1 Regular Expressions

Give regular expressions for each of the following languages over the alphabet $\{\mathtt{0}, \mathtt{1}\}$.

1. All strings containing the substring $\mathtt{000}$.

   Solution: *(Intuition)* Having a substring means a string w must contain the substring x, but also that any other substrings must appear before or after x.

   (Strategy) Each string in the set has three parts: any-string + substring + any-string. any-string can be described by $\Sigma^* = (\mathtt{0} + \mathtt{1})^*$. We know the described substring....

   (Result) $(\mathtt{0} + \mathtt{1})^*\mathtt{000}(\mathtt{0} + \mathtt{1})^*$ ■

2. All strings *not* containing the substring $\mathtt{000}$.

   Solution: *(Intuition)* For a string to not allow the substring w, we can manually define our string to only contain patterns that do not contain w.

   (Strategy) Since the substring is $\mathtt{000}$ and our alphabet contains of $\mathtt{0}$s and $\mathtt{1}$s, the only patterns that are allowed repetitively are $\mathtt{1}$, $\mathtt{01}$ or $\mathtt{001}$: $(\mathtt{1} + \mathtt{01} + \mathtt{001})^*$. Since the allowed patterns ends with $\mathtt{1}$, in order to also accommodate strings that can end with $\mathtt{0}$s (with a maximum of 2 $\mathtt{0}$s at that to avoid our substring $\mathtt{000}$) the last substring would be: $(\varepsilon + \mathtt{0} + \mathtt{00})$

   (Result) $(\mathtt{1} + \mathtt{01} + \mathtt{001})^*(\varepsilon + \mathtt{0} + \mathtt{00})$ ■

   Solution: *(Intuition)* To not contain the subtring w, with n consecutive repetitions of the symbol a, we must ensure that a string w must only allow substrings containing o to $n-1$ a's at any point which can precede or succeed substrings consisting of other symbols in the alphabet.

   (Strategy) The substring that describes a max of two $\mathtt{0}$s : $(\varepsilon + \mathtt{0} + \mathtt{00})$. Since the only other symbol in our alphabet is $\mathtt{1}$, sandwiching it between our previously defined substring, while allowing repetitions, gives us :

   (Result) $(\varepsilon + \mathtt{0} + \mathtt{00})(\mathtt{1}(\varepsilon + \mathtt{0} + \mathtt{00}))^*$ ■

3. All strings in which every run of $\mathtt{0}$s has length at least 3.

   Solution: *(Intuition)* Our alphabet only has $\mathtt{0}$s and $\mathtt{1}$s. Every run of $\mathtt{0}$s with length of atleast 3 can be preceded or succeeded by any number of $\mathtt{1}$s.

   (Strategy) Each string in the set has 2 parts : $\mathtt{1}$s of any length + $\mathtt{0}$s of length >= 3 .Runs of $\mathtt{0}$s with length of at least 3 can be described as $\mathtt{0000}^*$.

   (Result) $(\mathtt{1} + \mathtt{0000}^*)^*$ ■

Solution: *(Intuition)* Since every run of **0**s should have length of at least 3 ,We can breakdown the string to three substrings x,y,z. We can define the main pattern for the body of the string to allow either **1**s or **0**s of minimum length 3 in substring y. Substrings x and z are defined to accommodate the beginning and ending patterns respectively.

(Strategy) We define y, the main body of the string, to be alternating occurrences of either any number of **1**s or **0**s of length greater than 3 ,which gives us $((\varepsilon + \mathbf{0000}^*)1)^*$. To accommodate strings that start with 1, x can be described as $(\varepsilon + \mathbf{1})$ and for strings that end with three or more zeroes, z can be described as $(\varepsilon + \mathbf{0000}^*)$

(Result) $(\varepsilon + \mathbf{1})((\varepsilon + \mathbf{0000}^*)\mathbf{1})^*(\varepsilon + \mathbf{0000}^*)$ ■

4. All strings in which **1** does not appear after a substring **000**.

Solution: *(Intuition)* We can have a maximum of two **0**s before any **1**. So we can split the problem to two substrings x and y and write it as xy. x represents all substrings with **1**s where there are less than three **0**s before a **1**. y represents substrings containing only **0**s.

