WEEKLY-EXERCISE - 04

ICS 365-51   **Metropolitan State University/MN**

Week 4   **Due <u>11:59pm, Sunday, Sept. 18, 2022</u>**   **Fall 2022**

Name:_____Pong Lee_____   **Score: _____**

Please complete both Parts I and II and then upload the results to D2L under the dropbox for Weekly Exercise 04 before the deadline (total 20 points).

Part I: Based on the discussion in Lecture 4, please either **bold** or <mark>highlight</mark> your answers below, only one answer per question. (1 point each, total 10 points)

1. Based on the discussion in Chapter 4, which of the following is not one of the advantages for using BNF to describe syntax?
    A) The parser can be based directly on the BNF;
    <mark>B) BNF was introduced by John Backus and Peter Naur;</mark>
    C) It provides a clear and concise syntax description
    D) Parsers based on BNF are easy to maintain.

2. All of the followings are the reasons to separate lexical and syntax analysis except?
    A) Less complex approaches can be used for lexical analysis; separating them simplifies the parser;
    B) Parts of the lexical analyzer may not be portable, but the parser always is portable;
    C) The separation allows optimization of the lexical analyzer;
    <mark>D) It was recommended by IBM.</mark>

3. Based on the discussion in Chapter 4, which of the following is not an approach to building a lexical analyzer?
    <mark>A) Use a recursive-descent parser;</mark>
    B) Write a program that implements the state diagram that describes the tokens;
    C) Hand-construct a table-driven implementation of the state diagram that describes the tokens;
    D) Use a software tool that constructs a table-driven lexical analyzer from a formal description.

4. Given a grammar below
    ```
    A -> 0
    A -> 01A
    A -> 1AA
    ```
    Which of the following statements is true?

    A) The length of the bit strings produced by this grammar will always be 3
    <mark>B) The length of the bit strings produced by this grammar will always be an even number</mark>
    C) The length of the bit strings produced by this grammar will always be an odd number
    D) The length of the bit strings produced by this grammar can be either an even or odd number

5. Based on the discussion in Chapter 4, which of the following statements is not true?

    A) The lexical analyzer is usually a function that is called by the parser when it needs the next token;
    B) Useful parsers look only one token ahead in the input;
    C) A recursive-descent parser is an LL parser;
    <mark>D) The most common bottom-up parsing algorithms are in the LL family.</mark>

6. Based on the discussion in Chapter 4, which of the following is not one of the advantages of LR parsers?

A) Work on a larger class of grammars than other bottom-up algorithms, but are as efficient as any other bottom-up parser;

B) Produce the parse tree, beginning at the root;

C) Can detect syntax errors as soon as it is possible;

D) Work for nearly all grammars that describe programming languages.

7. The lexical analyzer can be recognized as

A) a "front-end" for the parser

B) a function that is called by the parser when it needs the next token;

C) a pattern matcher for character strings;

D) All of above.

8. Which of the following strings cannot be accepted by the state diagram shown in Figure 4.1 on page 167 of the textbook?

A) `abc123`

B) `a1b2c3`

C) `1a2b3c`

D) `ABC123`

9. Based on the discussion in Chapter 4, a *lexical analyzer* can be recognized as

A) a push-down automaton based on BNF;

B) a finite automaton based on BNF;

C) a push-down automaton based on a context-free grammar;

D) a finite automaton based on a regular grammar.

10. Based on the discussion in Chapter 4, why is EBNF ideally suited for being the basis for a recursive-descent parser?

A) Because EBNF minimizes the number of nonterminals

B) Because EBNF can be implemented in C

C) Because EBNF was introduced before BNF

D) Because EBNF can be implemented in C

Part II: Please study the discussion in class as well as covered in Chapter 4 of the textbook to complete the following tasks: (Total 10 points)

1 Given the following finite state machine (FSM)
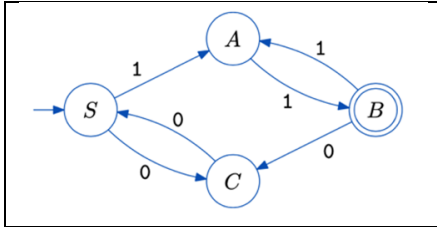
and determine whether each of the following binary strings is acceptable by this FSM. (1 point each, total 5 points)

```
1.1)    000111      Yes __      No __
1.2)    111000      Yes __      No __
1.3)    001111      Yes __      No __
1.4)    110011      Yes __      No __
1.5)    000011      Yes __      No __
```

2 Please convert the FSM shown above to its corresponding regular grammar. (5 points)

$S \rightarrow$ 1A|0C
$A \rightarrow$ 1B
$B \rightarrow$ 1A|0C
$C \rightarrow$ 0S|