I. Basic Counting Principles: The Product Rule

A. Definition: Suppose that a procedure can be broken down into a sequence of two tasks, T_1 and T_2. If there are N_1 ways to do T_1 and for each of the N_1 ways of doing T_1 there are N_2 ways to do T_2 then there are $N_1 \times N_2$ ways to do the procedure.

B. Proof: (By Mathematical Induction on N_2)

1. Basis Step:

   a. Assume $N_2 = 1$

   b. Therefore, for each of the N_1 ways in which T_1 can be accomplished, there is only 1 way in which T_2 can be accomplished.

   c. Therefore there are $N_1 \times 1 = N_1$ ways in which the procedure composed of tasks T_1 and T_2 can be accomplished.

2. Inductive Assumption:

   If a procedure can be broken down into a sequence of two tasks, T_1 and T_2 and if there are N_1 ways to do T_1 and, for each of these N_1 ways of doing T_1 there are N_2 ways to do T_2, then there are $N_1 \times N_2$ ways to complete the procedure.

3. Inductive Step:

   a. Let $X(N_1, N_2) = N_1 \times N_2$ be the number of ways to complete the procedure composed of T_1 and T_2 when there are N_1 ways to complete T_1 and N_2 ways to complete T_2

   b. If we add another alternative for the completion of task T_2 then for every completion of T_1 there is an additional path for completion of the entire task.

c. Therefore:

$$X(N_1,\ N_2{+}1) = X(N_1,\ N_2) + N_1$$
$$= N_1 \times N_2 + N_1$$
$$= N_1 \times (N_2 + 1)$$

4. Therefore: $X(N_1,\ N_2) = N_1 \times N_2$

C. Example: How many different bit strings of length 2 can be generated?

1. Task 1 is choosing the position in which to place a bit.

2. Task 2 is choosing the bit, either $'0'B$ or $'1'B$, that goes into the chosen position.

3. The number of alternatives for completing Task 1 is $2 = N_1$.

4. The number of alternatives for completing Task 2 is $2 = N_2$.

5. Therefore the number of ways in which the entire task can be completed is $N_1 \times N_2 = 2 \times 2 = 4$

6. The possible bit strings are:
   1. $[0,\ 0]$
   2. $[0,\ 1]$
   3. $[1,\ 0]$
   4. $[1,\ 1]$

II. Generalization of the Product Rule

A. Definition: Suppose that a procedure can be broken down into a sequence of N tasks, each one of which can be performed in M ways. Then there are M^N ways in which this procedure can be completed.

B. Proof: (By Mathematical Induction on N)

1. Basis Step: Assume $N = 1$
   a. Therefore we have one task which can be completed in M ways.

   b. Therefore there are $M = M^1 = M^N$ ways in which the procedure can be completed.

2. Inductive Assumption:
   If a procedure can be broken down into a sequence of N tasks, each one of which can be performed in M ways, then there are M^N ways in which this procedure can be completed.

3. Inductive Step:
   a. Let $X(N) = M^N$ be the number of ways to complete the procedure composed of N tasks when there are M ways to complete each task.

   b. If we add another task that must be completed then we have M more alternatives for the completion of that task and, hence, for the completion of the total procedure.

   c. If we consider the procedure of N tasks to be a single task, and we are adding M more alternatives, then by the product rule we have that:

   $$X(N+1) = X(N) \times M = M^N \times M = M^{N+1}$$

4. Therefore if a procedure can be broken down into a sequence of N tasks, each one of which can be performed in M ways, then there are M^N ways in which this procedure can be completed.

C. Example: How many different bit strings of length N can be generated?

1. The N tasks involve choosing a bit to place in each of the N positions in the string.

2. Task of choosing the bit, either $'0'B$ or $'1'B$, that goes into the chosen position can be performed in $M = 2$ different ways.

3. Therefore the number of different bit strings of length N that can be generated is: $M^N = 2^N$

4. The largest value that can be contained in a bit string of length N is $2N - 1$. If we add on the bit string that represents 0 we have a total of 2^N strings.