(Strategy) For x, we can have either no **0** before a **1** or one **0** before a **1** or two **0**s before a **1**. This can be represented by $(\mathbf{1} + \mathbf{01} + \mathbf{001})^*$. For y, we can use $\mathbf{0}^*$ since it represents zero or more **0**s.

(Result) $(\mathbf{1} + \mathbf{01} + \mathbf{001})^*\mathbf{0}^*$ ■

5. All strings containing at least three **0**s.

Solution: *(Intuition)* Containing at least three **0**s can be interpretted as having three substrings say x, y, z. Then this problem becomes similar to problem 1 in that we can have any other substring before, after and inbetween x, y and z. *(Strategy)* Each string has seven parts: any-string + substring 1 + any-string + substring 2 + any-string + substring 3 + any-string. Any-string can be described by $\Sigma^* = (\mathbf{0} + \mathbf{1})^*$ and our substrings are just **0**. $(\mathbf{0} + \mathbf{1})^*\mathbf{0}(\mathbf{0} + \mathbf{1})^*\mathbf{0}(\mathbf{0} + \mathbf{1})^*\mathbf{0}(\mathbf{0} + \mathbf{1})^*$ ■

Solution (clever): *(Intuition)* This problem can instead be viewed from the lens of counting the first three **0**s. Because our alphabet only have **0**s and **1**s, this means that the only symbols around the first three **0**s can be **1**s. Then any-string can follow after the first three **0**s. The resulting regular expression is simpler than the previous solution. The second clever solution is the same but in reverse. Reversing the expression works because the substring **0** is symmetric but it can also be viewed as counting the last three **0**s. *(Strategy)* Each string has seven parts: $\mathbf{1}s + \mathbf{0} + \mathbf{1}s + \mathbf{0} + \mathbf{1}s + \mathbf{0}$ + any-string. Any-string can be described by $\Sigma^* = (\mathbf{0} + \mathbf{1})^*$, **1**s is just $\mathbf{1}^*$ and**0**is just **0**. The second clever solution is just this in reverse. $\mathbf{1}^*\mathbf{01}^*\mathbf{01}^*\mathbf{0}(\mathbf{0} + \mathbf{1})^*$ or $(\mathbf{0} + \mathbf{1})^*\mathbf{01}^*\mathbf{01}^*\mathbf{01}^*$ ■

6. Every string except $\mathtt{000}$. *[Hint: Don't try to be clever.]*

Solution: Every string $w \neq \mathtt{000}$ satisfies one of three conditions: Either $|w| < 3$, or $|w| = 3$ and $w \neq \mathtt{000}$, or $|w| > 3$. The first two cases include only a finite number of strings, so we just list them explicitly. The last case includes *all* strings of length at least 4.

$$\varepsilon + \mathtt{0} + \mathtt{1} + \mathtt{00} + \mathtt{01} + \mathtt{10} + \mathtt{11}$$
$$+ \mathtt{001} + \mathtt{010} + \mathtt{011} + \mathtt{100} + \mathtt{101} + \mathtt{110} + \mathtt{111}$$
$$+ (\mathtt{1} + \mathtt{0})(\mathtt{1} + \mathtt{0})(\mathtt{1} + \mathtt{0})(\mathtt{1} + \mathtt{0})(\mathtt{1} + \mathtt{0})^*$$

■

Solution (clever): A string w in the language falls in one of the following three categories:

(a) Does not contain $\mathtt{1}$ and $|w| < 3$.

(b) Contains $\mathtt{1}$ in the first three symbols.

(c) Does not contain $\mathtt{1}$ in the first three symbols and $|w| > 3$.

