4.1 Basic Counting

A string is simply a finite sequence of symbols from some alphabet.

Example: How many strings of length 2 from the alphabet \( a, \ldots, z \) of lower case Latin letters are there?

\[
\begin{align*}
\text{26} & \mid \text{aa, ab, \ldots, az} \\
\text{26} & \mid \text{ba, bb, \ldots, bz} \\
\frac{26 \times 26}{2} & = 676
\end{align*}
\]

Product Rule

Suppose a task \( T \) can be broken down into two subtasks \( T_1, T_2 \) to be performed in succession. Suppose \( T_1 \) can be performed in \( n_1 \) ways, and when \( T_1 \) is complete, \( T_2 \) can be performed in \( n_2 \) ways. Then \( T \) can be performed in \( n_1 \cdot n_2 \) ways.

More generally suppose \( T \) can be broken down into \( m \) successive subtasks \( T_1, \ldots, T_m \) and that after \( T_1, \ldots, T_{i-1} \) are complete \( T_i \) can be performed in \( n_i \) ways (\( 1 \leq i \leq m \)). Then \( T \) can be performed in \( n_1 \cdot n_2 \cdots n_m \) ways.
Ex. How many strings of length 5 from A, ..., Z are there?

\[
\begin{array}{cccccc}
26 & 26 & 26 & 26 & 26 \\
1 & 2 & 3 & 4 & 5 \\
\end{array}
\]

\[26^5 = 11,881,376\]

Ex. How many such strings have no repeated letter?

\[
\begin{array}{cccccc}
26 & 25 & 24 & 23 & 22 \\
1 & 2 & 3 & 4 & 5 \\
\end{array}
\]

\[26 \cdot 25 \cdot 24 \cdot 23 \cdot 22 = 7,893,600\]

Ex. How many bit strings of length \( n \) are there? (i.e. strings from the alphabet \( \{0, 1\} \))

\[
\begin{array}{cccccccc}
2 & 2 & 2 & 2 & 2 & \ldots & 2 \\
1 & 2 & 3 & \ldots & n \\
\end{array}
\]

\[2 \cdot 2 \cdot 2 \ldots 2 = 2^n\]

Ex. The number of strings of length \( n \) from an alphabet of size \( m \) is \( m^n \).

\[
\begin{array}{cccccccc}
m & m & m & \ldots & m \\
1 & 2 & 3 & \ldots & n \\
\end{array}
\]
EX: Determine the number of strings of length \( n \) from an alphabet of size \( m \), in which no symbol is repeated.

\[
\frac{m}{1} \frac{m-1}{2} \frac{m-2}{3} \cdots \frac{m-n+1}{n}
\]

\[
m(m-1)(m-2) \cdots (m-n+1) = \frac{m!}{(m-n)!}
\]

**Theorem**

If \( A, B \) are finite sets, then so is \( A \times B \), and \( |A \times B| = |A| \cdot |B| \)

**Proof**

To form an ordered pair \((x,y)\) of \( A \times B \) one first chooses \( x \in A \) in one of \(|A|\) ways, then chooses \( y \in B \) in any of \(|B|\) ways.

**Theorem**

If \( S \) is a finite set then so is \( \mathcal{P}(S) \) and \( |\mathcal{P}(S)| = 2^{|S|} \).

**Proof:**

The task of constructing a subset \( A \subseteq S \) breaks naturally into \(|S|\) subtasks:

For each \( x \in S \), decide whether or not \( x \in A \). Since each subtask can be
PERFORMED IN 2 WAYS (x \in A or x \notin A), THERE ARE

\[ 2 \cdot 2 \cdot 2 \cdots 2 = 2^{181} \]

WAYS TO CONSTRUCT \( A \subseteq S \). THUS THERE ARE \( 2^{181} \) SUBSETS OF \( S \).

Remark: Suppose \( |S| = n \), say \( S = \{x_1, x_2, \ldots, x_n\} \). Then there is a bijection:

\[ f : \mathcal{P}(S) \to \{\text{bit strings of length } n\} \]

Define as follows: For \( A \subseteq S \), let \( f(A) \) be the bit string whose \( i \)-th bit is 1 IFF \( x_i \in A \).

