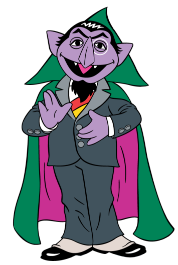
Learning objectives:

- ☐ relate properties of functions to the mapping rule,
- ☐ apply product and addition rules to count sets without explicitly listing all items in the set,
- ☐ apply the pigeonhole principle to relate the cardinalities of two sets.

Last lecture, we talked about functions – more specifically, we looked at properties of functions (injective, surjective and bijective). These properties allowed us to characterize the relationship between the input set (domain) and the output set (codomain) of a function. Today, we want to start relating the cardinalities of these input and output sets, without having to explicitly define a function that maps the domain to codomain.

We want to be able to count *without* explicitly listing the items in a set, or explicitly constructing a map. For example, we might want to count the number of possible passwords, the number of connections in a network or the number of ways we can distribute tasks to a processor.



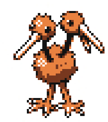
1 Rules

Recall our discussion about properties of functions. The **mapping rule** is a useful way to relate the sizes of sets, depending on how elements from a domain are mapped to a codomain. Given a function $f\colon A \to B$,

- if f is injective, then $|A| \leq |B|$,
- if f is surjective, then $|A| \geq |B|$,
- if f is bijective, then $|A| = |B|$.

This can be particularly useful if we know that a bijection exists between A and B, because we then know that the cardinalities of A and B are the same.

Three properties of functions



Remember that a function $f\colon A \to B$ is:

- **injective** if for every $x, y \in A$, $x \neq y$ means $f(x) \neq f(y)$.
- **surjective** if for every $b \in B$, there is an $a \in A$ with $f(a) = b$.
- **bijective** if f is both injective and surjective.

Product rule Let's look at some more counting rules. The **product rule** says that *if an event A can occur in m ways, and each possibility for B allows for exactly n ways for event B, then the event A **and** B can occurs in $m \times n$ ways.*

Addition rule If an event A can occur in m ways, and another **disjoint** event B can occur in n ways. Then the combined event A **or** B can occur in $m + n$ ways. When applying the addition rule, it is important to identify events that are **disjoint** (there is no way they can occur at the same time).

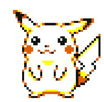
If the two events A and B are not disjoint, then we need to subtract the number of events in which A and B both occur simultaneously.

Let's now practice with some examples.

Example 1:
How many possible 6 character license plates are there if the first 3 characters must be letters, and the last 3 characters must be numbers?

Solution:
There are 26 possible letters for each of the first three characters, and 10 possible numbers for the last three characters, giving a total of $26 \times 26 \times 26 \times 10 \times 10 \times 10 = 17,576,000$ possible license plates.

What if everyone in California bought a car?



Hmmm, interesting question. If everyone in the state of California (population $\approx$ 39.5 million) bought a car, then we would run out of license plates with this model. This might be why California uses 7 characters on their license plates.

Example 2:
How many possible passwords are there (passwords must be between 6 and 8 characters long), where each character is either an uppercase letter or a digit and passwords <u>must</u> contain at least one digit?

Solution:
We know that passwords are either of 6, 7 **or** 8 characters. These events are all disjoint, so we can count the number of possibilities for each and then add them up. Let p_n be the number of passwords of length n. There are 36 possibilities for each password character resulting from the 26 letters and 10 digits. The total number of password possibilities with n characters is then $p_n = 36^n$. However, we have the restriction that passwords must contain at least 1 digit. This is violated if the passwords consist of only letters, for which there are 26^n possibilities. We need to subtract these possibilities, therefore giving a total of $p_n = 36^n - 26^n$ possibilities. To count passwords of length 6, 7 or 8, we have

$$\#\text{ passwords} = p_6 + p_7 + p_8 = (36^6 - 26^6) + (36^7 - 26^7) + (36^8 - 26^8) \approx 2.68 \times 10^{12}.$$

Principle of inclusion-exclusion In the last example, we could count the number of 6-, 7-, or 8-character passwords by adding up the possible number of passwords of each. We could do this because each event was disjoint. But what if there is some overlap between several events?

We actually hinted at this earlier. If the events are not disjoint, then we need to subtract the number of events in which the events occurs simultaneously. This is know is the **inclusion-exclusion principle** because we are *including* (by adding) the number of possible events of each, but excluding (by subtracting) the number of events that occur simultaneously. For two sets A and B, this can be written mathematically as

$$\underbrace{|A \cup B|}_{A \text{ or } B \text{ occurs}} = \overbrace{|A|}^{A \text{ occurs}} + \underbrace{|B|}_{B \text{ occurs}} - \overbrace{|A \cap B|}^{A \text{ and } B \text{ occurs}}.$$

Example 3:
How many 5-bit strings start with 1 or end with 00? 16? 20? 24 or 32?

Solution:
There are two choices for each bit: either a 0 or a 1. The number of 5-bit strings that start with a 1 is 2^4 since there are only 4 remaining bits to choose from after selecting a 1 for the first digit. By the same reasoning, the number of 5-bit strings that end with 00 is 2^3 since we only have three bits left to choose after selecting 00 for the last two bits. However, there are common bit strings in these two options. Specifically, those strings which both start with a 1 and end with a 00. The number of common bit strings is 2^2 since there are only two bits to choose from. The total number of bit strings that start with 1 or end with 00 is then $2^4 + 2^3 - 2^2 = 20$.

2 The pigeonhole principle

The pigeonhole principle is an extremely useful tool that can be used to show some interesting results (with applications to computer science).

Theorem 1. *If n objects are placed into k boxes then there is at least one box contains $\geq \lceil \frac{n}{k} \rceil$ objects.*

Proof. We use a proof by contradiction. Suppose n objects are placed into k boxes but there are no boxes containing $\geq \lceil \frac{n}{k} \rceil$ objects. In other words, all boxes contain $< \lceil \frac{n}{k} \rceil$ objects. We can rewrite this as: all boxes contain $\leq \lceil \frac{n}{k} \rceil - 1$ objects. With k boxes, the total number of objects is then $\leq k \left(\lceil \frac{n}{k} \rceil - 1 \right)$. But $\lceil \frac{n}{k} \rceil < \frac{n}{k} + 1$ so overall there would be less than $k \left(\left(\frac{n}{k} + 1 \right) - 1 \right)$ objects, which is a contradiction. □

What's that ⌈symbol⌉?

This is used to denote the **ceiling** which rounds an input real number to the next highest integer. For example $\lceil \frac{7}{2} \rceil = 4$.

Example 4:
Given 35 people in a single room, show that there are at least three people born in the same month.

Solution:
The following direct proof uses the pigeonhole principle along with the fact that there are 12 months in a year.

Proof. By the pigeonhole principle, there are 35 people and 12 months for each person's birthday. Therefore, at least $\lceil \frac{35}{12} \rceil = 3$ people are born in the same month. $\square$