

Application of Inclusion-Exclusion Principle in Solving The Famous Hat-Check Problem

Muhammad Hasan 13518012
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13518012@std.stei.itb.ac.id

Abstract—There are many problems regarding counting that can be solved by the inclusion-exclusion principle, one of those problems is The famous hat-check problem, a problem that asks for the probability that no person is given the correct hat back by a hat-check person who gives the hat back randomly. This problem is related to a term in combinatorial mathematics that is called derangement.

Keywords—Combinatorics, Inclusion-Exclusion Principle, Derangements, Counting.

I. INTRODUCTION

A lot of interesting problems that includes counting can be solved by The inclusion-exclusion principle, a counting technique in combinatorics (combinatorial mathematics) which generalizes the familiar method of obtaining the number of elements in a two or more sets. A simple application of the inclusion-exclusion principle is finding the number of elements in the union of two finites sets. The inclusion-exclusion principle can also be applied to more complex problem, one of which is the famous hat-check problem. A problem that asks for the probability that no person is given the correct hat back by a hat-check person who gives the hat back randomly. This hat-check problem is related to a term in combinatorics which is called a derangement. Derangement is a permutation of elements in a set in which there are no element that appear in its original position, it can also be said that a derangement is a permutation that has no fixed points. In terms of solving derangement problems there are two option of familiar solutions, one is using recursion and the other one is using inclusion-exclusion principle. This paper will mainly discuss the inclusion-exclusion principle way of solving problem related to derangement.

II. BASIC THEORY

It is important to review the basic theory that is related on the later explanation of this paper. This will help in gaining a clear understanding of what to be explained.

2.1 Set Theory

In this set theory, there will be the explanation of sets regarding definition, membership, cardinality, and basic operations.

2.1.1 Definition

In mathematics, a *set* is defined as a well-defined collection of distinct objects, considered as an object in its own right.^[1] Sets are often specified with curly brace notation. The set of even integers can be written:

$$\{2n : n \text{ is an integer}\}$$

The opening and closing curly brackets denote a set. $2n$ specifies the member of the set the colon says “such that” or “where” and everything following the colon are conditions that explain or refine the membership.^[2] The objects that make up a set is not only elements made of number, it can be anything as long as they are well-defined distinct objects. For an example the set {red, green, blue} is also a valid set. A set can also have no object or elements in it, this is called a *empty set* which is usually denoted by $\{\}$ or \emptyset . Sets are conventionally denoted with capital letters. For an example, the set A with elements 1, 2, 3, and 4 can be written as:

$$A = \{1, 2, 3, 4\}$$

It is important to note that because the elements in a set is distinct we cannot say for an example $\{1, 1, 3\}$ as a set, because number 1 as an element of the set occurs twice in the set.

2.1.2 Membership

If A is a set and x is one of the elements in A , then we can denote the symbol $x \in A$ to be understood as “ x is an element of the set A ”.^[1] On the contrary, if there is an object y which is not an element of the set A we can denote it with the symbol $y \notin A$. For example, if $A = \{1, 3, 5, 8, 9\}$ then we can say $1 \in A$, $3 \in A$, and $4 \notin A$.

If every element of set A is also in set B , then A is said to be a *subset* of B , written $A \subseteq B$ (pronounced A is contained in B). The relationship between sets established by \subseteq is called *inclusion* or *containment*. We can say the two sets is equal if they contain each other:

$$A = B \leftrightarrow A \subseteq B \text{ and } B \subseteq A$$

Where A and B are well defined sets. It can also be said that two

sets are equal if and only if they have precisely the same elements.

2.1.3 Cardinality

The *cardinality* of a set S , denoted $|S|$, is the number of members in S .^[3] For example, the set $A = \{\text{red, green, blue}\}$ has 3 elements, so we can say $|A| = 3$. The cardinality of an empty set is zero. Some sets have infinite cardinality. The set N of natural numbers for example has infinite cardinality.

2.1.4 Basic Operations

There are several fundamental operations for constructing new sets from given sets:

1) Unions

Two sets can be “added” together. The *union* of set A and B , denoted by $A \cup B$, is the set of all things that are a member of either A or B .