Ex. \( S = \{1, 2, 3, 4\}, \mathcal{P}(S) = \emptyset, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\} \)

\[
\begin{align*}
    f(\emptyset) &= 000 \\
    f(\{1\}) &= 100 \\
    f(\{2\}) &= 010 \\
    f(\{3\}) &= 001 \\
    f(\{1, 2\}) &= 110 \\
    f(\{1, 3\}) &= 101 \\
    f(\{1, 4\}) &= 011 \\
    f(\{2, 3\}) &= 011 \\
    f(\{2, 4\}) &= 011 \\
    f(\{3, 4\}) &= 011 \\
    f(S) &= 111
\end{align*}
\]
**Definition**

Let $A, B$ be sets. We denote by $B^A$ the set of functions with domain $A$ and codomain $B$. 

$$B^A = \{ f : A \to B \}$$

**Theorem**

If $A, B$ are finite sets, then so is $B^A$, and 

$$|B^A| = |B|^{|A|}$$

**Proof:**

Suppose $|A| = m$ and $|B| = n$. The task of constructing a function $f : A \to B$ breaks into $m$ successive subtasks: for each $x \in A$ choose one of the $n$ elements in $B$ as $f(x)$. Since there are $m$ subtasks, each with $n$ alternatives, the product rule says there are 

$$n \cdot n \cdot n \cdot \ldots \cdot n = n^m$$

ways to construct such a function. Hence $|B^A| = n^m$ as required.
Ex. How many functions \( f : A \to B \) are injective?

Let \(|A| = m\), \(|B| = n\) as before, and suppose \( A = \{x_1, \ldots, x_m\}\). Note that if \( n < m \) there are no injective functions. If \( n \geq m \) the task of constructing an injective function \( f : A \to B \) breaks into \( m \) subtasks:

- Choose \( f(x_1) \in B \) : \( n \) ways
- Choose \( f(x_2) \in B \) \( \setminus \{ f(x_1) \} \) : \( (n-1) \) ways
- Choose \( f(x_3) \in B \) \( \setminus \{ f(x_1), f(x_2) \} \) : \( (n-2) \) ways

\[ \vdots \]

- Choose \( f(x_m) \in B \) \( \setminus \{ f(x_1), f(x_2), \ldots, f(x_{m-1}) \} \) : \( (n-m+1) \) ways

Thus the number of injective functions in \( B \) in this case is:

\[
{}^n \left( \begin{array}{c}
0 & \text{if } n < m \\
\frac{n!}{(n-m)!} & \text{if } n \geq m
\end{array} \right)
\]
**Sum Rule.**

Suppose a task $T$ can be performed by one of two subtasks $T_1$ or $T_2$, but not both. Suppose $T_1$ can be performed in $n_1$ ways, and $T_2$ in $n_2$ ways. Then $T$ can be performed in $n_1 + n_2$ ways.

Stated in terms of sets, the sum rule says that if $A, B$ are finite sets with $A \cap B = \emptyset$, then

$$|A \cup B| = |A| + |B|.$$

More generally, if $T$ can be performed by doing exactly one of the subtasks $T_1, \ldots, T_m$, and $T_i$ can be done in $n_i$ ways (i.e., $1 \leq i \leq m$), then $T$ can be performed in $n_1 + n_2 + \ldots + n_m$ ways.

In terms of sets we would say that if $A_1, \ldots, A_m$ are pairwise disjoint (i.e., $A_i \cap A_j = \emptyset$ for $i \neq j$) then

$$|A_1 \cup A_2 \cup \ldots \cup A_m| = |A_1| + |A_2| + \ldots + |A_m|.$$
Ex. How many strings of length at most 5 can be formed from an alphabet of size 12? (Include the empty string.)

We have 6 subtasks:
- Form all strings of length 5: \(12^5\) ways
- " " " " "  4: \(12^4\) ways
- " " " " "  3: \(12^3\) ways
- " " " " "  2: \(12^2\) ways
- " " " " "  1: \(12^1\) ways
- " " " " "  0: \(12^0\) ways.

