**CSE 340 Spring B 2022**

**HOMEWORK 1**

**Claudio B Rodriguez Rodriguez**

3/28/2022

**Problem 1 (20 points). Consider the following regular expressions (we omit the dot operator)**

R0 = 1|2|3|4|5|6|7|8|9

R1 = 0|1|2|3|4|5|6|7|8|9

R2 = (0|1)\* R0 (0|1) R3 = 00 R0\*(0|1)\*

R4 = R3\* R2\* 000

Assume that the longest prefix-matching rule is used. Assume that ties are broken in favor of the regular expression listed first in the list.

1. Give an example of input for which getToken() returns R0

Any in the set {1,2,3,4,5,6,7,8,9}, e.g., 1

2. Give an example of input for which getToken() returns R1

Any in the set {0,1,2,3,4,5,6,7,8,9}, e.g., 0

3. Give an example of input for which getToken() returns R2

Starts with zero or more of {0,1}, then one token from 1, then a single {0,1}. E.g., 0070

4. Give an example of input for which getToken() returns R3

00900

5. Give an example of input for which getToken() returns R4

00900030000

6. If getToken() if called repeatedly on the following input, what is the sequence of tokens returned?

99001101678100010101030123457000010

**Explain your answers by showing the step by step table.**

|  |  |  |  |
| --- | --- | --- | --- |
| String | Matching | Potential | Longest Match |
| \_99001101678100010101030123457000010 ^ |  | All |  |
| 99001101678100010101030123457000010 ^ |  | R0, R1 |  |
| 99001101678100010101030123457000010  ^ |  | R0, R1 |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |
| 99001101678100010101030123457000010  ^ |  |  |  |

**Problem 2 (10 points).** Consider the grammar

S → AB

A → aB | 𝜀

B → bB | abA | A

1. Show that this grammar is ambiguous by constructing two different leftmost derivations for the sentence abab

|  |  |
| --- | --- |
| S | S → AB |
| A B ^ | A → ab |
| a b B  ^ | B → a b A |
| a b a b A  ^ | A → 𝜀 |
| a b a b |  |

|  |  |
| --- | --- |
| S |  |
| A B  ^ | A → 𝜀 |
| B  ^ | B → abA |
| a b A  ^ | A → aB |
| a b a B  ^ | B → bB |
| a b a b B  ^ | B → A |
| a b a b A  ^ | A → 𝜀 |
| a b a b |  |

1. Show that this grammar is ambiguous by constructing two different parse tresses for the string abab

Diagram

Description automatically generatedShape

Description automatically generated

**Problem 3 (20 points).** Compute FIRST and FOLLOW sets for the following grammar.

S → aABc | CD A → DC | BE |𝜀 B → aCB |AF

C → cC | 𝜀

D → CDb | 𝜀

E → eFc

F → Fg | 𝜀

**Show your work. An answer by itself does not count.**

FIRST

FIRST(𝜀) = {𝜀}

FIRST(a) = {a}

FIRST(b) = {b}

FIRST(c) = {c}

FIRST(e) = {e}

FIRST(g) = {g}

FIRST(S) = {a,c,𝜀,b}

FIRST(A) = {𝜀,c,b,a,e,g}

FIRST(B) = {a,𝜀,c,b,g,e}

FIRST(C) = {c,𝜀}

FIRST(D) = {c,𝜀,b}

FIRST(E) = {e}

FIRST(F) = {g,𝜀}

FOLLOWS

FIRST(S) = {}

FIRST(A) = {a,c,b,e,g}

FIRST(B) = {c,e}

FIRST(C) = {a,c,b,e,g}

FIRST(D) = {a,c,b,e,g}

FIRST(E) = {a,c,b,e,g}

FIRST(F) = {c,e,g}