

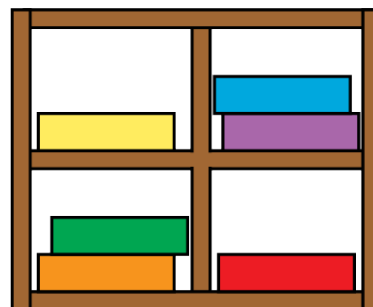
## Grade 6 Math Circles

November 18, 2020

### *Bijections and Binary Strings - Solutions*

#### Introduction

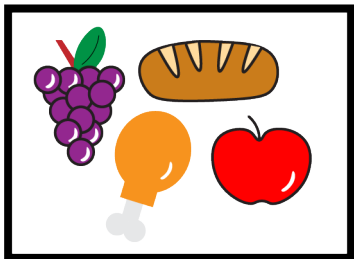
Consider the follow scenario: Say that you have just gone to the store with your family and have bought 6 board games. At home, you have 4 different shelves on which you could place a board game. We could then say that at least one of the shelves would have to house two of the new board games in order for all the new board games to be placed on the shelves. Why? Because we have more new board games than we do shelves. This is an idea known as the Pigeonhole Principle. If we have more objects than containers, at least one container has to have two objects.



This makes the idea of knowing if groups of objects are the same size very important in mathematics. Today, we will talk about bijections and how they can show that two groups have the same size. We will also talk about binary strings, their decompositions and uses in bijections. Before we can dive into bijections however, we need a bit more background.

#### Sets

A **set** is a collection of objects: any collection can be called a set! The objects contained in a set are called **elements**. When writing a set, we wrap the elements in  $\{ \}$  to represent that we are writing a set.



##### Examples:

*Numbers:*  $\{1, 2, 3, 4, 5\}$

*Letters:*  $\{a, b, c\}$

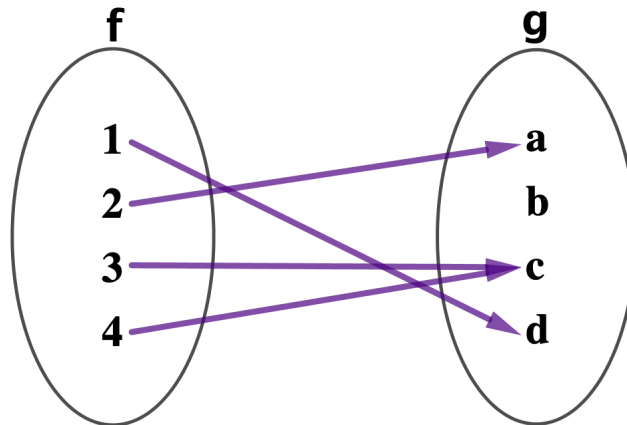
*Grocery List:*  $\{\text{apple, bread, chicken, grapes}\}$

*Sets of Sets:*  $\{\{1, 2\}, \{a, b, c\}, \{\text{apple}\}\}$

Sets can be as long as we want, but they can't have any duplicates and don't have a specific order. So, the set  $\{1, 2, 3\}$  is the same as the set  $\{2, 1, 3\}$ . The number of elements in a set is known as its **cardinality**. We represent cardinality with  $| \cdot |$  around the curly brackets. For example,  $|\{\text{apple, bread, chicken, grapes}\}| = 4$ .

## Relations

A **relation** is a way to relate objects from two sets. It may be helpful to think of a relation as a path from one set to another. For example, consider set  $f = \{a, b, c, d\}$  and set  $g = \{1, 2, 3, 4\}$ . One relation between these two sets is shown here:

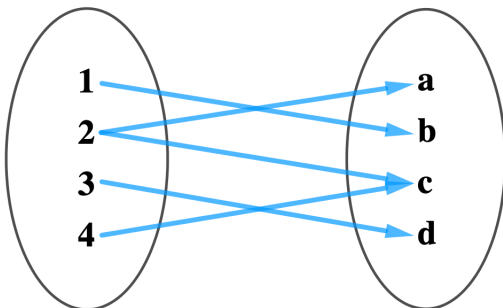


Note we have created connected pairs  $\{1, d\}$ ,  $\{2, a\}$ ,  $\{3, c\}$ ,  $\{4, c\}$ . The initial set is known as the **domain** and the second set is known as the **codomain**.

