3.3: Recursive Definitions

In this section we will apply the ideas of recursively defined functions to more general objects, before generalizing our notion of induction in the next section.

Question 1. Consider the pattern of numbers below. How is each row related to the row above it? What should the next row be?

$$
\begin{array}{c}
1 \\
1\ 1 \\
1\ 0\ 1 \\
1\ 1\ 1\ 1 \\
1\ 0\ 0\ 0\ 1 \\
1\ 1\ 0\ 0\ 1\ 1 \\
1\ 0\ 1\ 0\ 1\ 0\ 1 \\
1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\
1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1 \\
1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1 \\
1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1 \\
1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1 \\
1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1
\end{array}
$$

We will use the term *object* loosely; an object could be a number, a mathematical structure, a function, or almost anything else we want to describe. A recursive definition of a given object has the following parts:

Base Case. Here we usually define the simplest possible object.
Recursive Case. Here we define a more complicated object in terms of the simpler, already defined objects in the sequence.

Example 1. Any recurrence relation is a recursive definition of a function. For example, the recurrence relation

$$H(n) = \begin{cases} 1 & \text{if } n = 1 \\ H(n-1) + 6n - 6 & \text{if } n > 1 \end{cases}$$

is a recursive definition with

Base Case. $H(1) = 1$.
Recursive Case. For any $n > 1$, $H(n) = H(n-1) + 6n - 6$.

Example 2. The Bacon Number is calculated using a recursive definition on X, the set of actors and actresses with finite Bacon Number. The set X is defined as follows.

Base Case. Kevin Bacon $\in X$.
Recursive Case. Let x be an actor or actress. If, for some $y \in X$, there has been a movie in which both x and y appear, then $x \in X$.

A similar definition can be used to construct a set containing all people (or computers) infected with a virus or the collection of people that have heard about a tornado warning, etc.

Example 3. Given a list of symbols $a_1, a_2, \ldots, a_m$, a string of these symbols is:

Base Case 1. The empty string, denoted by λ, or
Base Case 2. any of the original symbols a_i with $i \in \{1, 2, \ldots, m\}$.
Recursive Case. xy, the *concatenation* of x and y, where x and y are strings.

Example 4. A special kind of string, from the list of symbols above, called a *palindrome* can be defined as follows.

Base Case 1. The empty string λ is a palindrome.
Base Case 2. Any of the original symbols a is a palindrome.
Recursive Case. If x and y are palindromes, then yxy is a palindrome.

Example 5. The set X of all *binary strings* (strings with only 0's and 1's) having the same number of 0's and 1's is defined as follows.

Base Case. The empty string λ is in X; i.e. $\lambda \in X$.
Recursive Case 1. If $x \in X$, so are $1x0$ and $0x1$.
Recursive Case 2. If x and y are in X, then so is xy.

Example 6. If s is a string, define its *reverse* s^R as follows.

Base Case. $\lambda^R = \lambda$.
Recursive Case. If s has one or more symbols, write $s = ra$ where a is a symbol and r is a (possibly empty) string. Then $s^R = (ra)^R = ar^R$.

Theorem 1. *If a is a symbol, then $a^R = a$.*

Proof.

<u>**Writing Recursive Definitions.**</u> The key to writing a recursive definition is to see the desired object as being built out of levels. The recursive case of the definition must describe a level in terms of the next simplest level. The base case should describe the simplest possible object.

Example 7. Suppose you start browsing the internet at some specified page p. Let X be the set of all pages you can reach by following links, starting at p. Give a recursive definition for the set X.

Recursive Case.

Base Case.

Example 8. Give a recursive definition for the set of all odd natural numbers.

Base Case.
Recursive Case.

Example 9. (Recursive Jokes)

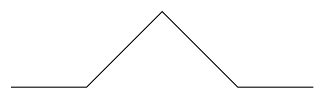
1. It isn't unusual for the following to be in the index of a book: Recursion, see *Recursion*
2. Consider the acronym for VISA (VISA International Service Association)
3. Pete and Repeat are in a boat. Pete fell out. Who was left?

<u>**Recursive Geometry.**</u> We can use recursive definitions for geometric patterns to describe *fractals*, a special type of shape with infinite layers of self-similarity.

Example 9. Define a sequence of shapes as follows.

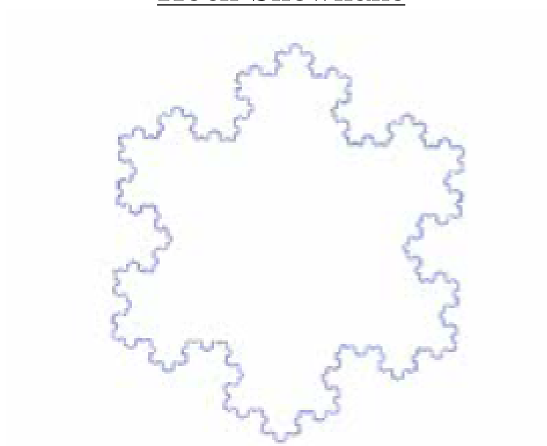
> **Base Case.** $K(1)$ is an equilateral triangle.
> **Recursive Case.** For $n > 1$, $K(n)$ is formed by replacing each line segment

of $K(n-1)$ with the shape



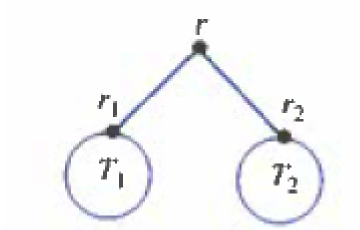
such that the central vertex points outward.

<u>**Koch Snowflake**</u>



Example 10. (Binary Trees) The set of all binary trees can be defined recursively.

Base Case 1. The empty tree is a binary tree.
Base Case 2. A single vertex is a binary tree. In this case, the vertex is the root of the tree.
Recursive Case. If T_1 and T_2 are binary trees with roots r_1 and r_2, respectively, and r is a single vertex, then the tree



is a binary tree with root r.

Homework. (Due Monday, November 19) Section 3.3: 2, 8, 12, 18,
Practice Problems. Section 3.3: 3, 4, 10, 11, 13, 17, **22, 23, 26-29**