**Bahria University, Lahore Campus**

Department of Computer Science

Lab Journal 11-12

**(Spring 2024)**

|  |  |  |
| --- | --- | --- |
| Course: | **Compiler Construction – Lab** | Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Course Code: | CSL 323 | Max Marks: 10 |
| Faculty’s Name: | Mr. M Mudassar |  |

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Enroll No: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Class: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Objective(s):

Upon completion of this lab session, learners will be able to:

* Design a compiler for lexical analyzer for any specified language.

## Question – 1: HTML-like Language

Consider the HTML-like language defined by the rules below. Draw its Finite Automaton and Transition Table.**Tags**:Tags in this language start with a < symbol and end with a > symbol.A tag name must start with an alphabetic character (a-z, A-Z) and can be followed by any combination of alphabetic characters, digits (0-9), and hyphens (-).Tags can optionally include attributes, which are key-value pairs. An attribute key starts with an alphabetic character and can include alphabetic characters, digits, hyphens, and underscores (\_). The key is followed by an equals sign (=) and a quoted value (either single or double quotes).

**Attributes**:An attribute within a tag follows the format key="value" or key='value'.Attribute values can contain any characters except the quote character that encloses the value.

**Self-closing Tags**: Tags can be self-closing by ending with /> instead of >.

**Text Content**:Text content is any sequence of characters that are not part of a tag. Text can include spaces, digits, symbols, and alphabets.

**Comments**:Comments start with <!-- and end with -->. Comments can span multiple lines.Whitespace and Delimiters:Whitespace characters (spaces, tabs, newlines) are used to separate tokens but are not part of any token.

**Your task is to**:

Draw the Finite Automaton for the HTML-like language rules described above.Create the Transition Table for the Finite Automaton you designed, detailing the state transitions for each type of character encountered.

## Question – 2: HTML-like Language

Design the Syntax analyzer considering the Context Free Grammar for the language as given below:

S -> Tag S' | Text S' | Comment S' | εS' -> Tag S' | Text S' | Comment S' | εTag -> '<' TagName AttributeList TagEndTagEnd -> '>' | '/>'TagName -> [a-zA-Z] [a-zA-Z0-9-]\*AttributeList -> Attribute AttributeList' | εAttributeList' -> Attribute AttributeList' | εAttribute -> AttributeKey '=' AttributeValueAttributeKey -> [a-zA-Z] [a-zA-Z0-9-\_]\*AttributeValue -> '"' [^"]\* '"' | '\'' [^\']\* '\''SelfClosingTag -> TagName AttributeList '/>'Text -> char Text' | εText' -> char Text' | εComment -> '<!--' CommentContent '-->'CommentContent -> char CommentContent' | εCommentContent' -> char CommentContent' | ε

**Lab Grading Sheet :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Max Marks** | **Obtained Marks** | **Comments(*if any*)** |
| 1. | 10 |  |  |
| 2. | 10 |  |  |
|  |  |  |  |
|  |  |  |  |
| **Total** | **10** |  | **Signature** |

**Note: Attempt all tasks and get them checked by your Lab Instructor.**