Lecture 12 - Functions

6.042 - March 20, 2003

1 Functions

A function $f : A \mapsto B$ maps each element of the set A to exactly one element of the set B. More formally:

Definition 1 *A function $f : A \mapsto B$ is a relation $f \subseteq A \times B$ such that for every $x \in A$, there is a unique $y \in B$ such that $(x, y) \in f$. This unique value y is denoted $f(x)$. The set A is called the* domain *of f, and B is called the* range *of f.*

The *image* of a function $f : A \mapsto B$ is the set of all values that the function takes on; that is, the image of f is the set:

$$\{y \in B \mid \exists x \ f(x) = y\}$$

The image is always a subset of the range, but may or may not be equal to the range. For example, let $f : \mathbb{R} \mapsto \mathbb{R}$ be the function defined by $f(x) = x^2$. The range of this function is all the real numbers, but the image is only the nonnegative real numbers. On the other hand, the image of the function $g : \mathbb{R} \mapsto \mathbb{R}$ defined by $g(x) = x^3$ is equal the whole range, $\mathbb{R}$.

(Note: Rosen uses the term "codomain" for what we call the range, and "range" for what we call the image. Unfortunately, both his naming scheme and ours are in wide use.)

1.1 Types of Functions

There are some important properties that a function may or may not possess. A function $f : A \mapsto B$ is:

- *surjective* if its image is equal to its range; that is, if:

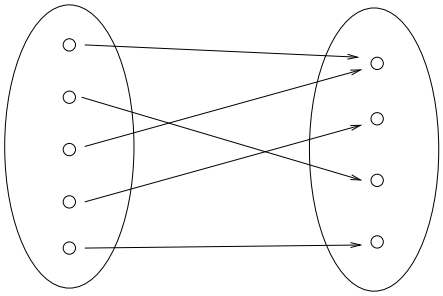
$$\forall y \in B \ \exists x \in A \ f(x) = y$$

- *injective* if distinct elements of the domain are always mapped to distinct elements of the range; that is, if:

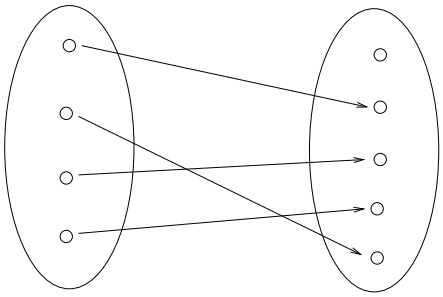
$$\forall x, x' \in A \ (x \neq x') \rightarrow (f(x) \neq f(x'))$$

- *bijective* if the function is both surjective and injective

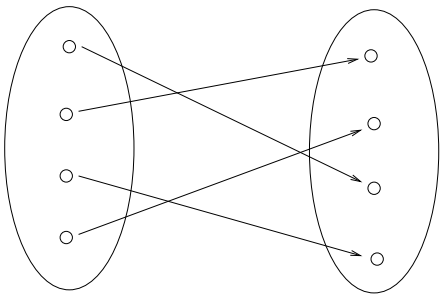
To get a feel for these properties, let's look at some funtions where the domain and range are finite. In each diagram below, the domain is on the left and the range is on the right. Small circles indicate set elements, and arrows show how each element of the domain is mapped to an element of the range.



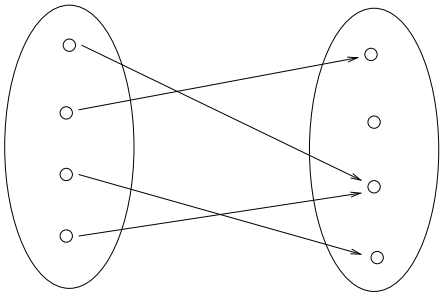
This first function is surjective, because each element of the range is mapped to by some element of the domain. However, it is not injective, because the topmost element in the range is mapped to by two elements of the domain.



This function is not surjective, because the topmost element of the range is not mapped to. On the other hand, it is injective because no two elements of the domain map to the same element in the range.



This function is surjective, because every element of the range is mapped to. It is also injective, because no two elements of the domain map to the same element in the range. Because this function is both surjective and injective, it is also bijective.



