Bijection between Two Sets

Miguel A. Lerma

August 5, 2025

Problem. Let n be a positive integer, and let $m = (n+1)!+n$. If T is the set of n-tuples with distinct elements of $\{1, \dots, m\}$, and S is the set of $(n + 1)$-element subsets of $\{1, \dots, m\}$. Prove that there is a bijection $f : T \to S$ such that for each n-tuple $(x_1, \dots, x_n) \in T$, $x_1, \dots, x_n$ are in $f(x_1, \dots, x_n)$.

Solution. We will use a well known result from graph theory. The basic idea is to define a graph whose vertices are the elements of T and S, and join each tuple $t \in T$ with each subset in $s \in S$ such that all elements of t are contained in s. The desired result then follows from a known theorem in graph theory.

We will be using the following definitions:

(1) A *regular* graph is a graph where each vertex has the same number of neighbors. A regular graph with vertices of degree k is called a k-regular graph or regular graph of degree k.

(2) A *bipartite* graph is a graph whose vertices can be divided into two disjoint and independent sets such that every edge connects a vertex in one set to a vertex in the other set. In simpler terms, you can color the graph with two colors, and no two adjacent vertices have the same color.

(3) A T-perfect (of T saturated) matching in a bypartite graph with parts T and S is a matching with disjoint edges that covers every vertex in T. If the two parts have equal size $|T| = |S|$, then a T-perfect matching is also an S-perfect matching, and it is not necessary to specify respect to which part it is saturated.

The basic result is given by the following theorem:

Theorem. In any k-regular bipartite graph with equal partition sizes T and S, a perfect matching exists.

Proof of the Theorem. We use Hall's Marriage Theorem, which provides a necessary and sufficient condition for the existence of a perfect matching in bipartite graphs with equal parts. The Hall's condition is: For every subset A of T, the neighborhood $N(A)$ in S satisfies $|N(A)| \geq |A|$. This condition is in fact satisfied by any k-regular bipartite graph with equal partition sizes because, taking into account $E(A) \subseteq E(N(A))$, we have

$$k \cdot |A| = |E(A)| \leq |E(N(A))| = k \cdot |N(A)| \,,$$

hence $|A| \leq |N(A)|$, and the theorem is proved. □

Proof of the main result. Consider the graph G whose vertices are $T \cup S$, and its edges join each element $(x_1, \ldots, x_n)$ from T to each $(n+1)$-element subset of X containing $x_1, \ldots, x_n$. The graph G is bipartite, and also k-regular with $k = (n+1)!$ because each vertex has exactly $(n+1)!$ neighbors, i.e., for each $(x_1, \ldots, x_n)$, $x_1, \ldots, x_n$ are contained in $m - n = (n+1)!$ subsets of X with $(n+1)$ elements, and for each $(n+1)$-element subset $\{x_1, \ldots, x_{n+1}\}$ of X there are $\binom{n+1}{n} \cdot n! = (n+1)!$ n-tuples whose elements are in $\{x_1, \ldots, x_{n+1}\}$.

So, we have a bipartite k-regular graph G, which by theorem 1 has a perfect matching M. Then, the desired bijection $f : T \to S$ can be obtained by mapping each ordered pair $(x, \ldots, x_n)$ in T to the element of S matched to $(x, \ldots, x_n)$ by M.

This completes the proof of the assertion. □

Remark. The sequence $(n+1)! + n = 1, 3, 8, 27, 124, 725, \ldots$ is A030495 in The On-Line Encyclopedia of Integer Sequences.

Application: "Communicating the Card" Magic Trick. Alice draws n+1 cards from the deck at random, without replacement, and passes n of them, one by one, to her accomplice Bob. If the desk has no more than m = (n+1)! + n cards the order in which Alice passes the cards to Bob contains enough information for him to deduce the remaining card; they just need to agree which perfect matching between n-tuples and $(n+1)!$-element subsets of m cards to use.

Computational Approach

An algorithm. The result shown above is merely existential and does not produce any specific bijection with the required property. Here we provide and actual program (in Python) that does find a bijection as a perfect matching between T and S. The algorithm stops because a perfect matching always exists.

```
from itertools import permutations, combinations
import networkx as nx
from math import factorial

def find_perfect_matching(n):
    '''
    m is (n+1)! + n = number of elements in X = {1, ..., m}.
    T is the set on n-tuples with distinct elements of X.
    S is the set of (n+1)-element subsets of X.
    G is the bipartite graph with nodes T and S.
    perfect_matching is the matching between T and S.
    '''

    m = factorial(n+1) + n
```

```
    # Step 1: Define the sets X, T, and S
    X = set(range(1, m+1))

    # Create n-tuples of distinct elements (T)
    T = list(permutations(X, n))

    # Create (n+1)-element subsets of X (S)
    S = list(combinations(X, n+1))

    # Step 2: Initialize the bipartite graph G
    G = nx.Graph()

    # Add nodes with the bipartite attribute
    G.add_nodes_from(T, bipartite=0)  # Nodes from T
    G.add_nodes_from(S, bipartite=1)  # Nodes from S

    # Step 3: Add edges between nodes in T and S
    for t in T:
        # Find all subsets in S that contain all elements of t
        for s in S:
            if all(elem in s for elem in t):
                G.add_edge(t, s)

    # Step 4: Find a perfect matching
    matching = nx.algorithms.bipartite.matching.\
                hopcroft_karp_matching(G, top_nodes=T)

    # Extract the matching between T and S
    perfect_matching = {k: v for k, v in matching.items() if k in T}

    return perfect_matching
```

The following code prints the matching for the case $n = 2$:

```
# Find the perfect matching
perfect_matching = find_perfect_matching(2)

# Output the matching
for t_node, s_node in perfect_matching.items():
    print(f"{t_node} -> {s_node}")
```

Complexity. Here we look at the computing resources needed to solve the problem as n grows.