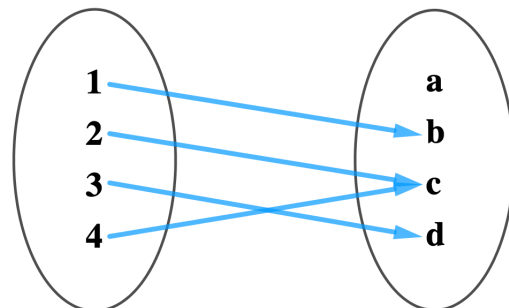
## Functions

A **function** is a relation where each element of the domain is connected to exactly one element in the codomain. For example:

1. Not a function:



2. Function:

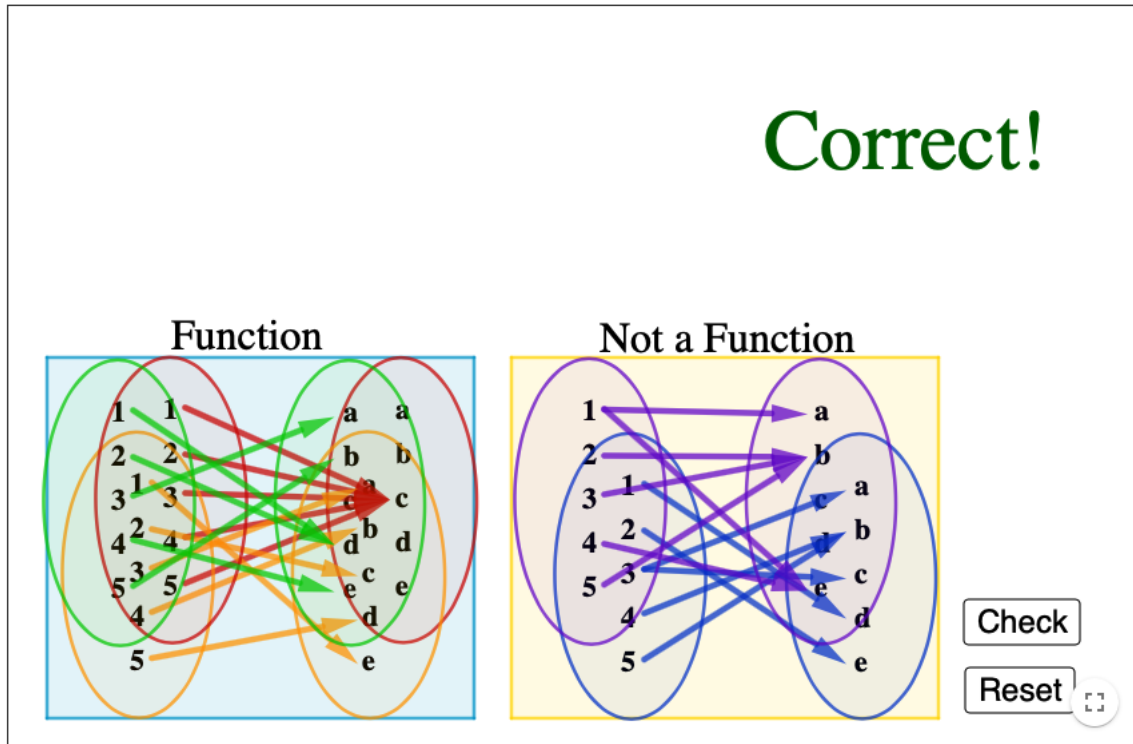


Our first relation is not a function as element 2 from the domain is connected to both elements  $a$  and  $c$  from the codomain. However, our second relation has no such doubles and is thus a function.

**Exercise 1.** Practice distinguishing between functions and other types of relations here: <https://www.geogebra.org/m/fmferybu>.