This last function is neither surjective nor injective.

There are several other names in common use for the properties described here. A surjective function is sometimes called *onto*. An injective function may be called *into* or *one-to-one*. Bijective functions are sometimes called *one-to-one correspondences*. One type of bijection is particularly important:

Definition 2 *A* permutation *is a bijective function mapping a finite set to itself.*

1.2 Examples of Functions

Let's determine the properties of some common functions.

1. Let $f : \mathbb{R} \mapsto \mathbb{R}$ be the function defined by $f(x) = |x|$.

   This function is not injective, because $f(1) = f(-1)$, for example. It is not surjective, because $f(x)$ is never equal to, say, -1. (If we restricted the range to the nonnegative real numbers, the function would be surjective.)

2. Let $f : \mathbb{R} \mapsto \mathbb{R}$ be the function defined by $f(x) = \sqrt{x}$.

   This is not even a function! When x is nonnegative, $\sqrt{x}$ denotes the positive square root of x. When x is negative, $\sqrt{x}$ is certainly not a real number and, arguably, not defined at all.

3. Let $f : \mathbb{N} \mapsto \mathbb{N}$ be the function defined by $f(x) = 2x$.

   This function is injective; doubling different numbers gives different numbers. However, it is not surjective; doubling a natural number never gives 7, for example. (If the domain and range were enlarged to $\mathbb{R}$, this function would be both surjective and injective, and therefore also bijective.)

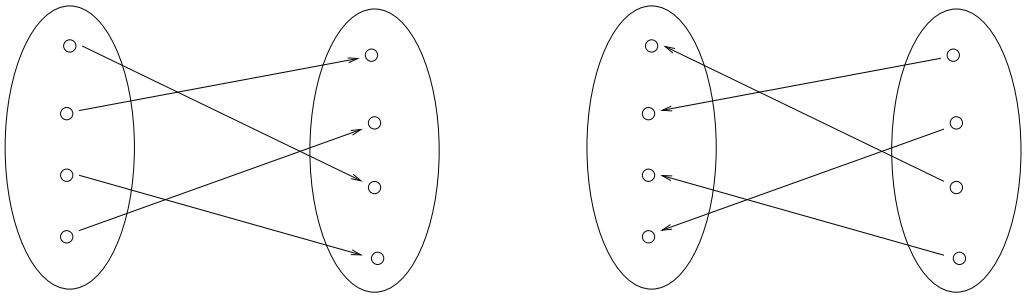
4. Let $f : \mathbb{N} \mapsto \mathbb{N}$ be the function defined by $f(x) = \lfloor x/2 \rfloor$.

   This function is not injective, because $\lfloor 5/2 \rfloor = \lfloor 4/2 \rfloor$, for example. However, the function is surjective; for every $y \in \mathbb{N}$, there is an x such that $f(x) = y$, namely, $x = 2y$.

5. Let $f : \mathbb{R} \mapsto \mathbb{R}$ be the function defined by $f(x) = x^3$. This function is surjective (since every real number is the cube of some other real number), injective (since different real numbers have different cubes), and bijective (since it is both surjective and injective).

1.3 Inverses

For every bijective function f, there is a related function called the *inverse of* f. Intuitively, the inverse of f maps each element of the range back to the corresponding element of the domain. In the example below, the inverse of the function on the left is the function on the right. Note that the domain of the inverse is the range of f, and vice versa.



Formally, the *inverse of a function* $f : A \mapsto B$ is a function $f^{-1} : B \mapsto A$ that maps each $y \in B$ to the unique $x \in A$ such that $f(x) = y$. A function has a well-defined inverse if and only if it is bijective. An an example, the inverse of the function $f : \mathbb{R} \mapsto \mathbb{R}$ defined by $f(x) = 10x$ is the function $f^{-1} : \mathbb{R} \mapsto \mathbb{R}$ defined by $f^{-1}(x) = x/10$.