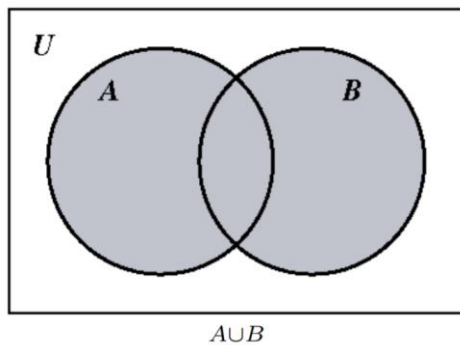


Figure 1. Union of set A and B depicted with the Venn's Diagram.^[2]

Here are some of the example of unions:

- $\{1, 2\} \cup \{1, 2\} = \{1, 2\}$
- $\{1, 2\} \cup \{2, 3\} = \{1, 2, 3\}$
- $\{1, 2, 3\} \cup \{3, 4, 5\} = \{1, 2, 3, 4, 5\}$

There are also some basic properties of unions:

- $A \cup B = B \cup A$
- $A \cup (B \cup C) = (A \cup B) \cup C$
- $A \cup A = A$
- $A \cup \emptyset = A$

2) Intersections

A set can also be constructed by determining which members two sets have in “common”. The *intersection* of A and B , denoted by $A \cap B$, is the set of all things that are members of both A and B . If $A \cap B = \emptyset$, then A and B is called to be *disjoint*.

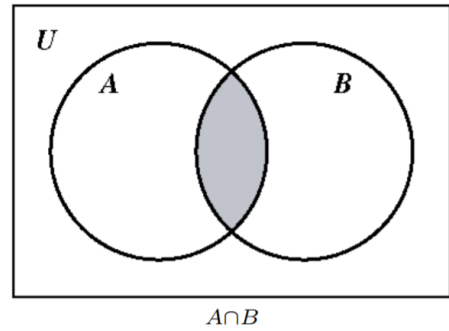


Figure 2. Intersection of set A and B depicted with the Venn's Diagram.^[2]

Here are some of the example of intersections:

- $\{1, 2\} \cap \{1, 2\} = \{1, 2\}$
- $\{1, 2\} \cap \{2, 3\} = \{2\}$
- $\{1, 2\} \cap \{3, 4\} = \emptyset$

There are also some basic properties of intersections:

- $A \cap B = B \cap A$
- $A \cap (B \cap C) = (A \cap B) \cap C$
- $A \cap A = A$
- $A \cap \emptyset = \emptyset$

3) Complements

Two sets can also be “subtracted”. The *relative component* of B in A (also called the *set-theoretic difference* of A and B), denoted by $A - B$, is the set of all the elements that are members of A but are not members of B . It is only valid to “subtract” members of a set there are not in the set, such as removing the element *red* in the set $\{1, 3, 5\}$; doing so has no effect. In certain settings all sets under discussion are considered to be subsets of a given *universal set* U . In such case. $U - A$ is called the *absolute component* or simply called the *complement* of A , and is denoted by A^c .

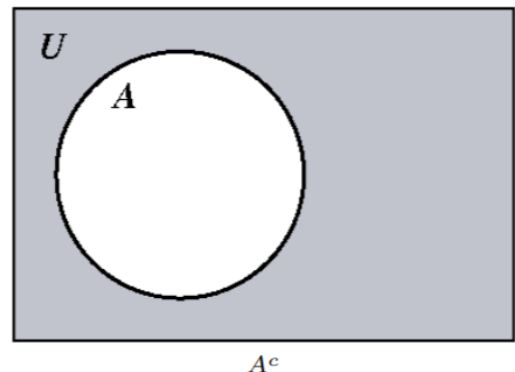


Figure 3. The complement of set A (A^c) depicted with the Venn's Diagram.^[2]

Here are some of the example of complements:

- $\{1, 2\} - \{1, 2\} = \emptyset$
- $\{1, 2, 3, 4\} - \{1, 3\} = \{2, 4\}$
- $\{1, 3, 5, 8\} - \{1, 5, 9\} = \{3, 8\}$

There are also some basic properties of intersections:

- $A - B \neq B - A$ for $A \neq B$
- $A - A^c = U$
- $A - A = \emptyset$
- $(A^c)^c = A$
- $\emptyset - A = \emptyset$
- $A - \emptyset = A$
- $A - A = \emptyset$
- $A - B = A \cap B^c$

2.2 The Inclusion-Exclusion Principle

In combinatorial mathematics, *The inclusion-exclusion principle* or *Principle of Inclusion and Exclusion* (PIE) is a counting technique that computes the number of elements that satisfy at least one of several properties while guaranteeing that elements satisfying more than one property not counted twice.^[4] This name comes from an idea that principle is based on over-generous *inclusion*, followed by compensating *exclusion*. This concept is attributed to Abraham de Moivre (1718).^[5]

An underlying idea behind PIE is that summing the number of elements that satisfy at least one category and subtracting the overlap prevents *double counting*. For an example, the number of people that have at least one cat or at least one dog can be found by taking the number of people who own a cat, adding the number of people that have a dog, then subtracting the number of people who own both.