By the sum rule there are
\[12^6 + 12^5 + 12^4 + 12^3 + 12^2 + 12 + 1 = \frac{12^7 - 1}{12 - 1} = 271,453\]
strings of length at most 5.

Ex. A computer system uses passwords consisting of 6 to 8 characters which are either upper or lower case alphabetic (52) or digits (10) subject to the following constraints: A password must contain at least 1 digit and at least 1 alphabetic character. How many valid passwords are there?
Let \( P \) be the number of such passwords, and \( P_i \) be the number of passwords of length \( i \) (\( i = 6, 7, 8 \)). Then \( P = P_6 + P_7 + P_8 \) by the sum rule, and

\[
P_6 = 62^6 - 52^6 - 10^6 \times (3.7) \times 10^{10}
\]

\[
P_7 = 62^7 - 52^7 - 10^7 \times (1.0) \times 10^{12}
\]

\[
P_8 = 62^8 - 52^8 - 10^8 \times (1.0) \times 10^{14}
\]

\[
P \approx (1.61037) \times 10^{14} \times 161 \text{ trillion}
\]

**Principle of Inclusion-Exclusion (PIE)**

Recall that if \( A, B \) are finite sets, not necessarily disjoint, then

\[
|A \cup B| = |A| + |B| - |A \cap B|
\]

Ex. How many bit strings of length 7 either begin with 2 0's or end with 3 1's?

Let \( A = \{00xxxxx\} \), \( B = \{xxxxxxx\} \), then \( A \cap B = \{00xx111\} \), and

\[
|A \cup B| = 2^5 + 2^4 - 2^2 = 44
\]
Recall the Euler totient function $\phi(n)$ is defined to be the number of numbers in $\{1, 2, \ldots, n\}$ which are relatively prime to $n$.

Ex. Determine $\phi(1000)$. (Note this was an earlier HW assignment.)

\[ \phi(1000) = 1000 - (\# \text{ of } #') \text{ not rel. pr. to } 1000 \]

A number is not relatively prime to 1000 iff it has a factor in common with 1000 = $2^3 \cdot 5^3$, i.e. divisible by 2 or 5.

\[ (\# \text{ of } #'s \ \text{divisible by } 2) = \frac{1000}{2} = 500 \]
\[ (\# \text{ of } #'s \ \text{divisible by } 5) = \frac{1000}{5} = 200 \]
\[ (\# \text{ of } #'s \ \text{divisible by both}) = \frac{1000}{2 \cdot 5} = \frac{1000}{10} = 100 \]

\[ \Rightarrow \phi(1000) = 1000 - (500 + 200 - 100) = 400 \]

Generalizations of PIP:

\[ |A_1 \cup A_2 \cup A_3| = (|A_1| + |A_2| + |A_3|) \]
\[ - (|A_1 \cap A_2| + |A_1 \cap A_3| + |A_2 \cap A_3|) \]
\[ + |A_1 \cap A_2 \cap A_3| \]
\[ \bigcup_{i=1}^{n} A_i = \sum_{i} |A_i| - \sum_{i, j} |A_i \cap A_j| + \sum_{i, j, k} |A_i \cap A_j \cap A_k| - |A_1 \cap A_2 \cap A_3 \cap A_4| \]

\[ \bigcup_{i=1}^{n} A_i = \sum_{i} |A_i| - \sum_{i, j} |A_i \cap A_j| + \sum_{i, j, k} |A_i \cap A_j \cap A_k| \]

\[ - \sum_{i, j, k, l} |A_i \cap A_j \cap A_k \cap A_l| + \ldots \]

\[ \ldots + (-1)^{n+1} |A_1 \cap A_2 \cap \ldots \cap A_n| \]

**Tree Diagrams**

Ex. How many bit strings of length 4 contain neither 3 consecutive 0's nor 3 consecutive 1's?

**Bit Strings:** \{0000, 0001, 0010, 0100, 1110, 1011, 0011, 1001, 0101, 1100\}

There are 10 such strings.