*Solution:*

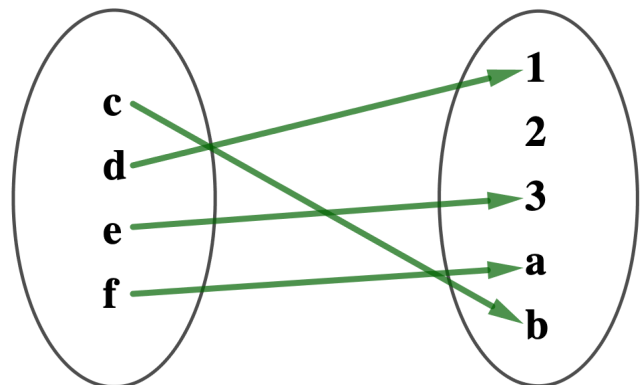
The first, third and fourth (green, orange and red) relations are functions and the second and fifth (blue and purple) relations are not.



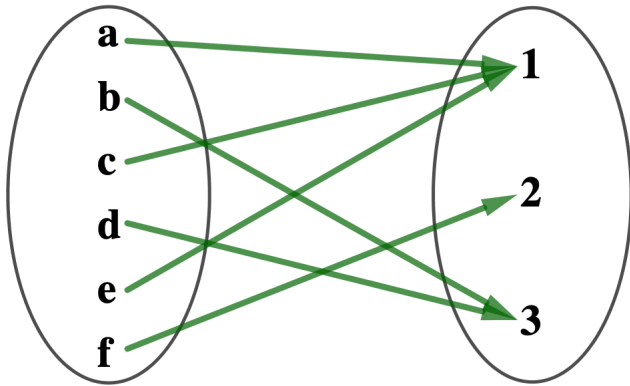
There are a few different categories of functions. Today, we will look at three types of functions: injective, surjective and bijective.

**Definition 1.** *Injective Function*

An **injective** or **one-to-one** function is a relation in which each element from the domain is connected to a unique element from the codomain. In other words, if we pick any element from the codomain, it is either connected to exactly one or no elements from the domain.



**Injective**



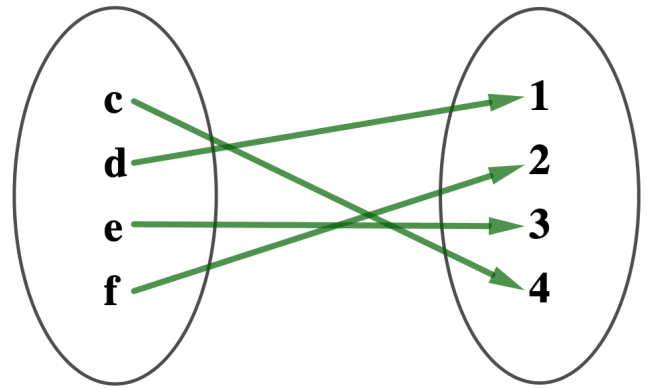
Surjective

**Definition 2.** *Surjective Function*

A **surjective** or **onto** function is one where every element in the codomain connects to an element from the domain. Thus, if we choose any element from the codomain, we can find at least one one element from the domain that it is connected to.

**Definition 3.** *Bijective Function*

A **bijective** function is a relation that is both injective and surjective. What this means is that each element in the domain is mapped to exactly one element in the codomain and vice versa. So, a bijection perfectly matches up all the elements in the two sets.



Bijective

**Exercise 2.** Practice finding injective, surjective and bijective functions here:

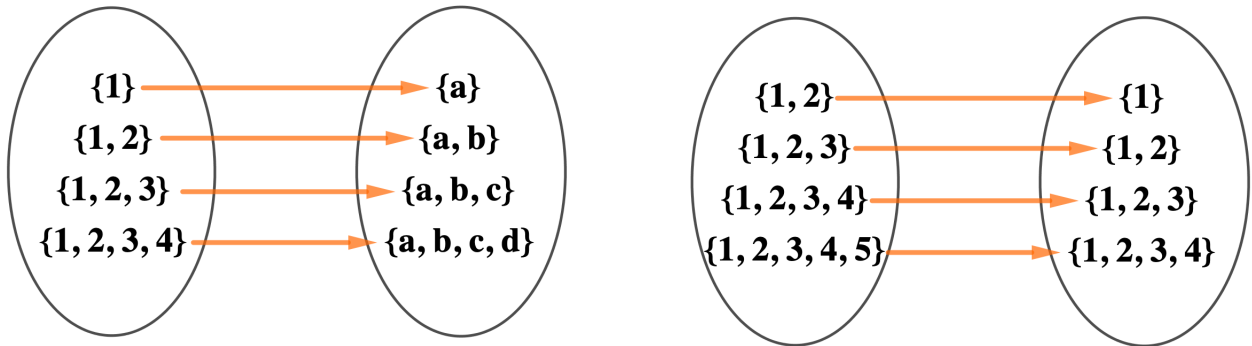
<https://www.geogebra.org/m/dh7yhu7u>.

