Regular Expressions

An *alphabet* is any nonempty finite set. One alphabet you probably know well is the modern English alphabet $\{a, b, \dots, z\}$. For this problem, we'll use the much simpler alphabet $\Sigma = \{0, 1\}$. A *string* is a finite sequence of symbols from the alphabet; for example, in our simple alphabet, possible strings are $0, 01010, 111$, etc. The *length* of a string is the number of symbols the string contains. Note that it is possible to have a string of length zero, called the *empty string*. Since it is hard to see a string with no symbols, we will denote the empty string with the symbol ε. Finally, a *language* is a set of strings over a given alphabet.
In arithmetic, we can use the operations $+$ and $\times$ to build up expressions such as $(5 + 3) \times 4$. The value of the arithmetic expression is the number 32. Just as $+$ and $\times$ are operations in arithmetic, we can use operations called *regular operations* to build expressions called *regular expressions* that describe a language. Given two languages A and B, these regular operations are:

- Union, denoted $\cup$. $A \cup B$ is the set of all strings that are in A or B (or possibly both).
- Concatenation, denoted $\circ$. $A \circ B$ is the set of strings xy, where x is any string in A and y is any string in B. For simplicity, we can often leave the $\circ$ symbol out and simply write AB.
- Star, denoted *. A^* is the set of all strings $x_1x_2 \dots x_k$ for all $k \geq 0$, where each $x_1, x_2, \dots, x_k$ is in A.

Let us provide some simple examples of regular expressions. Suppose we have two languages $A = \{0\}, B = \{1\}$. Then $A \cup B = \{0, 1\}$, $A \circ B = \{01\}$, and $A^* = \{\varepsilon, 0, 00, 000, \dots\}$. Remembering our simple alphabet $\Sigma = \{0, 1\}$, Σ^* is the regular expression describing the language of all possible strings over Σ. Similarly, the regular expression $1\Sigma^*$ (short for $1 \circ \Sigma^*$) describes the language of all strings starting with a 1. $\Sigma\Sigma\Sigma$ consists of all strings of length 3.

1. Concisely interpret these regular expressions:
   (a) 0^*10^*.
   (b) $(\Sigma\Sigma)^*$.
   (c) $1^*(0001^*)^*$.
2. Find regular expressions for the following languages:
   (a) $\{w : w \text{ has at least one } 1\}$.
   (b) $\{w : w \text{ starts and ends with the same symbol}\}$.
   (c) $\{w : \text{the sum of the digits of } w\text{, minus the first digit, is even}\}$.

Generating Functions

"A generating function is a clothesline on which we hang up a sequence of numbers for display."
– Herbert Wilf, *generatingfunctionology*
To be precise, the *generating function* for the sequence a_n is the formal power series

$$\sum_{n=0}^{\infty} a_n x^n = a_0 + a_1 x + a_2 x^2 \cdots$$

where the coefficient of x^n is a_n. (You might be worried about whether a generating function converges when you evaluate it at a specific value of x, but we won't be evaluating x for any values.)

For example,

$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + \cdots$$

is the generating function for the sequence $(1, 1, 1 \ldots)$, defined by the relation $a_n = 1$. Clearly, we like the expression on the left much better than the one on the right, because of its lack of reference to infinity; we call such a form *closed form*. The right hand side is called its *series expansion*.

3. Generating Function Basics

   (a) Find the generating function for the sequence

   $$(1, 0, 3, 0, 9, 0, 27, \ldots)$$

   where $a_{2n} = 3^n$, $a_{2n+1} = 0$, and prove that your answer is correct.

   (b) Let $f(x) = \sum_{n=0}^{\infty} a_n x^n$ and $g(x) = \sum_{n=0}^{\infty} b_n x^n$ be the generating functions of a_n and b_n, respectively, and let their product be

   $$(f \cdot g)(x) = (a_0 + a_1 x + a_2 x^2 + \ldots)(b_0 + b_1 x + b_2 x^2 + \ldots) = c_0 + c_1 x + c_2 x^2 + \ldots$$

   Find a closed-form formula for the coefficients c_n. (That is, find c_n as a function of the sequences a_n and b_n)

   (c) Let A_n be the number of ways to make change for n cents using pennies (1 cent), nickels (5 cents), dimes (10 cents), and quarters (25 cents). Show that

   $$\frac{1}{(1-x)(1-x^5)(1-x^{10})(1-x^{25})}$$

   is the generating function whose coefficients are A_n.

4. Generating Functions and Recurrence Relations

   (a) Let the Fibonacci numbers be defined by the recurrence relation

   $$F_{n+2} = F_{n+1} + F_n,\ F_0 = 1,\ F_1 = 1$$

   Find a closed-form generating function for the Fibonacci numbers.

   (b) Given the generating function

   $$f(x) = \frac{1 + x^2}{1 + 2x - x^3}$$

   find a recurrence relation for the series expansion of $f(x)$ and prove that your answer is correct.

   (c) Given a generating function

   $$f(x) = \sum_{n=0}^{\infty} a_n x^n = \frac{p(x)}{c_0 + c_1 x + c_2 x^2 + \ldots + c_k x^k}$$

   where $p(x)$ is a d-degree polynomial in x, prove that the coefficients of the generating function a_n satisfy the recurrence relation:

   $$c_0 a_n + c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} = 0;\ n > d\,, n \geq k$$

Combining Regular Expressions and Generating Functions

We define the generating function of a regular expression to be the generating function for a sequence a_n of the number of strings of length n. For example, 1111 is x^4, $\epsilon \cup 00 \cup 01 \cup 11$ is $1 + 3x^2$, and $0(\epsilon \cup 0 \cup 1 \cup 11)0$ is $x^2 + 2x^3 + x^4$.

5. Let A and B be two regular expressions, and let $f_A(x)$ and $f_B(x)$ be their generating functions, respectively. Show that the generating function for $A \circ B$ is $f_A(x) \cdot f_B(x)$.

6. Let A and B be two regular expressions with $A \cap B = \emptyset$ (that is, A and B don't have any strings in common), and let $f_A(x)$ and $f_B(x)$ be their generating functions, respectively. Show that the generating function for $A \cup B$ is $f_A(x) + f_B(x)$.

7. Let A be a regular expression, and let $f_A(x)$ be its generating function. What is the generating function for A^*?

Final Task

8. Concisely describe the language of strings described by the regular expression

$$A = (\varepsilon \cup 1 \cup 11)((00)^*0(1 \cup 11))^*(\varepsilon \cup (00)^*0)$$

9. Find a generating function for A.

10. Find a recurrence relation for the number of strings in A of length n. (Don't worry about providing initial conditions.)