III. Another Generalization of the Product Rule

A. Definition: Suppose that a procedure can be broken down into a sequence of N tasks. Each task T_i can be completed in M_i different ways. Then the number of ways K in which the procedure may be completed is

$$K = M_1 \times M_2 \times M_3 \times ... \times M_N = \prod_{i=1}^{N} M_i$$

B. Proof: (By Mathematical Induction on N)

1. Basis Step: Assume $N = 1$

   a. Therefore we have one task T_1 which can be completed in M_1 different ways.

   b. Therefore $K = M_1 = \prod_{i=1}^{1} M_i$ is the number of ways in which the procedure can be completed.

2. Inductive Assumption:

   If a procedure can be broken down into a sequence of N tasks and each task T_i can be completed in M_i different ways then the procedure may be completed is K where:

$$K = M_1 \times M_2 \times M_3 \times ... \times M_N = \prod_{i=1}^{N} M_i$$

3. Inductive Step:

   a. Let $K_N = \prod_{i=1}^{N} M_i$ be the number of ways in which a procedure involving N tasks T_i for which each task T_i can be completed in M_i different ways can be completed.

   b. Consider the task with N sub-tasks to be a single task.

   c. If we add another task T_{N+1} that can be completed in M_{N+1} different ways to the procedure then we have M_{N+1} more alternatives for the completion of the total procedure.

   d. By the product rule, then, the number of ways in which this new task can be completed is the product of the number of ways the original task can be completed and the number of ways the added project can be completed.

   e. Therefore: $$K_{N+1} = K_N \times M_{N+1} = \prod_{i=1}^{N} M_i \times M_{N+1}$$ $$= \prod_{i=1}^{N+1} M_i$$

4. Therefore if a procedure can be broken down into a sequence of N tasks, each one of which can be performed in M_i ways, then there are $\prod_{i=1}^{N} M_i$ ways in which this procedure can be completed.

C. Example: How many permutations of a list of N distinct elements can be constructed?

1. Task T_1 is the selection of the first element of a permutation.

   Since there are N elements in the list the task T_1 can be accomplished in any one of N different ways.

2. Task T_2 is the selection of the second element of a permutation.

   Since there are N elements in the list and:

   a. We have already chosen one of them to be the first element in the permutation.

   b. There are $N-1$ elements from which to choose the second element of the permutation.

   the task T_2 can be accomplished in any one of $N-1$ different ways.

3. We proceed in this manner to task T_N, the choice of the Nth, or last, element of the permutation.

   a. We have already chosen $N-1$ elements of the permutation leaving only one element of the list unchosen.

   b. Therefore the task T_N can be completed in only one manner.

4. Therefore, according to the product rule, the task of creating a permutation of a list of N elements can be completed in any one of $\prod_{i=1}^{N} M_i$ ways where:

   $$\prod_{i=1}^{N} M_i = N \times (N-1) \times (N-2) \times ... \times 3 \times 2 \times 1 = N!$$

IV. Proof by Mathematical Induction: There are $N!$ possible permutations of a list of N elements.

A. Basis Step: $N = 1$

1. There is only one possible permutation of a list with only 1 element.

2. $1! = 1 = N!$

B. Inductive Assumption: There are $N!$ possible permutations of a list of N elements.

C. Inductive Step:

1. Consider a list L of N elements with
$$L = \; < a_1, \, a_2, \, a_3, \dots, a_N >$$

2. If we wish to add a new element, the $(N+1)$th, to the list we can:

a. Add the new element at the beginning of the list generating a second ordering of the new list
$L_1 = \; < a_{N+1}, \, a_1, \, a_2, \, a_3, \dots, a_N >$.

b. Add the new element between each pair of existing elements of the list, creating $N-1$ orderings for the new list, as in:
$$L_2 = \; < \; a_1, \; a_{N+1}, \, a_2, \, a_3, \dots, a_N >$$
$$L_3 = \; < \; a_1, \; a_2, \, a_{N+1}, \, a_3, \dots, a_N >$$
$$L_4 = \; < \; a_1, \; a_2, \, a_3, \, a_{N+1}, \dots, a_N >$$
$$\vdots$$
$$L_N = \; < \; a_1, \; a_2, \, a_3, \dots, a_{N+1}, \, a_N >$$

c. Add the new element at the end of the list for the first ordering of the new list or $N+1$ elements generating
$L_{N+1} = \; < a_1, \, a_2, \, a_3, \dots, a_N, \, a_{N+1} >$.

d. Therefore, from a single list of N elements we have created $N+1$ orderings of $N+1$ elements.

