# CS 310, Assignment 1

Answers

1. Let $\Sigma = \{a, b\}$ and consider languages $A = \{b, aa, ba\}$ and $B = \{\varepsilon, a, bb\}$.

   (a) Write down all strings in $\Sigma^*$ that have length at most two.

   ---

   ANSWER:

   $\varepsilon$, $a$, $b$, $aa$, $ab$, $ba$, $bb$

   ---

   (b) How many strings are in $A \cdot B$? Write down all of them.

   ---

   ANSWER:

   $A \cdot B = \{b, ba, bbb, aa, aaa, aabb, baa, babb\}$. The language consists of 8 strings rather than 9 since $ba$ has two different decompositions as a concatenation of strings from $A$ and $B$; however a language is a set, and duplicates are ignored in a set.

   ---

   (c) How many strings are in $B \cdot A$? Write down all of them.

   ---

   ANSWER:

   $B \cdot A = \{b, aa, ba, ab, aaa, aba, bbb, bbaa, bbba\}$. The language consists of 9 (distinct) strings this time.

   ---

2. Let $R = (ba^*b + aba^*ba)^*$ and $S = (a^*ba^*ba^*)^*$, both over $\Sigma = \{a, b\}$.

   (a) Give an example of a string $z$ that is both in $R$ and in $S$ (that is, $z \in R \cap S$ ).

   ---

   ANSWER:

   Many such strings exist including $\varepsilon$, $bb$, $ababa$, etc.

   ---

   (b) Is it possible to find a string $x$ that is in $R$ and is not in $S$ (that is, $x \in R \cap \overline{S}$)? If yes, write it down; if not explain briefly why.

---

ANSWER:

No. Every string in $R$ is also in $S$, meaning that $R \cap \overline{S} = \emptyset$. Indeed, note first that $S$ contains all the strings with an even number of $b$s other than the strings in $a^*$. In turn, both $ba^*b$ and $aba^*ba$ contain exactly two $b$s and none of them is in $a^*$, so any string in the closure of the union of the two will certainly contain an even number of $b$s, will not be in $a^*$, and so will be in $S$.

---

(c) Is it possible to find a string $y$ that is in $S$ and is not in $R$ (that is, $y \in S \cap \overline{R}$)? If yes, write it down; if not explain briefly why.

---

ANSWER:

A string from $R$ that contains only two occurrences of $b$ must start and end with the same symbol (either $a$ or $b$). Any string containing two $b$s and starting and ending with different symbols will thus be a possible $y$ (meaning a member of $S \cap \overline{R}$). Such strings include $baba$, $abab$, $abaaab$, and so on.

---

3. Give regular expressions for each of the following languages over $\Sigma = \{0, 1\}$.

(a) All strings that begin with 1 *and* end with 00.

---

ANSWER:

$1(0+1)^*00$

---

(b) All strings that have both 00 and 01 as substrings. Note that the substrings can occur in either order and possibly overlap.

---

ANSWER:

The possible variants are: 00 before 01, 01 before 00, and one occurrence of 001 (the two substrings overlapping). The most straightforward answer is therefore the following regular expression: $(0+1)^*00(0+1)^*01(0+1)^* + (0+1)^*01(0+1)^*00(0+1)^* + (0+1)^*001(0+1)^*$.

---

4. Describe the language $L$ over $\Sigma = \{0, 1\}$ defined by each of the following equations. Justify your answers as fully as you can.

(a) $L = 0 + 1L$

---

ANSWER:

0 is certainly in $L$. The recursive case will add one 1 in front, and we can apply that case as many times as we like. I am therefore guessing that $L = 1^*0$.

If we want to be more organized we can apply the approximating from below method from the lectures:

$$\begin{aligned}
L_0 &= \emptyset \\
L_1 &= 0 \\
L_2 &= \{0, 10\} \\
L_3 &= \{0, 10, 110\} \\
L_4 &= \{0, 10, 110, 1110\} \\
L_5 &= \{0, 10, 110, 1110, 11110\} \\
\ldots
\end{aligned}$$

At this point I am fairly confident that $L = 1^*0$.

Now the above is just a guess. We still need to make sure that $1^*0$ is indeed a solution of the equation. We note that a string in $1^*0$ either has no 1 symbols (case in which it is 0) or has at least one 1 (case in which it belongs to the language $11^*0$). There is no other alternative and therefore:

$$1^*0 = 0 + 11^*0$$

This proves that $L = 1^*0$.

Are there any more solutions to the equation? It does not appear so. There is no other way to describe $L$ other than the way I did in the first paragraph of this answer. More formally the equation is guarded (we add something at each recursive step), which guarantees the uniqueness of the solution; you are however not expected to know that (even if now you know).

---

(b) $L = L + 1 + L$

---

ANSWER:

Lucky for us, the approximation scheme converges pretty quickly:

$$\begin{aligned}
L_0 &= \emptyset \\
L1 &= \emptyset + 1 + \emptyset \;=\; 1 \\
L_2 &= 1 + 1 + 1 \;=\; 1 \;=\; L_1 \text{ (no change)}
\end{aligned}$$

and so L = 1. Since we were able to actually compute the whole set $L$ this way we do not need to verify the solution.

It is also easy to note that $L + 1 + L = L$ whenever $1 \in L$ (since $1 \in L$ implies $L + 1 = L$ and thus $L + 1 + L = L + L = L$). Therefore any set that includes 1 is a solution for the equation.

Why did the approximation method give 1 as a solution? This method approximates from below and so will produce the *smallest* solution. Clearly 1 (the solution given by our method) is the smallest set containing the string 1.

---