which input. [04]

Q3 [20 marks]

Consider the following grammar G1:

$$S \rightarrow a \mid S\#S \mid S@S$$

Where, S is the start symbol and all others (including $\$$ which represents the end of the sequence of tokens) are terminals.

- Give a concrete argument why the grammar is ambiguous. [04]
- Assume that
 - the operator $\#$ has *low* precedence and is *right-associative*
 - the operator $@$ has *high* precedence and is *left-associative*

Give a new grammar G2 which describes the same language as G1 and follows the rules just given. You may introduce new non-terminals. [07]

- Now, consider the grammars G1 and G2, as well as the following grammar G3:

$$S \rightarrow a \mid S\#S \mid S@S \mid +S+$$

where the symbol $+$ is new terminal.

- Which of the languages $L(G1)$, $L(G2)$, and $L(G3)$, are *regular* and which not, briefly explain. [06]
- Give a regular expression for those which languages which happen to be regular. [03]