Chapter 6

Nature of Regular Languages

1. If the alphabet is $\{a\}$, what is the complement of a^*? Of a^+?

   Solution: Complement of a^* is $\{\}$. Complement of a^+ is $\{\lambda\}$.

2. We know that the concatenation of two regular languages is a regular language. Consider the language $L = 0^n1^n$ over $\{0, 1\}$; L is not regular. Now consider, the language $L_1 = \{0^n\} = 0^*$ and $L_2 = \{1^n\} = 1^*$. L_1 and L_2 are obviously regular. Explain why although L_1 and L_2 are regular, L which could be seen as a concatenation of L_1 and L_2 is not regular.

   Solution: Because the concatenation of the two languages $L_1.L_2$ includes any string from L_1 concatenated with any string from L_2, not just corresponding strings of equal length. The given language L is a proper subset of this language and includes only corresponding pairs. Identifying matching pairs of equal length is not possible in an infinite regular language and therefore it is not regular. This is not a violation of the closure property of regular languages under the concatenation operator.

3. What happens if we apply the Pumping Lemma to show that a formal language such as $((a + b)\ (a + b)\ (a + b))^*$ that is actually regular is not regular? Explain.

   Solution: We will not be able to establish a contradiction. We assume that the language is regular and in fact it is. As such, no matter which string and its partitions are chosen, pumping the string up or down always results in strings that do belong to the language. With no contradiction, we will not be in a position to conclude (wrongly, in this case) that the language is not regular.

   For the given example string, if we choose $w = a^m b^m a^m$ the opponent can always choose a partition so that y contains any three symbols. Pumping y up or down will always keep the length of string divisible by 3 and there is no contradiction at all.

A. Using closure properties of regular languages, construct a finite automaton (NFA or DFA) for:

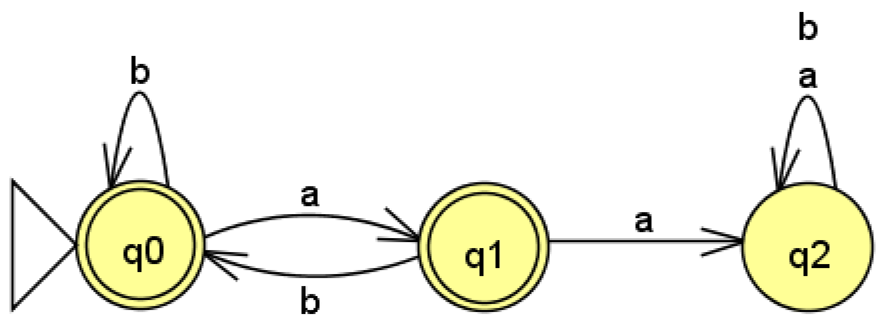
4. Binary strings which when interpreted as positive integers are not divisible by 4.

   Solution: We know how to construct a DFA for binary strings which when interpreted as positive integers are divisible by 4 (see Figure-2.4). We also know that a complement of a regular language is a regular language. Therefore the complement of the language of this DFA, that is, binary strings that are not divisible by 4 is also regular. We can construct a DFA for it by simply interchanging the accepting and non-accepting states of the DFA in Figure-2.4.

   Please see also the VIDEO SOLUTION for this exercise.

5. Strings over $\{a, b\}$ that do not contain two consecutive a s.

   Solution: Let us construct a DFA for strings that do contain two consecutive a s and then complement its language by interchanging accepting and non-accepting states:



B. Using closure properties of regular languages, show that the following languages are regular:

6. Binary strings that do not contain the substring 101.

   Solution: We know that binary strings that do contain 101 are regular since we can construct a RegEx for them: (0 + 1)*101(0 + 1)*
   See also the DFA in Figure-2.10. Therefore its complement, the set of binary strings that do not contain the substring 101 must also be regular.

   Please see also the VIDEO SOLUTION for this exercise.

7. Binary strings made up of two parts; the first part begins with a 1 and ends with a 0; the second part begins and ends with a 1.

   Solution: A RegEx for the first part is: 1(0 + 1)*0
   A RegEx for the second part is: 1 + 1(0 + 1)*1
   Both are regular. The concatenation of two regular languages is also regular. Therefore the given set of binary strings is also regular.

8. Binary strings which represent positive integers that are not divisible by 5.

   Solution: We can construct a DFA for binary strings which represent positive integers that are divisible by 5 (see Figure-2.9). Its complement must also be regular. Therefore, binary strings which represent positive integers that are not divisible by 5 are also regular.

9. Binary strings which represent positive integers that are divisible by 3 but not by 5.

   Solution: We know that binary strings representing positive integers divisible by 3 are regular (see DFA in Figure. 2.8). Similarly, we know that binary strings representing positive integers divisible by 5 are regular (see DFA in Figure-2.9). If we take the complement of this language and then its intersection with the first language, the resulting language must also be regular since we know that regular languages are closed under both complementation and intersection.

10. Binary strings which when reversed represent positive integers that are divisible by 3.

Solution: We know that binary strings divisible by 3 are regular (see DFA in Figure-2.8). We also know that the reverse of a regular language is regular. Therefore the given language is regular. In fact, the automaton in Figure. 2.8b is fully symmetrical; reversing it does not change it in any way at all. In other words, the set of binary strings which when reversed represent positive integers that are divisible by 3 are the same as those that are divisible by 3 (if leading 0 s are permitted).