1.4 Cube Roots

You can use bijections and inverses to compute cube roots in your head! First, let's look at the cubes of the first ten natural numbers:

0	$\underline{0}$
1	$\underline{1}$
2	$\underline{8}$
3	$2\underline{7}$
4	$6\underline{4}$
5	$12\underline{5}$
6	$21\underline{6}$
7	$34\underline{3}$
8	$51\underline{2}$
9	$72\underline{9}$

Consider the function f mapping each number in the range 0 to 9 to the last digit of its cube. For example, $f(2) = 8$, $f(6) = 6$, and $f(7) = 3$. Notice that f is a bijection. Now suppose that we cube a two-digit number $n = 10a + b$. Then we have:

$$\begin{aligned} n^3 &= (10a + b)^3 \\ &= 1000a^3 + 300a^2b + 30ab^2 + b^3 \end{aligned}$$

This shows that the last digit of n^3 is the last digit of b^3, which is $f(b)$. Since f is a bijection, it has an inverse. Therefore, *the last digit of n^3 determines the last digit of n.*

You can use this fact to compute cube roots of perfect cubes up to a million in your head. The bad news it that you must memorize the table above. Suppose that you are given a perfect cube m.

- Take the last digit of m and apply the function f^{-1}. This is the last digit of the cube root.
- Drop the last three digits of m. Determine the first digit of the cube root by interpolating in the table.

For example, suppose $m = 474552$. The last digit of the cube root is $f^{-1}(2) = 8$. Dropping the last three digits of m leaves 474. This lies between 343 and 512, so the first digit of the cube root is 7. Therefore, $\sqrt[3]{474552} = 78$. You'll have more hot dates than you can handle, with this nifty trick to show off!

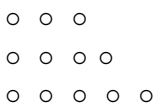
1.5 Cardinality

Every finite set A has a size or *cardinality*, which is denoted $|A|$. For example, $|\{a, b, c\}| = 3$. The number of elements in a set seems like a fundamental property, not something requiring a definition. However, without a careful definition of cardinality, one can never *prove* anything about the sizes of sets. And we'll be doing exactly that for much of the remainder of this course. So here is the definition of cardinality:

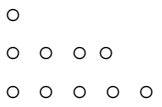
Definition 3 *The cardinality of a finite set A is defined to be n if there is a bijection from A to the set $\{1, 2, 3, \ldots, n\}$.*

2 The Game of Nim

Nim is a game involving two players, some pennies, and mathematical induction. The game begins with a bunch of pennies, arranged in one or more rows. For example, here we have three rows of pennies:



The two players take turns. On each turn, a player must remove one or more pennies, all from a single row. The player who takes the last penny wins. For example, suppose that the first player removes two pennies from the first row:



Now the second player removes all the pennies from the last row, leaving this configuartion:

The first player then takes three pennies from the last row:

The second player is now in trouble; she must take exactly one of these two pennies. Either way, the first player takes the last penny and wins the game.

There is a compact way to describe a configuration in Nim: list the number of pennies in each row. For example, the starting configuartion in the game above is $(3, 4, 5)$. The game then passed through the configurations $(1, 4, 5)$, $(1, 4, 0)$, and $(1, 1, 0)$.

2.1 Nimsums

A Harvard professor, Charles Bouton, discovered the optimal strategy for Nim in 1901. Bouton's strategy relies on a special mathematical operation called a *Nimsum*. The Nimsum of natural numbers $c_1, \ldots, c_k$ is itself a natural number that is computed as follows:

- Regard $c_1, \ldots, c_k$ as binary numbers.
- The i-th bit of the Nimsum is the xor of the i-th bits of $c_1, \ldots, c_k$.

(Xor is pronounced "eks-or" and is short for "exclusive-or". The xor of bits b_1 and b_2 is denoted $b_1 \oplus b_2$ and defined as follows:

b_1	b_2	$b_1 \oplus b_2$
0	0	0
0	1	1
1	0	1
1	1	0

As a consequence of this definition, the xor of bits $b_1, \ldots, b_k$ is 0 if the sum of the b_i is even and 1 if the sum is odd. For example, $1 \oplus 0 \oplus 1 \oplus 1 = 1$, but $1 \oplus 1 \oplus 0 \oplus 0 = 0$.)

As an example, the Nimsum of 3, 4, and 5 is computed as follows:

$$\begin{array}{rcl} 3 & = & 011 \\ 4 & = & 100 \\ 5 & = & 101 \\ \hline & & 010 \quad = \quad 2 \end{array}$$

Here, we xor each column of bits to obtain the Nimsum 010, which is the binary representation of 2.

