Math Circle - Generating Secret Codes

A **formal grammar** is a collection of four things:

i. A finite alphabet denoted Σ. Elements in Σ are called **terminals**. Usually *terminals* are denoted with lower-case letters. Strings of *terminals* are called **words**.

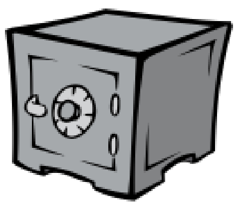
ii. A finite set of **states** denoted by Ω. Usually *states* are denoted with upper-case letters.

iii. A particular starting state $S \in \Omega$.

iv. A finite set of **rules** for transforming states into words consisting of both *states* and/or *terminals*.

Your team of hackers from the last activity was able to break into the software system for each of the nine countries. Shame on you! However, because of your expertise, you have now been hired by the countries to help encode some new and hopefully more secure codes! Now, the alphabet used is $\Sigma = \{a, b, c\}$; that is, all codes will be strings of as, bs, and cs.

For each of the sets of codes (languages) given below, you and your team have two jobs. First, give three distinct codes which are acceptable to this system. Second, create a grammar which generates only all those codes which are acceptable. Happy encoding!



1. $\{(abc)^n : n \geq 0\}$.

2. $\{a^n b^m c^k : n, m, k \geq 0\}$.

3. $\{a^n b^m c^n : n, m \geq 0\}$.

4. Words in which any a or b must be followed immediately by at least one c.

5. Words in which any a is immediately followed by exactly one c and any b is immediately followed by exactly two c's.

6. Words in which a^2 does not appear as a substring.