Programming Languages - Lab 2: Grammars, Parse Trees, & Automata

Name: YourName

1. Describe the language denoted by the regular expression: 0(0|1)\*0

1. Write a regular expression for the language that includes all strings of 0’s and 1’s with an even number of 0’s followed by an odd number of 1’s.
2. Consider the following grammar consisting of terminals {-, /, (, ), id}, the set of non-terminals {E, T, F}, the start symbol E, and the following rules:

E → T  
E → E – T  
T → F  
T → T / F  
F → id

F → (E)

* 1. Construct a leftmost derivation for the same string.
  2. Construct a rightmost derivation for the same string.
  3. Construct a parse tree for the string: id – id / id

1. Replace the productions E → E – T and T → T / F from the grammar described in Problem 3 with the productions E → E / T and T → T – F respectively, so that the resulting rules are as follows:

E → T  
E → E **/** T  
T → F  
T → T **–** F  
F → id

F → (E)

* 1. Construct a new parse tree for the string: id – id / id
  2. What is the difference? Is the new grammar ambiguous? Defend.

1. Consider the CFG consisting of terminals {+, (, ), id}, non-terminals {E, T}, start symbol E, and production rules:

E → E + T | T  
T → (E) | id

Give the rightmost and leftmost derivations for id + (id + id). Is this grammar ambiguous? Defend!

Questions 6 – 9 refer to the following context free grammar:

A → B & A | B

B → B @ C | C

C → C \* x | x | (A)

1. Indicate True or False for each of the following statements.
   1. The & operator has higher precedence than the @ operator.
   2. The & operator has higher precedence than the \* operator.
   3. The & operator associates to the left.
   4. The \* operator associates to the left.
   5. The grammar is ambiguous.

*Hint*: binary + is left associative: id + id + id => (id + id) + id Also, try drawing the parse tree for id + id + id

1. The abstract syntax tree for x\*x&x@x@x is the same as which of the following?
   1. (((x\*x)&x)@x)@x
   2. (x\*x)&((x@x)@x)
   3. (x\*x)&(x@(x@x))
   4. x\*((x&x)@(x@x))
2. Which of the following is a rightmost derivation? Discussed in pages 34-40 of Mead.
   1. A → B&A → B&B → B&B@C → B&B@x → B&C@x → B&x@x → C&x@x → x&x@x
   2. A → B&A → C&A → C&B → C&B@C → C&C@C → C&C@x → C&x@x → x&x@x
   3. A → B&A → B&B → C&B → C&B@C → x&B@C → x&C@C → x&x@C → x&x@x
   4. A → B&A → C&A → x&A → x&B → x&B@C → x&C@C → x&x@C → x&x@x
3. All derivations in the previous question correspond to the same parse tree. Draw that parse tree.
4. Provide a regular expression for the regular grammars (a and b), a regular grammar *or* an FSA for the regular expression (c), and either a regular expression or a regular grammar for the FSAs (d and e).
   1. S → aA

A → bB

B → aA | a

* 1. S → aA

A → bS |

* 1. (a | b)\*

a

b







a

a

a

a

b

b

1. Activity 1 on p. 37 of the Mead book.
2. Activity 2 on p. 38 of the Mead book.