2.2 The Winning Strategy

Suppose that $(c_1, \ldots, c_k)$ is a configuration in Nim; that is, there are a total of k rows, and the i-th row contains c_i pennies. This configuration is called:

- *safe* if the Nimsum of $c_1, \ldots, c_k$ is nonzero
- *unsafe* if the Nimsum of $c_1, \ldots, c_k$ is zero

In this way, all possible configurations in Nim are now partitioned into two parts: the safe configurations and the unsafe configurations. Bouton's strategy is as follows:

- If you see a safe configuration on your turn, then select a move that leaves your opponent with an unsafe configuration.
- If you see an unsafe configuration on your turn, you're doomed. Every possible move leaves your opponent with a safe configuration; select among them arbitrarily.

Thus, if the first player sees a safe position, she selects a move that leaves the second player an unsafe position. Every move available to the second player then leaves the first player with another safe position. This cycle repeats until the first player wins the entire game.

2.3 An Example

Let's use Bouton's strategy in one example game before trying to prove a general theorem. The bizarre French movie "Last Year at Marienbad" features a Nim game beginning in the configuration $(1, 3, 5, 7)$. Let's analyze this configuration by computing the Nimsum:

$$\begin{array}{rcl} 1 & = & 001 \\ 3 & = & 011 \\ 5 & = & 101 \\ 7 & = & 111 \\ \hline & & 000 \quad = \quad 0 \end{array}$$

Since the Nimsum is zero, this is a hopeless situation for the first player. In particular, no matter what she does, the second player is left with a safe configuration; that is, one in which the Nimsum is nonzero. For example, suppose that the first player removes the entire fourth row. Then the Nimsum becomes:

$$\begin{array}{rcl} 1 & = & 001 \\ 3 & = & 011 \\ 5 & = & 101 \\ 0 & = & 000 \\ \hline & & 111 \quad = \quad 7 \end{array}$$

This is a safe configuration. To maintain her advantage, the second player needs to leave the first player in an unsafe position; that is, a position with Nimsum zero. Only one move accomplishes this: removing three pennies from the third row:

$$\begin{array}{rcl} 1 & = & 001 \\ 3 & = & 011 \\ 2 & = & 010 \\ 0 & = & 000 \\ \hline & & 000 \quad = \quad 0 \end{array}$$

Now the first player is again confronted with an unsafe position, one with a Nimsum of zero. No matter what she does, the second player will be left with a safe position. For example, suppose that the first player takes all three pennies from the second row. Then the second player is left with:

$$\begin{array}{rcl} 1 & = & 001 \\ 0 & = & 000 \\ 2 & = & 010 \\ 0 & = & 000 \\ \hline & & 011 \quad = \quad 3 \end{array}$$

Sure enough, this is a safe position. One again, the second player needs to leave the first player in an unsafe position. Removing one penny from the third row accomplishes this:

$$\begin{array}{rcl} 1 & = & 001 \\ 0 & = & 000 \\ 1 & = & 001 \\ 0 & = & 000 \\ \hline & & 000 \quad = \quad 0 \end{array}$$

The first player is clearly going to lose. She must take one of the two remaining pennies, leaving the second player to take the other and win the game.

2.4 Proof of Correctness

Now let's prove that Bouton's strategy works. We'll need two lemmas.

Lemma 1 *If a player sees an unsafe configuration on her turn, then every possible move leaves her opponent with a safe configuration.*

Proof. Suppose that the player sees an unsafe configuration; that is, a configuration $(c_1, \ldots, c_k)$ with Nimsum zero. Then the player must remove pennies from some row j, leaving $c'_j < c_j$ pennies behind. In binary, the numbers c_j and c'_j must differ in some bit position. But then the Nimsum of the resulting configuration has a 1 in that bit position. Therefore, the Nimsum is nonzero, meaning that the resulting configuration is safe. □

Lemma 2 *If a player sees a safe configuration on her turn, then she has a move that leaves her opponent with an unsafe configuration.*

Proof. Suppose that the player sees a configuration $(c_1, \ldots, c_k)$ with Nimsum $s \neq 0$.