The strings in (a) can be listed explicitly. The strings in (b) can be further categorized based on the location of the first occurrence of $\mathtt{1}$, and can be written as $\big(\mathtt{1} + \mathtt{01} + \mathtt{001}\big)(\mathtt{1} + \mathtt{0})^*$. Notice that strings with multiple $\mathtt{1}$s in the first three symbols are also captured by the given expression. For example, $\mathtt{111}$ can be generated by $\mathtt{1}(\mathtt{1} + \mathtt{0})^*$. Finally, the strings in (c) can be written as $\big(\mathtt{000}(\mathtt{1} + \mathtt{0})\big)(\mathtt{1} + \mathtt{0})^*$. Combining the three catefories, we obtain the following expression:

$$\varepsilon + \mathtt{0} + \mathtt{00} + \big(\mathtt{1} + \mathtt{01} + \mathtt{001} + \mathtt{000}(\mathtt{1} + \mathtt{0})\big)(\mathtt{1} + \mathtt{0})^*$$

■

7. All strings w such that *in every prefix of* w, the number of **0**s and **1**s differ by at most 1.

Solution: Equivalently, strings that alternate between **0**s and **1**s: $(\mathbf{01}+\mathbf{10})^*(\varepsilon+\mathbf{0}+\mathbf{1})$ ■

8. Any string not in $\mathbf{0}^*+\mathbf{1}^*$

Solution: So the language represented by $\mathbf{0}^*+\mathbf{1}^*$ contains all the binary strings that have only **0**'s and all the binary strings that have only **1**'s. Hence, all the strings not included in that language have at least one **0** and one **1**. Something like this would do:

$$(\mathbf{0}+\mathbf{1})^*\mathbf{0}(\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{0}+\mathbf{1})^*+(\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{0}+\mathbf{1})^*\mathbf{0}(\mathbf{0}+\mathbf{1})^*$$

Note that we need to account for **01** and **10** so that's why we have to have two parts to the above expression. ■

*9. All strings containing at least two **0**s and at least one **1**.

Solution: There are three possibilities for how such a string can begin:

- Start with **00**, then any number of **0**s, then **1**, then anything.
- Start with **01**, then any number of **1**s, then **0**, then anything.
- Start with **1**, then a substring with exactly two **0**s, then anything.

All together: $\mathbf{000}^*\mathbf{1}(\mathbf{0}+\mathbf{1})^* + \mathbf{011}^*\mathbf{0}(\mathbf{0}+\mathbf{1})^* + \mathbf{11}^*\mathbf{01}^*\mathbf{0}(\mathbf{0}+\mathbf{1})^*$

Or equivalently: $\left(\mathbf{000}^*\mathbf{1} + \mathbf{011}^*\mathbf{0} + \mathbf{11}^*\mathbf{01}^*\mathbf{0}\right)(\mathbf{0}+\mathbf{1})^*$ ■

Solution: There are three possibilities for how the three required symbols are ordered:

- Contains a **1** before two **0**s: $(\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{0}+\mathbf{1})^*\mathbf{0}(\mathbf{0}+\mathbf{1})^*\mathbf{0}(\mathbf{0}+\mathbf{1})^*$
- Contains a **1** between two **0**s: $(\mathbf{0}+\mathbf{1})^*\mathbf{0}(\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{0}+\mathbf{1})^*\mathbf{0}(\mathbf{0}+\mathbf{1})^*$
- Contains a **1** after two **0**s: $(\mathbf{0}+\mathbf{1})^*\mathbf{0}(\mathbf{0}+\mathbf{1})^*\mathbf{0}(\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{0}+\mathbf{1})^*$

So putting these cases together, we get the following:

$$\begin{aligned}&(\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{0}+\mathbf{1})^*\mathbf{0}(\mathbf{0}+\mathbf{1})^*\mathbf{0}(\mathbf{0}+\mathbf{1})^*\\ +&(\mathbf{0}+\mathbf{1})^*\mathbf{0}(\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{0}+\mathbf{1})^*\mathbf{0}(\mathbf{0}+\mathbf{1})^*\\ +&(\mathbf{0}+\mathbf{1})^*\mathbf{0}(\mathbf{0}+\mathbf{1})^*\mathbf{0}(\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{0}+\mathbf{1})^*\end{aligned}$$