*Solution:*

The first and fifth (red and blue) functions are injective, the third and seventh (yellow and purple) are surjective, the second and fourth (orange and green) functions are bijective and the sixth (indigo) is neither injective nor surjective.

# Bijections

We can use bijections to show that different sets are equal in cardinality, since we are making pairs of elements where each element in each set must have exactly one connection. The most useful bijective functions are ones that use a pattern to create the matching of pairs. For example:



In the first bijection, we change each number in the given element to a letter corresponding to its place in the alphabet. So, all 1's become a's, 2's become b's, etc. In the second, we remove the largest number from the element.

**Exercise 3.** Try this activity: <https://www.geogebra.org/m/sv85rjrb> to practice finding patterns in bijections.

*Solution:*

		<p><b>Correct!</b></p> <input checked="" type="checkbox"/> Show Patterns <ol style="list-style-type: none"> <li>1. Multiply each number by 2</li> <li>2. Remove 0 and 1 from the set</li> <li>3. Shift each letter by 7</li> <li>4. Create a set from the remaining elements of 1, 2 and 3</li> </ol> <p>Check    Reset</p>

## Binary Strings

A **string** is a sequence of letters or numbers. A **binary string** is a string that is composed of only 1's and 0's. The **length** of a binary string is the total number of 1's and 0's in the string. For example, 101 is a binary string of length 3. We represent the **empty binary string**, or binary string of length 0 by  $\varepsilon$ , the Greek letter epsilon.

We can represent sets of binary strings using **decomposition**. To understand decomposition, we first need a few definitions:

### Definition 4. Concatenation

The binary string produced by concatenating two binary strings  $a$  and  $b$  is  $ab$ . For example, the concatenation of 101 and 11 would be 10111. We can also concatenate sets of binary strings.  $\{1, 11\}\{01, 00, 0\}$  is the set of all binary strings that could be created by concatenating one string from the first set and one from the second. In our example, this is  $\{101, 100, 10, 1101, 1100, 110\}$ .

### Definition 5. $A^*$

The  $*$ -operation represents the set of binary strings created by concatenating binary strings in set  $A$  with each other any number of times (including zero!). For example,  $\{1\}^* = \{\varepsilon, 1, 11, 111, 1111, 11111, \dots\}$ .

### Definition 6. Decomposition

The decomposition of a set of binary strings is a way of using smaller sets, concatenation and the  $*$ -operator to express the larger set. For example, a decomposition of the set of all binary strings would be  $\{0, 1\}^*$ , since at each step in writing a binary string, we choose to add another 0, 1 or stop. Note that decompositions are not unique: another way to decompose the set of all binary strings is  $(\{1\}^*\{0\})^*\{1\}^*$ .

### Watch this video:

<https://youtu.be/cmZnorptwj0> for examples of determining if a given binary string can be created by a certain decomposition.

**Exercise 4.** Try this activity: <https://www.geogebra.org/m/pe8bqsft> to practice determining if a binary string is in the set represented by a given decomposition.

*Solution:*

$\{1\}\{0,001\}^*\{\epsilon,11\}$

Created by Decomposition	Not Created by Decomposition
<div style="display: flex; justify-content: space-between;"> <span>1</span> <span>10</span> <span>111</span> </div> <div style="text-align: center; margin: 10px 0;">10000000111</div> <div style="display: flex; justify-content: space-between;"> <span>10010001</span> <span>1011</span> </div>	<div style="text-align: left; margin-bottom: 10px;">101</div> <div style="display: flex; justify-content: space-around; margin-bottom: 10px;"> <span>11001</span> <span>0011</span> </div> <div style="text-align: right;">10011011</div>
<b>Correct!</b>	<input type="button" value="Check"/> <input type="button" value="Reset"/>

When creating binary string decompositions, we prefer to only have one way to create each binary string in the resultant set\*. If there is more than one way to create a particular binary string from a decomposition, we call that decomposition **ambiguous**. For example,  $\{00,0\}^*$  is ambiguous because I could create the binary string 00 from using 00 once or from sticking two 0's together.