Let i be the position of the most-significant 1 in the binary representation of s. Since the i-th position of s is nonzero, there must be some row size c_j that is nonzero in the i-th position as well. Suppose that the player removes all but c'_j pennies from the j-th row, where c'_j differs from c_j in exactly those positions where s has a 1.

We must check that c'_j, the number of pennies the player leaves behind, is less than c_j, the number of pennies originally in the j-th row. This is true because c_j has a 1 in the i-th position, c'_j has a 0 in this position, and the two numbers agree in all higher positions.

Modifying the game configuration by changing c_j by c'_j reduces the Nimsum to zero, because c_j and c'_j disagree in exactly those positions where the original Nimsum had a 1. □

An example may clarify the preceding argument. Suppose that a player sees the configuration $(1, 2, 4, 4)$. We can compute the Nimsum as follows:

$$\begin{array}{rcl} 1 & = & 001 \\ 2 & = & 010 \\ 4 & = & 100 \\ 4 & = & 100 \\ \hline & & 011 \quad = \quad 3 \end{array}$$

The Nimsum is 011 in binary. The most significant 1 is in the second position from the right. The number of pennies in the second row, 010 in binary, also has a 1 in this position. We should leave 001 pennies in this row, since this differs from the number of pennies there now (010) in every position where the Nimsum (011) has a 1. Now we compute the Nimsum of the resulting configuration:

$$\begin{array}{rcl} 1 & = & 001 \\ 1 & = & 001 \\ 4 & = & 100 \\ 4 & = & 100 \\ \hline & & 000 \quad = \quad 0 \end{array}$$

As expected, this is an unsafe position.

We're ready to prove the main result.

Theorem 3 *If the current player has a safe position, then she can guarantee a win. Otherwise, the other player can guarantee a win.*

The proof uses induction, with the statement of the theorem as the induction hypothesis. This hypothesis is quite complicated! Suppose that we define three smaller propositions:

$$\begin{array}{rcl} A & = & \text{current player has a safe position} \\ B & = & \text{current player can guarantee a win} \\ C & = & \text{next player can guarantee a win} \end{array}$$

In these terms, the induction hypothesis is $(A \to B) \wedge (\overline{A} \to C)$. Induction hypotheses containing implications are always a bit tricky, but this one has *two* implications! Some square-bracketed comments are included in the proof to clairfy the structure.

Proof. The proof is by strong induction on n, the total number of pennies remaining. Let $P(n)$ be the proposition that in all Nim configurations with n pennies, if the current player has a safe position, then she can guarantee a win and, otherwise, the other player can guarantee a win.

First, suppose that $n = 0$; that is, no pennies remain. This is an unsafe position. Therefore, the first implication is trivially true because the if-part is false. The second

implication is also true, because when the current player is left with no pennies, the other player has just won the game. [In other words, the proposition $(A \to B) \wedge (\overline{A} \to C)$ is true because A is false and C is true.]

Next, we must show that for all $n \geq 0$, the propositions $P(0), \ldots, P(n)$ imply $P(n+1)$. Assume $P(0), \ldots, P(n)$ and consider a configuration with $n+1$ pennies remaining. Now we must show two things:

1. *If the configuration is safe, the current player can guarantee a win.* Suppose that the configuration is safe. By Lemma 2, the current player has a move that leaves the next player with an unsafe configuration involving fewer than $n+1$ pennies. If $m < n+1$ pennies remain, then assumption $P(m)$ implies that the player *after* the next player (which is the current player) can guarantee a win. [This shows $A \to B$.]

2. *If the configuration is unsafe, the next player can guarantee a win.* Suppose that the configuration is unsafe. By Lemma 1, every move leaves the next player with a safe configuration involving $m < n+1$ pennies. The assumption $P(m)$ then implies that the next player can guarantee a win. [This shows $\overline{A} \to C$.]

Putting these arguments together gives $P(n+1)$. [That is, $A \to B$ and $\overline{A} \to C$ together imply $(A \to B) \wedge (\overline{A} \to C)$.] Therefore, the theorem follows by the principle of induction. $\square$