Please see also the VIDEO SOLUTION for this exercise.

11. Strings over $\{a, b, c\}$ that either have the same symbol in all odd positions or have the same symbol in all even positions.

Solution: We know from the DFA in Figure-2.12 that strings with the same symbol in all odd positions is a regular language (with a suitable extension to include a third branch for the additional symbol c). On the same lines, those with the same symbol in all even positions are also a regular language (since we can construct a similar DFA for them). In fact, we can also construct a RegEx for them:

$$(((a + b + c)a)^* + ((a + b + c)b)^* + ((a + b + c)c)^*)(a + b + c + \lambda)$$

Since the union of two regular languages is also a regular language, the given language is regular.

12. Strings over $\{a, b, c\}$ whose length is neither an even number nor divisible by 3 or 5.

Solution: Strings whose length is even is a regular language:

$$((a + b + c)(a + b + c))^*$$

Strings whose length is divisible by 3 are also regular:

$$((a + b + c)(a + b + c)(a + b + c))^*$$

Strings whose length is divisible by 5 are also regular:

$$((a + b + c)(a + b + c)(a + b + c)(a + b + c)(a + b + c))^*$$

The complement of each of the above is also a regular language. If we take the intersection of the three complements, we get the given language which must also be regular since regular languages are closed under complementation and intersection.

13. The set of encrypted strings in a rather simple encryption scheme that uses the set of prime numbers less than 1000. A given message is encrypted by repeated concatenations of prime numbers from the set.

Solution: The set of prime numbers less than 1000 is a finite set and is therefore regular (see Theorem 30 in Appendix B). Replacing the symbols in a given message by prime numbers is a homomorphism and regular languages are closed under homomorphism. Concatenation also preserves regularity and therefore the set of encrypted messages is also a regular language.

C. Show that the following languages are not regular:

14. Binary strings of even length having the same number of 1 s in its two halves.

Solution: Let us assume that they are regular and apply the Pumping Lemma. Let us choose $w = 1^m01^m0$ where m is an unknown constant. $w = xyz$. y must contain at least one 1. Pumping it up or down either makes the string length odd (and thus it does not belong to the language) or creates an imbalance between the two halves of the string. For example, if y = 11 and we pump the string up to i = 2, both 0 s belong to the second half of the string and thus, the first half contains more 1 s than the second. In all cases, a contradiction occurs since the resulting string does not belong to the language. Therefore it is not a regular language.

15. The language 0^n1^m, $n <= m$.

    Solution: Using the Pumping Lemma in a proof by contradiction, let $w = 0^m1^m$, where m is the Pumping Lemma constant. The repeating pattern y must contain at least one 0 (and no 1) and pumping it up will create more 0 s than 1 s. The resulting string does not belong to the language. This is a contradiction and therefore the given language is not regular.

16. Binary strings containing more 1 s than 0 s.

    Solution: Using the Pumping Lemma in a proof by contradiction, let $w = 0^m1^{m+1}$, where m is the Pumping Lemma constant. The repeating pattern y must contain at least one 0 (and no 1) and pumping it up will create more 0 s than 1 s. The resulting string does not belong to the language. This is a contradiction and therefore the given language is not regular.

17. Odd palindromes over {a, b}, that is, wcw^R.

    Solution: Using the Pumping Lemma in a proof by contradiction, let $w = a^mca^m$, where m is the Pumping Lemma constant. The repeating pattern y must contain at least one a and pumping it up will create more a s to the left of c than to its right. The resulting string does not belong to the language; it is no longer a palindrome. This is a contradiction and therefore the given language is not regular.

18. Binary strings whose length is a perfect square.

    Solution: Using the Pumping Lemma in a proof by contradiction, let $w = 0^{m \times m}$, where m is the Pumping Lemma constant. The repeating pattern y must contain at least one 0 and a maximum of m 0 s. Pumping it up once will create at most m additional 0 s. The resulting string does not belong to the language since $(m \times m) + m < (m + 1) \times (m + 1)$ which is the length of the next string in the language of perfect squares. This is a contradiction and therefore the given language is not regular.

19. The language of subtraction (in the unary number system). For convenience, use the sublanguage in which the result of subtraction is always zero or positive.

    Solution: The given language is $a^mb^nc^k$ where $k = m - n$. Using the Pumping Lemma in a proof by contradiction, let $w = a^mb^mc^0$, where m is the Pumping Lemma constant. The repeating pattern y must contain at least one a (and no b) and pumping it up will create more a s. The resulting string does not belong to the language since it contains more a s than b s and yet no c at all. This is a contradiction and therefore the given language is not regular.

20. The language of unary multiplication.

Solution: The given language is $a^m b^n c^k$ where $k = m \times n$. Using the Pumping Lemma in a proof by contradiction, let $w = a^m b^m c^{m \times m}$, where m is the Pumping Lemma constant. The repeating pattern y must contain at least one a (and no b) and pumping it up will create more a s. The resulting string does not belong to the language since it contains more a s than originally and yet there is no increase in the number of c s at all. This is a contradiction and therefore the given language is not regular.