3. If we repeat this process for each of the $N!$ permutations of the list of N elements we will have generated $N \times N!$ new orderings of a list of $N+1$ elements.

4. Therefore, assuming that a list of N elements has $N!$ permutations or orderings leads to the conclusion that a list of $N+1$ elements has $N!$ orderings or permutations.

D. Since the statement: There are $N!$ possible permutations of a list of N elements.

is true for $N = 1$ and the assumption that it is true for and arbitrary value of N leads to the conclusion that it is true for $N+1$ the statement has been proven true for all N by the Principle of Mathematical Induction.

V. Erroneous Application of the Product Rule

A. Problem:

Three offices; a president, a treasure, and a secretary; are to be filled from a slate of four candidates. The candidates are Ann, Bart, Cyd, and Dan. For undisclosed reasons Ann cannot be president and either Cyd or Dan must be secretary. In how many ways can the offices be filled?

B. Incorrect Solution:

1. There are three choices for president: Bart, Cyd, and Dan.

2. Any person except that chosen for president can be chosen for treasurer, or three choices.

3. There are two choices for secretary, Cyd or Dan.

4. Therefore, by the product rule, there are $3 \times 3 \times 2 = 18$ possibilities for filling the slate.

5. If Bart is chosen for president and Ann for treasurer then there are two choices for secretary, and we have $3 \times 3 \times 2 = 18$ alternatives for filling the slate as given by the product rule.

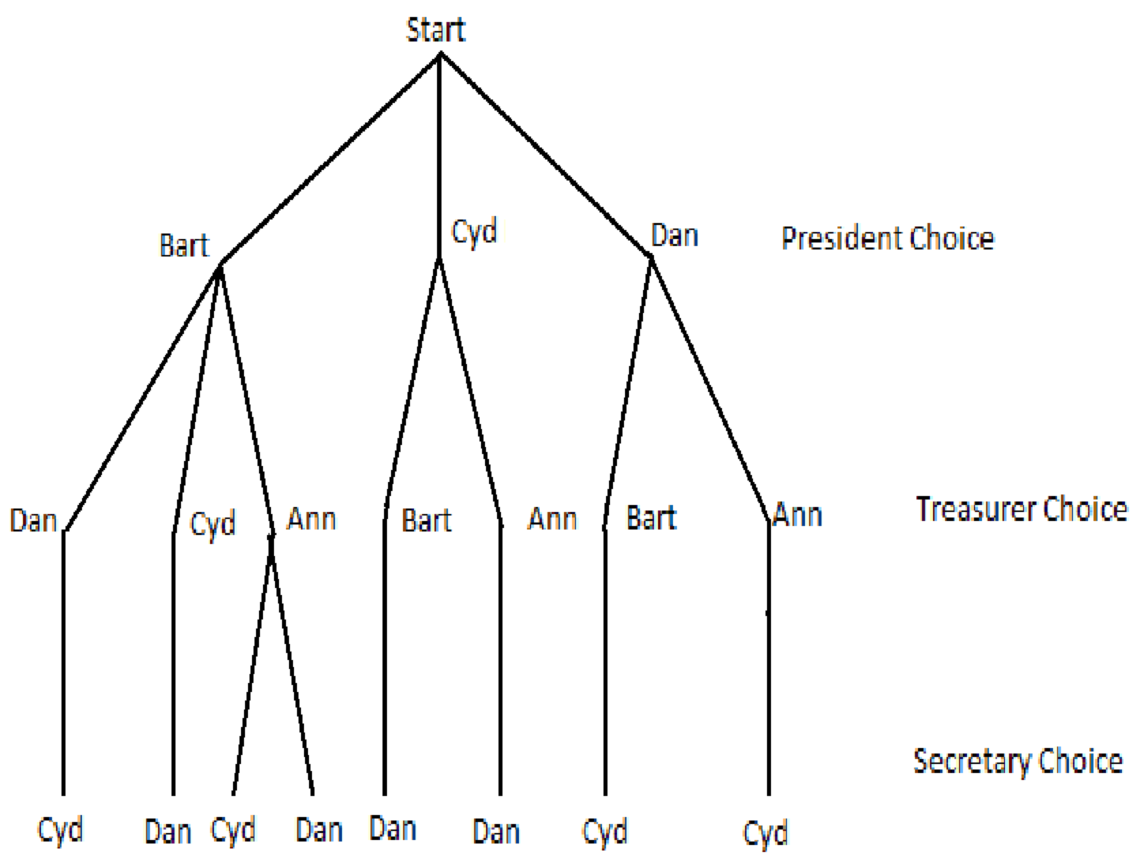
C. Error:

1. If Bart is chosen for president and either Cyd or Dan for treasurer then either Dan or Cyd must be chosen for secretary, generating $8 \neq 18$ possibilities for filling the slate.

2. If Cyd is chosen for president and either Ann or Bart for treasurer then there is just one choice for secretary, i.e., Dan, generating $8 \neq 18$ possibilities for filling the slate.

3. If Dan is chosen for president and either Ann or Bart for treasurer then there is just one choice for secretary, i.e., Dan, generating $8 \neq 18$ possibilities for filling the slate.

D. Analysis: The product rule was used incorrectly in this case. The chart below shows the 8 possibilities.



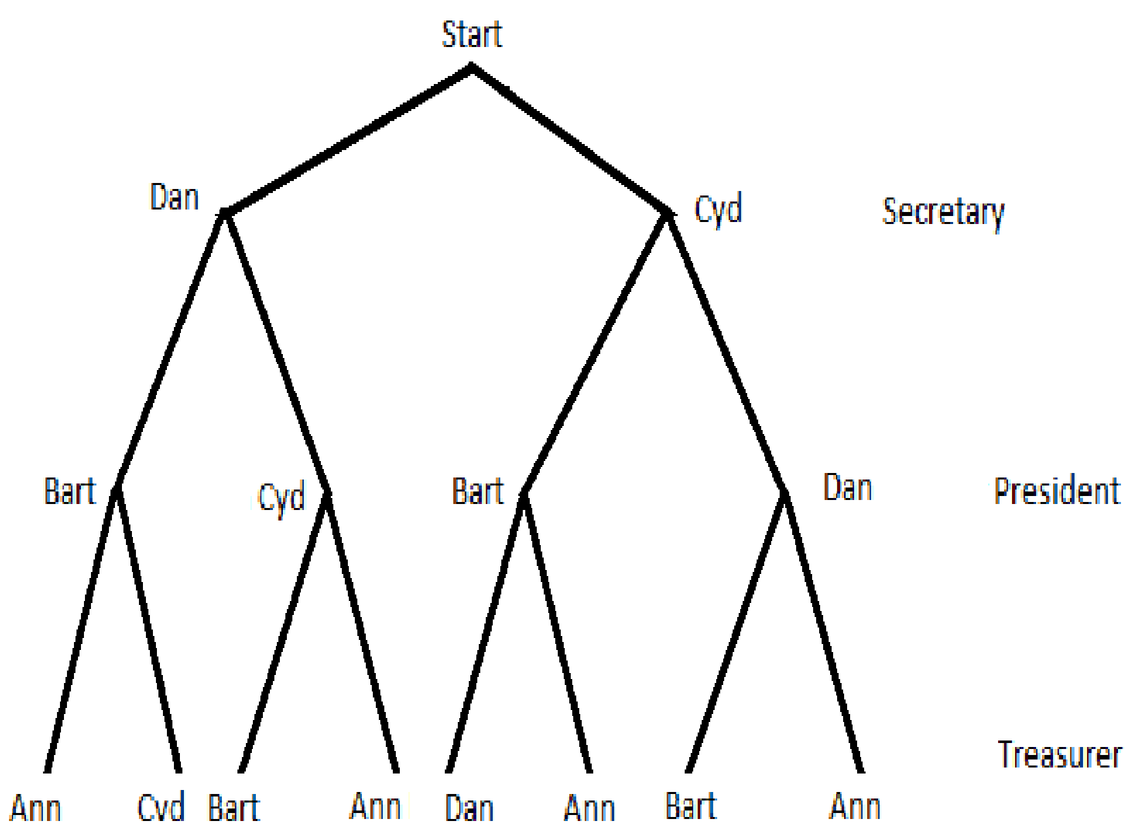
E. Remedy: Start the selection with the position with the most restrictions.