In the case of objects being separated into two (possibly disjoint) sets, the principle of inclusion-exclusion states

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

where $|S|$ denotes the cardinality of the set S in set notation. As a Venn diagram, PIE for two sets can be depicted easily:

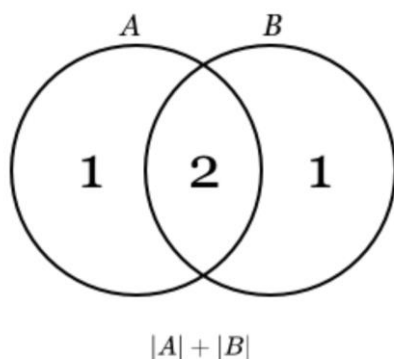


Figure 4. The depiction of $|A| + |B|$ in Venn's Diagram with numbers showing in each subset that represents how many times the subset has been counted.^[4]

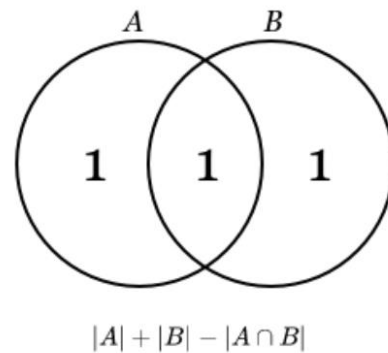


Figure 5. The depiction of $|A| + |B| - |A \cap B|$ in Venn's Diagram with numbers showing in each subset that represents how many times the subset has been counted.^[4]

The inclusion-exclusion principle can also be applied with more than two sets, in the case of three sets the PIE states:

$$|A \cup B \cup C|$$

is equal to

$$|A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

We can verify these statements for ourselves by considering the Venn diagram events:

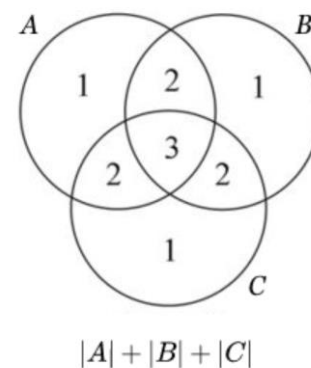


Figure 6. The depiction of $|A| + |B| + |C|$ in Venn's Diagram with numbers showing in each subset that represents how many times the subset has been counted.^[4]

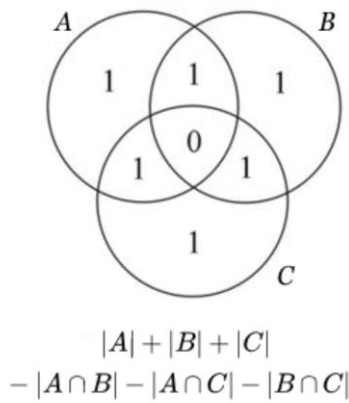


Figure 7. The depiction of $|A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C|$ in Venn's Diagram with numbers showing in each subset that represents how many times the subset has been counted.^[4]

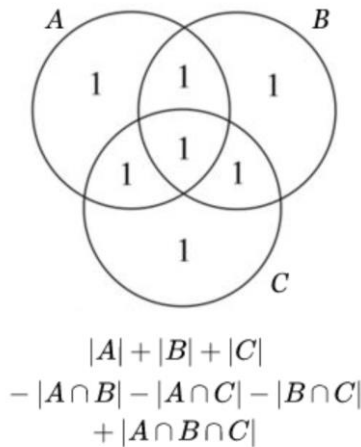


Figure 8. The depiction of $|A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$ in Venn's Diagram with numbers showing in each subset that represents how many times the subset has been counted.^[4]

More generally, if $A_1, A_2, A_3, \dots, A_n$ are finite sets, then the principle of inclusion and exclusion states:

$$\left| \bigcup_{i=1}^n A_i \right| = \sum_{i=1}^n |A_i| - \sum_{1 \leq i < j \leq n} |A_i \cap A_j| + \dots + (-1)^{n-1} |A_1 \cap \dots \cap A_n|$$

with this formula we can now have a general way of solving problems related to counting.