Space Complexity. For a given positive integer n we have $|X| = m = (n+1)! + n$ and

$$|T| = \binom{m}{n} \cdot n! = \frac{m!}{(m-n)!}, \qquad |S| = \binom{m}{n+1} = \frac{m!}{(m-n-1)!\,(n+1)!}.$$

For $m = (n+1)! + n$ we have $|T| = |S|$, although there are situations, such as some "communicating-the-card" tricks in which m (the size of the card desk) may be allowed to be less than $(n+1)! + n$. Here we will continue assuming $m = (n+1)! + n$. Under this hypothesis we get

n	m	$\|T\| = \|S\|$
1	3	3
2	8	56
3	27	17,550
4	124	225,150,024
5	725	197,554,684,517,400

This leads to a fast increase in the size of the graph. The asymptotic behavior is (using Stirling's formula):

$$m = (n+1)! + n \sim (n+1)! \sim \sqrt{2\pi(n+1)}\left(\frac{n+1}{e}\right)^{n+1} \quad (n \to \infty),$$

and

$$|T| = |S| \sim ((n+1)!)^n \sim (2\pi)^{n/2}(n+1)^{n(n+1)+n/2}e^{-n(n+1)} \quad (n \to \infty).$$

Hence, the number of vertices V and edges E grow as $|V| = |T| + |S| \sim 2((n+1)!)^n$ and $|E| = |T| \cdot |S| \sim ((n+1)!)^{2n}$ respectively. They provide an estimation of the space needed to contain the graph, which will be approximately proportional to $((n+1)!)^{2n}$.

Time Complexity. After the graph has been built the program uses the Hopcroft–Karp algorithm, with worst-case time complexity $O(|E|\sqrt{|V|}) \sim ((n+1)!)^{5n/2}$ steps. In theory we should add the time needed to build the graph, but creating vertices and edges of a graph can be done much faster than the time taken by each step in the Hopcroft–Karp algorithm, so here we can safely ignore the graph building time as negligible.

Explicit example

The following is an example of bijection from T to S for the particular case $n = 2$, $m = 8$, that satisfies the desired property:

$$\begin{array}{lcl@{\qquad}lcl}
(1,2) & \mapsto & (1,2,4) & (1,3) & \mapsto & (1,3,5) \\
(1,4) & \mapsto & (1,4,7) & (1,5) & \mapsto & (1,4,5) \\
(1,6) & \mapsto & (1,2,6) & (1,7) & \mapsto & (1,5,7) \\
(1,8) & \mapsto & (1,7,8) & (2,1) & \mapsto & (1,2,5) \\
(2,3) & \mapsto & (2,3,6) & (2,4) & \mapsto & (2,4,6) \\
(2,5) & \mapsto & (2,3,5) & (2,6) & \mapsto & (2,5,6) \\
(2,7) & \mapsto & (2,3,7) & (2,8) & \mapsto & (1,2,8) \\
(3,1) & \mapsto & (1,2,3) & (3,2) & \mapsto & (2,3,4) \\
(3,4) & \mapsto & (1,3,4) & (3,5) & \mapsto & (3,5,6) \\
(3,6) & \mapsto & (1,3,6) & (3,7) & \mapsto & (1,3,7) \\
(3,8) & \mapsto & (3,4,8) & (4,1) & \mapsto & (1,4,6) \\
(4,2) & \mapsto & (2,4,5) & (4,3) & \mapsto & (3,4,6) \\
(4,5) & \mapsto & (4,5,8) & (4,6) & \mapsto & (4,6,7) \\
(4,7) & \mapsto & (4,7,8) & (4,8) & \mapsto & (1,4,8) \\
(5,1) & \mapsto & (1,5,8) & (5,2) & \mapsto & (2,5,7) \\
(5,3) & \mapsto & (3,4,5) & (5,4) & \mapsto & (4,5,6) \\
(5,6) & \mapsto & (5,6,8) & (5,7) & \mapsto & (5,6,7) \\
(5,8) & \mapsto & (2,5,8) & (6,1) & \mapsto & (1,6,7) \\
(6,2) & \mapsto & (2,6,7) & (6,3) & \mapsto & (3,6,8) \\
(6,4) & \mapsto & (4,6,8) & (6,5) & \mapsto & (1,5,6) \\
(6,7) & \mapsto & (6,7,8) & (6,8) & \mapsto & (2,6,8) \\
(7,1) & \mapsto & (1,2,7) & (7,2) & \mapsto & (2,4,7) \\
(7,3) & \mapsto & (3,4,7) & (7,4) & \mapsto & (4,5,7) \\
(7,5) & \mapsto & (3,5,7) & (7,6) & \mapsto & (3,6,7) \\
(7,8) & \mapsto & (3,7,8) & (8,1) & \mapsto & (1,3,8) \\
(8,2) & \mapsto & (2,3,8) & (8,3) & \mapsto & (3,5,8) \\
(8,4) & \mapsto & (2,4,8) & (8,5) & \mapsto & (5,7,8) \\
(8,6) & \mapsto & (1,6,8) & (8,7) & \mapsto & (2,7,8)
\end{array}$$