**Watch this video:**

<https://youtu.be/HiiroCQK978> for an example of showing a decomposition is ambiguous.

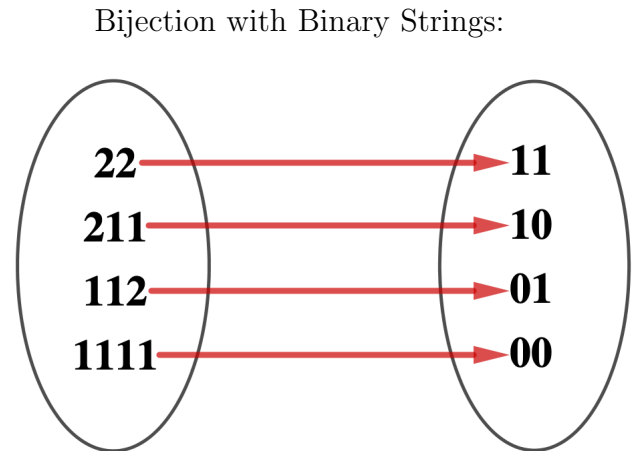
*\*Note that the reason we prefer the decomposition to be unambiguous is to help with determining the number of strings in the set created by the decomposition. Doing this requires mathematics that you will learn in university and we won't go into it here.*

**Exercise 5.** Show that  $\{1,11\}\{10,0,01\}^*$  is ambiguous. Check your answer by posting your response on Piazza!

*Solution:*

One binary string that could be used to show that this is ambiguous is 110. We could either create it with 1 then 10 or 11 then 0. Since there are two ways to create this binary string with the decomposition, it is ambiguous.

When using binary strings in bijections, we often talk of all binary strings of length  $n$ . This is the set of all possible binary strings that have a certain length,  $n$ . For example, the set of all binary strings of length 3 is  $\{000, 001, 010, 011, 100, 101, 110, 111\}$ . You can check to make sure that you have all the binary strings of a certain length by multiplying 2 by itself  $n$  times. So, for the strings of length 3, there should be  $2 * 2 * 2 = 8$  possible strings. As shown above, this is indeed the case. The example on the right is a bijection that uses the binary strings of length 2 as the codomain.



In this bijection, we change each 2 from the first set to a 1 and each pair of 1's to a 0 to get the elements of the second set.



# Problem Set

1. Give an example of each of the following type of relations:

(a) Function

(d) Bijective Function

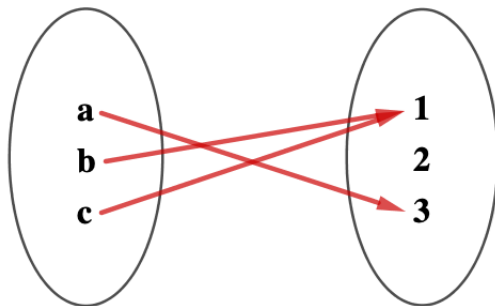
(b) Injective, but not surjective function

(e) Relation but not a function

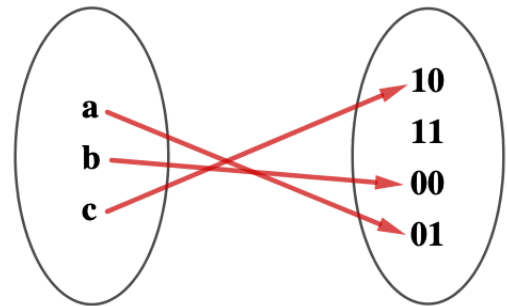
(c) Surjective, but not injective function