1. Choose the secretary from either Dan or Cyd, giving two choices.

2. Choose the president from the remaining candidates, i.e., Bart and Dan or Bart and Cyd (depending on which of Dan and Cyd was chosen for secretary) giving two choices since Ann is not a candidate.

3. Choose the treasurer from Ann, Bart, Cyd, or Dan depending on which have not already been chosen for office. Since Ann cannot be president and the person chosen for secretary cannot be treasurer we have two choices remaining.

4. Therefore there are $2 \times 2 \times 2 = 8$ possible choices according to the product rule (see diagram below).



VI. Summation Rule

A. Definition: If a task can be done in one of N_1 ways ***OR*** in one of N_2 ways and none of the set of N_1 ways is the same as any of the set of N_2 ways then there are $N_1 + N_2$ ways to accomplish the task.

B. Alternative Definition:
Suppose a finite set A equals the union of N distinct mutually disjoint subsets $A_1,\ A_2,\ A_3,\ ...,\ A_N.$

Then: $|A| = |A_1| + |A_2| + |A_3| + \ ... + |A_N|$

C. Example: How many passwords contain three or fewer letters?

1. The term letter is assumed to mean a lower case letter of the English alphabet.

2. The set S of all passwords of length less than or equal to three can be separated into three mutualy disjoint subsets:
   a. $S_1 =$ the set of all passwords of length one.

   b. $S_2 =$ the set of all passwords of length two.

   c. $S_3 =$ the set of all passwords of length three.

3. We then have:
   a. $|S_1| = 26$

   b. $|S_2| = 26^2$ (from the product rule).

   c. $|S_3| = 26^3$ (again, from the product rule).

4. Therefore: $|S| = |S_1| + |S_2| + |S_3| = 18,278$

VII. Inclusion/Exclusion or Subtraction Principle or Difference Rule

A. Definition (for two and three sets):

If A, B, and C are any finite sets then:

1. $|A \cup B| = |A| + |B| - |A \cap B|$

2. $|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| - |A \cap B \cap C|$

B. Alternative Statement (for two sets):

If a task can be done in either N_1 ways or in N_2 ways then the number of ways to do the task is $N_1 + N_2$ minus the number of ways to do the task that are common to the two different ways.

C. Application: A professor in a discrete mathematics class passes out a form asking students to check all the mathematics and computer science courses they have recently taken. The results were that out of a total of 50 students in the class:

30	took precalculus	$=$	P
18	took calculus	$=$	C
26	took Java	$=$	J
9	took both precalculus and calculus	$=$	$P \cap C$
16	took both precalculus and Java	$=$	$P \cap J$
8	took both calculus and Java	$=$	$C \cap J$
47	took at least one of the three courses.	$=$	S

1. Problem 1: How many students did not take any of the three courses?

   a. By the difference rule, the number of students who did not take any of the three courses equals the number in the class minus the number who took at least one course.

   b. Therefore: $50 - 47 = 3$students did not take any of the three courses.