■

Solution (clever): $(\mathbf{0}+\mathbf{1})^*\left(\mathbf{101}^*\mathbf{0}+\mathbf{011}^*\mathbf{0}+\mathbf{01}^*\mathbf{01}\right)(\mathbf{0}+\mathbf{1})^*$ ■

★10. All strings in which the substring **000** appears an even number of times.
(For example, **0001000** and **0000** are in this language, but **00000** is not.)

Solution: Every string in $\{\mathbf{0},\mathbf{1}\}^*$ alternates between (possibly empty) blocks of **0**s and individual **1**s; that is, $\{\mathbf{0},\mathbf{1}\}^* = (\mathbf{0}^*\mathbf{1})^*\mathbf{0}^*$. Trivially, every **000** substring is contained in some block of **0**s. Our strategy is to consider which blocks of **0**s contain an even or odd number of **000** substrings.

Let X denote the set of all strings in $\mathbf{0}^*$ with an even number of **000** substrings. We easily observe that $X = \{\mathbf{0}^n \mid n = 1 \text{ or } n \text{ is even}\} = \mathbf{0} + (\mathbf{00})^*$.

Let Y denote the set of all strings in $\mathbf{0}^*$ with an *odd* number of **000** substrings. We easily observe that $Y = \{\mathbf{0}^n \mid n > 1 \text{ and } n \text{ is odd}\} = \mathbf{000}(\mathbf{00})^*$.

We immediately have $\mathbf{0}^* = X + Y$ and therefore $\{\mathbf{0},\mathbf{1}\}^* = ((X+Y)\mathbf{1})^*(X+Y)$.

Finally, let L denote the set of all strings in $\{\mathbf{0},\mathbf{1}\}^*$ with an even number of **000** substrings. A string $w \in \{\mathbf{0},\mathbf{1}\}^*$ is in L if and only if an even number of blocks of **0**s in w are in Y; the remaining blocks of **0**s are all in X. We consider two simple subcases to help us make sure we don't make mistakes.

- Exactly zero blocks of **0**s are in Y. This means all blocks of **0**s are in X and they alternate with blocks of **1**s. We capture this by $(X\mathbf{1})^*X$ where the last X is to allow ending in blocks of **0**s.
- Exactly two blocks of **0**s are in Y. The expression $(X\mathbf{1})^*Y\mathbf{1}\cdot(X\mathbf{1})^*Y(\varepsilon+\mathbf{1}(X\mathbf{1})^*X)$ captures this where we are paying attention in the expression to what comes after the second block of Y; either we end in that block or we need a **1** to separate a string that has zero blocks from Y.

Now we consider the general case with an even number of blocks in Y which can be zero, two or a multiple of two that can be simulated by repetitions. We can obtain this by considering the expression for exactly two blocks and allowing concatenation arbitrary number of times while being careful to ensure that we separate with a **1** as needed and allowing for the expression to end appropriately. The expression $\left((X\mathbf{1})^*Y\mathbf{1}\cdot(X\mathbf{1})^*Y\mathbf{1}\right)^*$ allows an arbitrary even number of blocks in Y with the caveat that it always ends with $Y\mathbf{1}$ if it has non-zero repetitions. We need to allow ending in Y or $(X\mathbf{1})^*X$ as in the second subcase above. Letting $Z = (X\mathbf{1})^*Y\mathbf{1}\cdot(X\mathbf{1})^*Y$ helps us simplify the final expression.

Putting the preceding observations together we obtain the following expression.

$$L = (Z\mathbf{1})^*(Z + (X\mathbf{1})^*X)$$

We will not plug in the expressions for X and Y because it will make the expression too long. There are likely to be shorter expressions based on better case analysis but debugging them is probably not worth it unless it is very important in some application.

Whew! ■