2.3 Derangement

In combinatorial mathematics, derangements are arrangements of some elements in a set so there is no element appears in its original position. The number of derangements of a set of size n is known as the *subfactorial* of n or the n -th *derangement number* or the n -th de Mornmort number. Notations for subfactorials in common use include $!n$, D_n , or d_n .^[6] For better understanding, take as an example when $n = 4$ and the set is $A = \{1, 2, 3, 4\}$, then we will have 9 set from

permutations of A , where every elements does not have the same position in its original set:

- 1) $\{2, 1, 4, 3\}$
- 2) $\{2, 3, 4, 1\}$
- 3) $\{2, 4, 3, 1\}$
- 4) $\{3, 1, 4, 2\}$
- 5) $\{3, 4, 1, 2\}$
- 6) $\{3, 4, 2, 1\}$
- 7) $\{4, 1, 2, 3\}$
- 8) $\{4, 3, 1, 2\}$
- 9) $\{4, 3, 2, 1\}$

So we can say that the number 4-th Derangement number is 9, or to simply put $!4 = 9$.

To find the number of derangements, one can probably just use brute force all the way on n elements with $n!$ tries, but of course that will be to tedious, so in order to find that we could use The inclusion-exclusion principle, but for now it is sufficient to only know the meaning of derangement. The part in using the inclusion-exclusion principle will be explained later on.

III. SOLVING THE FAMOUS HAT-CHECK PROBLEM

The famous hat-check problem goes by many name (originally described by Montmort in 1713).^[7] This problem is generally described as:

A group of n men enter a restaurant and check their hats. The hat-checker is absent minded, and upon leaving, she redistributes the hats back to the men at random. What is the probability P_n that no men gets his correct hat ?

by the description above, we could see quite clearly that this problem is very similar to that of derangement, in fact to answer the probability P_n we might have to only find D_n , the n -th Derangement, divided with every possibility which is $n!$ so we could state:

$$P_n = \frac{D_n}{n!}$$

so now the only problem is how to find D_n , and that is where The inclusion-exclusion principle comes in place. Without any further ado, let's get to the solution.

Let N denote the total number of permutations of n hats. To calculate the number of derangements, D_n , we want to exclude all permutations possessing any of the attributes a_1, a_2, \dots, a_n where a_i is the attribute that man i gets his correct hat for all i , such that $1 \leq i \leq n$. Let $N(i)$ denote the number of permutations possessing attribute a_i (and possibly others), $N(i, j)$ the number of permutations possessing attribute a_i and a_j (and possibly others), and so on. Then the inclusion-exclusion principle will state that:

$$D_n = N - \sum_{1 \leq i \leq n} N(i) + \sum_{1 \leq i, j \leq n} N(i, j) - \dots + (-1)^n N(1, 2, \dots, n)$$

By symmetry, we could see that

$$N(1) = N(2) = \dots = N(i).$$

and also

$$N(1, 2) = N(1, 3) = \dots = N(i, j)$$

and so on. Because of that, we will have:

$$D_n = N - \binom{n}{1}N(1) + \binom{n}{2}N(1, 2) - \dots + (-1)^n \binom{n}{n}N(1, 2, \dots, n)$$

Now, $N(1)$, the number of permutation where man 1 gets his correct hat, is simply $(n-1)!$, since the remaining hats can be distributed in any order. Similarly, we would also have that $N(1, 2) = (n-2)!$, $N(1, 2, 3) = (n-3)!$, and so forth.

Therefore, we now have:

$$D_n = N - \binom{n}{1}(n-1)! + \binom{n}{2}(n-2)! - \dots + (-1)^n \binom{n}{n}(n-n)!$$

Replacing N , the total permutation for n hats by $n!$, and we will have our final expression:

$$D_n = n! \left(1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \dots + (-1)^n \frac{1}{n!} \right)$$

or in a short way, we could say:

$$D_n = n! \sum_{i=0}^n \frac{(-1)^i}{i!}$$

With this expression, we will have our final answer for P_n that is:

$$P_n = \frac{D_n}{n!} = \left(1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \dots + (-1)^n \frac{1}{n!} \right)$$

Now, it's actually interesting to see that this is very similar to the series approaching $1/e$ as n approaches infinity.^[7]

IV. IMPLEMENTATION CODE IN C++

After having the final expression, it's quite easy to find the result with programming. But for that sake matter, it's probably not that interesting to only find the final result, in this part we will see also on how to see all the permutations in the given n -th Derangement.

Now, we will begin with finding the probability for the famous hat-check problem, we already know that the solution for the problem has a straightforward formula, so it will be very easy to implement in C++. Here is the implementation:

```
#include <bits/stdc++.h>

using namespace std;

int main() {
```

```
    int n;
    cout << "Input the number of hats : ";
    cin >> n;
    long double fact[n + 1];
    fact[0] = 1.0;
    for (int i = 1; i <= n; i++) {
        fact[i] = (long double) i * fact[i - 1];
    }
    long double Pn = 0;
    for (int i = 0; i <= n; i++) {
        Pn += (i & 1 ? -1 / fact[i] : 1 / fact[i]);
    }
    cout << "The answer for P(" << n << ") is : ";
    cout << fixed << setprecision(9) << Pn << '\n';

    return 0;
}
```

After compiling and executing the code, we can have the answer for $n = 10$:

Input the number of hats : 10
The answer for P(10) is : 0.367879464

Figure 9. The result of P_{10} of the hat-problem checker using the C++ implementation code.