2. Problem 2: How many students took all three courses?

   a. The answer to this question is $|P \cap C \cap J|$

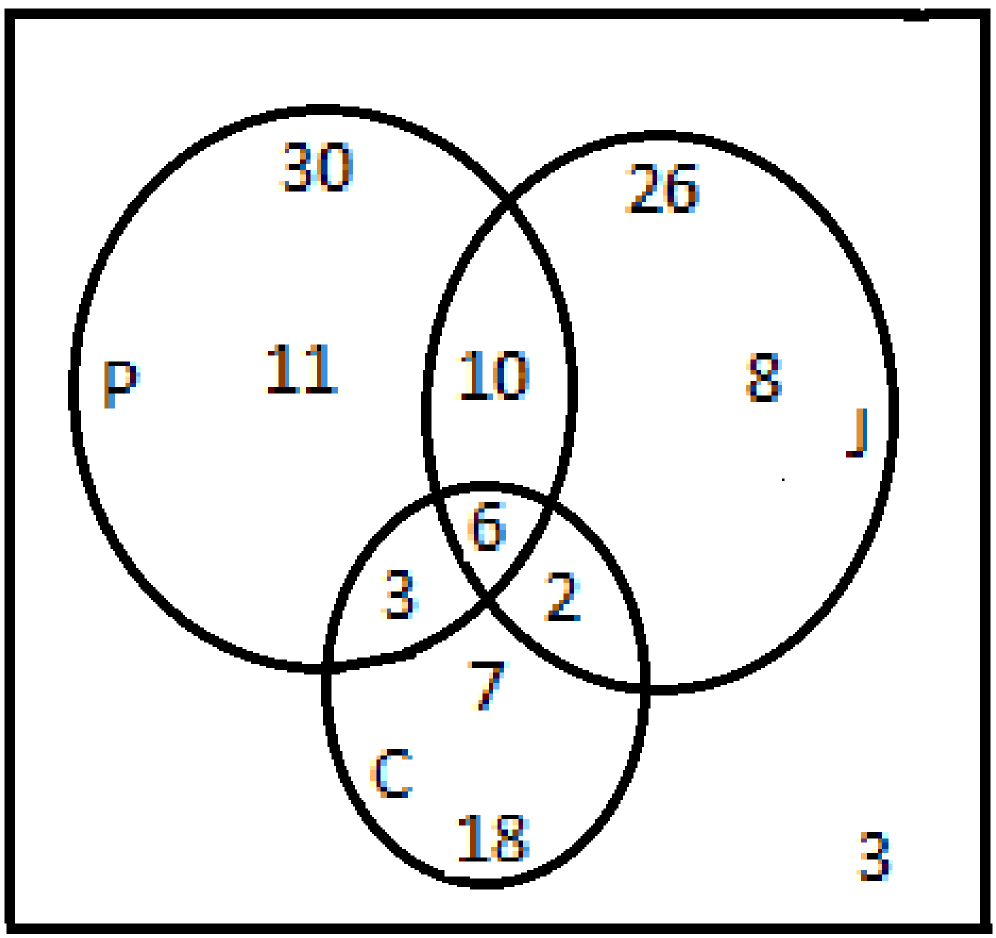
   b. By the inclusion/exclusion rule we have that:

   $$|P \cup C \cup J| = |P| + |C| + |J| - |P \cap C| - |P \cap J| - |C \cap J| + |P \cap C \cap J|$$

   c. Then: $$47 = 30 + 26 + 18 - 9 - 16 - 8 + |P \cap C \cap J| = 41 + |P \cap C \cap J|$$

   d. Therefore: $|P \cap C \cap J| = 6$

3. Refer to the Venn Diagram below for the remainder of the questions.



4. Loading the values:

   a. Since $|P \cap C \cap J| = 6$ we have that six students took all three courses.

   b. Since nine students took both calculus and precalculus and six took all three courses the number of students who took calculus and precalculus but not Java must be:

   $$|P \cap C| - |P \cap C \cap J| = 9 - 6 = 3$$

   c. Since 16 students took precalculus and Java and six took all three courses the number of students who took calculus and Java but not precalculus must be:

   $$|P \cap J| - |P \cap C \cap J| = 16 - 6 = 10$$

   d. Since eight students took calculus and Java and six took all three courses the number of students who took calculus and Java but not precalculus must be:

   $$|C \cap J| - |P \cap C \cap J| = 8 - 6 = 2$$

5. How many students took precalculus but not calculus and not Java?

   $$|P| - |P \cap J| - |P \cap C| + |P \cap C \cap J|$$
   $$= 30 - 9 - 16 + 6 = 11$$

6. How many students took Java but not precalculus and not calculus?

   $$|J| - |P \cap J| - |C \cap J| + |P \cap C \cap J|$$
   $$= 26 - 16 - 8 + 6 = 8$$

7. How many students who took calculus but not precalculus and not Java is:

   $$|C| - |P \cap C| - |C \cap J| + |P \cap C \cap J|$$
   $$= 18 - 9 - 8 + 6 = 7$$