1. (3 pts) Let $\Sigma = \{a, b, c\}$. Find two languages $A, B \subseteq \Sigma^*$ so that $|A| = 5$, $|B| = 3$, and $|AB| \neq 15$.

We want to choose our languages so that there the same string can occur with more than one factorization. If we take $A = \{\lambda, a^2, a^4, a^6, a^8\}$, then $|A| = 5$ and if we take $|B| = \{a, a^3, a^5\}$, then $|B| = 3$, and $AB = \{a, a^2, a^5, a^7, a^9, a^{11}, a^{13}\}$, so $|AB| = 7$. In this case we even have $A \cap B = \emptyset$, which was not required.

2. (3 pts) Let $\Sigma = \{a, b, c\}$ and $A = \{a, b, c^2\}$ and $B = \{a, b^2, c\}$.

Compute $A \cup B$ and $AB$.

Is $A \subseteq B^*$?

$A \cup B = \{a, b, c^2\} \cup \{a, b^2, c\} = \{a, b, b^2, c, c^2\}$

$AB = \{a, b, c^2\}\{a, b^2, c\} = \{a^2, ab^2, ac, ba, b^3, bc, c^2a, c^2b^2, c^3\}$

$a \in B^*$ and $c^2 \in B^*$, however $b \notin B^*$, so $A \nsubseteq B^*$.

3. (4 pts) Give a recursive definition of the language $L \subseteq \{a, b\}^*$ consisting of all strings having twice as many $a$’s as $b$’s.

(So for example $aaabab \in L$ and $\lambda \in L$, but $aa \notin L$.)

There are many different elements in this language and we have to be sure that we describe how to recursively generate all of them.

If there are twice as many $a$’s as $b$’s, then if the string is not empty, then there is some $b$ which will have to match to two $a$’s, but they need not be together. So, to make sure we have everything we can do as follows:

**BASIS:** $\lambda \in L$

**RECURSIVE STEP:** If $w = u_1u_2u_3u_4 \in L$ then the three strings $u_1au_2au_3bu_4$, $u_1au_2bu_3au_4$, and $u_1bu_2au_3au_4$, are all in $L$. (Note nothing prevents $u_i = \lambda$.)

**CLOSURE:** All elements of $L$ are generated from the basis after a finite number of recursive steps.

Another solution is to note that if no $a$’s occur next to one another, then the $a$’s would have to alternate with powers of $b$, and the only way for there to be twice as many $a$’s is if the string is just $aba$. Otherwise there must be a substring $a^2b$ or $ba^2$, and we have a second method.

**BASIS:** $\lambda, aba \in L$

**RECURSIVE STEP:** If $w = u_1u_2 \in L$ then the two strings $u_1a^2bu_2$, and $u_1ba^2u_2$ are both in $L$. (Note nothing prevents $u_i = \lambda$.)

**CLOSURE:** All elements of $L$ are generated from the basis after a finite number of recursive steps.