We could see that for a bigger n , we might have some problem in precision, because the *long double* on C++ does not handle really large number and can only hold up to a number of $2^{64} - 1$. So it might only be precise for $n \leq 20$.

Now, we will continue with finding the permutation in the given n -th Derangement. Of course, it's quite hard to find a direct answer, so I will only use *brute-force* to find the permutation. Luckily, the chosen C++ programming language has very good *Standard Template Library* (STL) that we could work with, so it is actually quite easy to implement. For the sake of simplicity, we will see permutations on the first n number only, so if $n = 4$ as an example we will have the set $\{1, 2, 3, 4\}$, and so goes for any other value of n . Without any more hesitation, here is the implementation:

```
#include <bits/stdc++.h>

using namespace std;

int main() {
    // number of the n-th derangement
    int n;
    cout << "Input your n-th derangement : ";
    cin >> n;
    vector<int> v(n);
    iota(v.begin(), v.end(), 0);
    int k = 0;
    cout << "The result of the permutations is:\n";
    do {
        bool isValid = true;
        for (int i = 0; i < n; i++) {
            // if the element has the same position
            if (v[i] == i) {
                // than this is not a valid permutation
                isValid = false;
                break;
            }
        }
        if (isValid) {
            // print the permutation
            for (int i = 0; i < n; i++) {
                cout << v[i] << " ";
            }
            cout << '\n';
            k++;
        }
    } while (next_permutation(v.begin(), v.end()));

    return 0;
}
```

```

// if we have found a valid permutation
if (isValid) {
    // print the permutation
    cout << ++k << " : ";
    for (int i = 0; i < n; i++) {
        cout << v[i] + 1 << " ";
    }
    cout << '\n';
}
} while (next_permutation(v.begin(), v.end()));

return 0;
}

```

And so after we compile and execute the code we can see the result when we input $n = 4$:

```

Input your n-th derangement : 4
The result of the permutations is:
1 : 2 1 4 3
2 : 2 3 4 1
3 : 2 4 1 3
4 : 3 1 4 2
5 : 3 4 1 2
6 : 3 4 2 1
7 : 4 1 2 3
8 : 4 3 1 2
9 : 4 3 2 1

```

Figure 10. The result of every permutation on the 4-th Derangement using the C++ implementation code.

Note that it might not be a good idea to input a large number on this particular code, because this code works on $O(N!)$ Time Complexity which is quite slow, so it might be good to input only $n \leq 11$.

V. CONCLUSION

With The inclusion-exclusion principle, we can solve many problems in mathematics regarding counting. One of that is The famous hat-check problem, a problem that asks for the probability that no person is given the correct hat back by a hat-check person who gives the hat back randomly.

VI. ACKNOWLEDGMENT

In this paper, the author would like to first and foremost thanks to the Almighty God, Allah Azza Wa Jalla, for His Grace and Guidance so that the Author was able to compete this paper. The author will also like to Mrs. Fariska Zakhralativa, M. T as the lecturer of IF2120, Author would also like to thanks all my colleagues or all the support and inspiration that they had given to me. Lastly, Author would like to apologize if there any many mistakes throughout this paper.

REFERENCES

- [1] P. K. Jain; Khalil Ahmad; Om P. Ahuja (1995). Functional Analysis. New

Age International.

- [2] https://www.math.uh.edu/~dlabate/settheory_Ashlock.pdf accessed at 4 December 2019.
- [3] Yiannis N. Moschovakis (1994). Notes on Set Theory. Springer Science & Business Media.
- [4] <https://brilliant.org/wiki/principle-of-inclusion-and-exclusion-pie/> accessed at 5 December 2019..
- [5] Allenby, R.B.J.T.; Slomson, Alan (2010), How to Count: An Introduction to Combinatorics, Discrete Mathematics and Its Applications (2 ed.), CRC Press, pp. 51–60
- [6] Ronald L. Graham, Donald E. Knuth, Oren Patashnik, Concrete Mathematics (1994), Addison–Wesley, Reading MA.
- [7] <http://homepages.math.uic.edu/~kauffman/OldHats.pdf> accessed at 5 December 2019

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 5 Desember 2019



Muhammad Hasan 13518012