*Solution:*

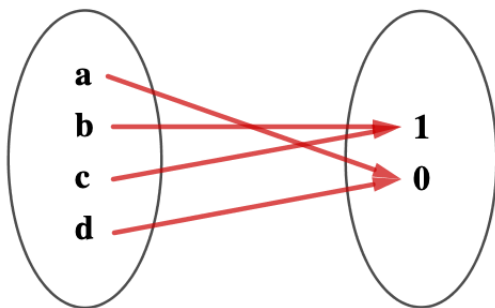
a)



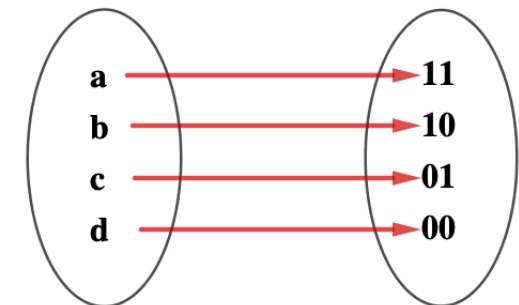
b)



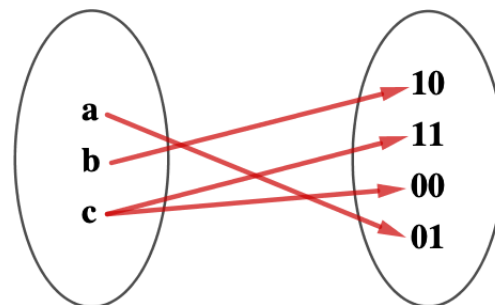
c)



d)



e)



2. Give an example of a set with a cardinality of 10.

*Solution:*

A set with cardinality 10 is a set with 10 elements. One such set is:

{apples, bananas, oranges, kiwi, strawberries, mango, blackberries, raspberries, blueberries, grapes}

3. Let A represent the set of binary strings of length 3 and B represent the set of odd numbers greater than 4 but less than 18. Explain why it's not possible to find a bijection between these two sets.

*Solution:*

Note that  $A = \{000, 001, 010, 011, 100, 101, 110, 111\}$  and  $B = \{5, 7, 9, 11, 13, 15, 17\}$ .

So, the cardinality of A is 8 and the cardinality of B is 7. To make a bijection, we must pair each element from set A with a different element in set B (the function must be injective). However, A has more elements than B so this isn't possible. So, we cannot make a bijection between A and B.

4. Find a bijective function that follows a pattern between the following pairs of sets. Explain the pattern and draw a chart showing the connections between the domain and codomain.

(a)  $\{000, 00110, 110, 1, 001\}$   
and  $\{111, 11001, 001, 0, 110\}$

(d)  $\{\{1\}, \{2, 3\}, \{1, 3\}, \{1, 2, 3\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}$   
and  $\{1000, 0110, 1010, 1110, 0111, 1111\}$

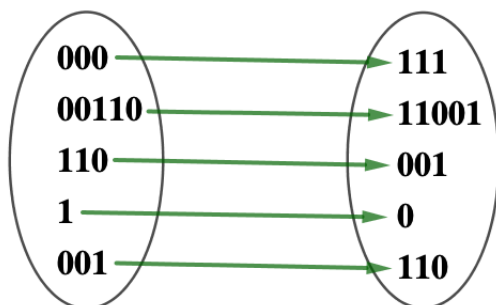
(b)  $\{000, 001, 010, 011, 100, 101, 110, 111\}$   
and  $\{aaa, aab, aba, abb, baa, bab, bba, bbb\}$

*Hint: Match the elements according to their listed order in the set. Try to find a pattern to the functions found.*

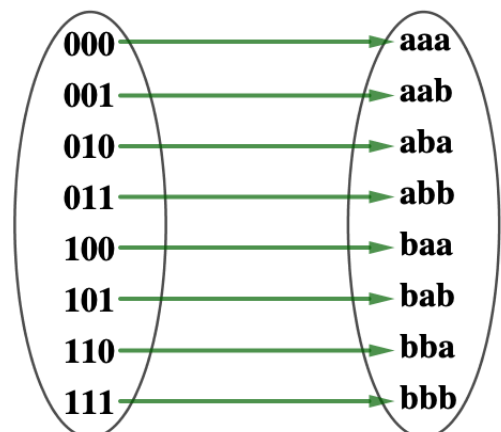
(c)  $\{43, 26, 32, 10, 67, 88, 91\}$   
and  $\{78, 95, 89, 01, 54, 33, 20\}$

*Solution:*

a) Change all 0s to 1s and 1s to 0s.

