CS 4510 Automata and Complexity 1/26/2022

Worksheet 2: Regular Expressions

Worksheets are provided for your study purposes only. They are not graded. The answers to prime-numbered questions will be uploaded to Canvas on Friday, Feb 5th. For the rest of questions, however, you are encouraged to discuss on Canvas and try to solve them collaboratively. You are also welcomed to discuss the problem during office hours.

In this worksheet, you will be asked to provide regular expressions for regular languages.

1. **Warmup: Binary Strings**

   Give regular expressions for following languages:

   a. Let $\Sigma = \{0, 1\}$. Let A be the set of strings of even length.

   b. Let $\Sigma = \{0, 1\}$. Let B be the set of strings having an odd number of 1s.

2. **Binary strings**

   Give regular expressions for following languages where $\Sigma = \{0, 1\}$:

   a. Binary strings that represent numbers divisible by 8. (Note: Don't include the empty string.)

   b. Binary strings where consecutive characters are different.

   c. Binary strings where exactly two of the following are true: The second character is 1, string is of even length, string ends with 0.

3. **Substring**

   Give regular expressions for following languages where $\Sigma = \{0, 1\}$:

   a. Binary strings not containing "111" as a substring

   b. Binary strings not containing any of "00" or "11" as a substring

   c. Binary strings not containing any of "01" or "10" as a substring

4. **Multiples of 4**

   Write a regex for strings that are multiples of 4 given the following alphabets.

   a. $\Sigma = \{0, 1\}$. (binary strings)

   b. $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. (strings of decimal numbers)

5. **Binary Comparison Operators**

   Write a regular expression that only accepts binary strings **(allowing any number of leading 0's)** that represent the following when interpreted as a binary number:

   a. The integer 14. (This should accept "1110", "01110", "000001110", etc.)

b. Any integer that is larger or equal to 10. (This should accept "01010", "10000", "111111111", etc.)

c. Any positive integer that is less or equal to 9. (This should accept "000", "1001", "000100", etc.)

d. Any positive integer that is "less or equal to 5" or "larger or equal to 10". (This should accept "00101", "1", "01010", "10000", "111111111", etc.)

6. **English Alphabets**

We have had enough binary strings now! Give regular expressions that generate each of the following languages. In all cases, the alphabet is $\Sigma = \{a, b\}$.

a. Set of odd length strings

b. Strings ending in double letter (A string contains a double letter if it contains "aa" or "bb" as a substring)

c. String which don't end in double letter

7. **Alternative Regular Expressions**

Let's take a break from generating regular expressions for various regular languages. The regular expressions we learned in class use three special operations: Concatenation, union, and repetition (Kleene star). However, what about some other alternative regular expressions?

a. Let L^+ be the operation that repeats strings in L at least 1 time. For example, if $L = \{a, b\}$, L^+ will contain strings like "a", "b", "aa", "bb", "ab", "ba", etc. However, unlike L^*, L^+ will not contain the empty string. Now, replace Kleene star with this new operation. Show that this alternative form of regular expressions using concatenation, union, and at least once repetition ($+$) is equally powerful as regular expressions we learned in class. Note, this proof should contain 2 directions.

b. Let $L_1 \circ L_2$ be the complement of the intersection of L_1 and L_2. That is, $L_1 \circ L_2$ accepts any language that is not both accepted by L_1 and L_2. For example, if $L_1 = \{a, b\}$ and $L_2 = \{b, c\}$, $L_1 \circ L_2$ accepts any string that is not b (It will accept "a", "c", ε, "aaaa", "abbb", "bb", "ccccca", etc). Now, replace union with this new $\circ$ operation. Show that this alternative form of regular expressions using concatenation, $\circ$, and repetition (Kleene star) is equally powerful as regular expressions we learned in class. Note, this proof should contain 2 directions.