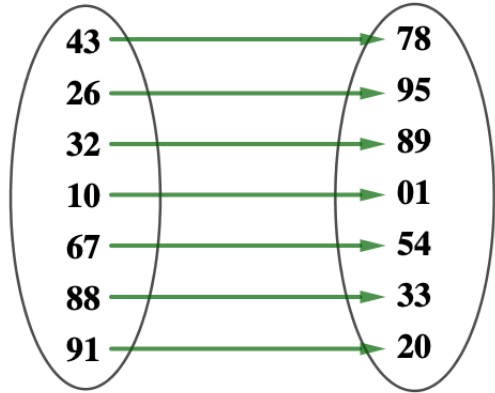


b) Change all 0s to as and 1s to bs.



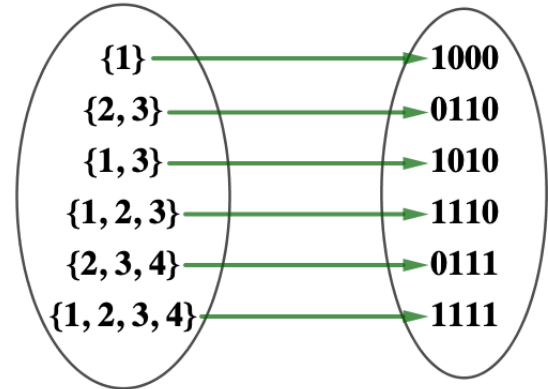
c) Change each number according the the following table:

<b>Start</b>	1	2	3	4	5	6	7	8	9	0
<b>End</b>	0	9	8	7	6	5	4	3	2	1



d) Consider a binary string of length 4. If the initial element has 1, then the first number in the binary string is 1, otherwise it is 0.

Follow a similar procedure for 2 and the second number, 3 and the third number and 4 and the last number in the binary string.



5. Determine if the following binary strings are in the set of binary strings created by this decomposition:  $\{0, 00\}\{1, 10, 100\}^*\{001\}$

(a) 10001

(d) 0110101

(b) 0010111001

(e) 0111010010001

(c) 0001

(f) 0110001001

*Solution:*

(a) Cannot be created by the decomposition. Any binary string created by this decomposition must start with a 0 and this binary string starts with a 1.

(b) Can be created by the decomposition. A break down of the creation would be 00 10 1 1 1 001.

(c) Can be created by the decomposition. A break down of the creation would be 0 001.

(d) Cannot be created by the decomposition. Any binary string created by this decomposition must end with 001 and this binary string ends with a 101.

(e) Can be created by the decomposition. A break down of the creation would be 0 1 1 10 100 10 001.

(f) Cannot be created by the decomposition. There is a block of three 0s in the middle of the proposed binary string. Since we still have another 1 before starting the ending 001, we know that it has to come from the \* portion. However, there is

no way to make three 0s in a row by pasting 1, 10 and 100s together. So, this binary string cannot be created by the given decomposition.

6. The following decompositions are ambiguous. Find a binary string that can be created in two different ways with the decompositions to show this fact.

(a)  $\{11, 111\}^*$

(c)  $\{1\}^*\{0, 10\}^*$

(b)  $\{1\}\{0, 00\}^*\{1, 11\}$

(d)  $\{0\}^*\{1\}^*\{0\}^*\{1\}^*$

*Solution:*

(a) Consider the binary string 111111. It can be created using 11 11 11 or 111 111. Thus, it can be created in two different ways by the decomposition and the decomposition is ambiguous.

(b) Consider 1001. It can be created using 1 0 0 1 or 1 00 1.

(c) Consider 1010. It can be created with 1 0 10 or 10 10.

(d) Consider 01. It can be created using  $0 1 \varepsilon \varepsilon$  or  $0 \varepsilon \varepsilon 1$  or  $\varepsilon 1 0 \varepsilon$  or  $\varepsilon \varepsilon 0 1$ .