Recursion

**Definition 1 (Recursive).** Recursive: adj. See: Recursive – STAN KELLY-BOOTLE, “Devil’s DP Dictionary” English computer scientist and author (1929 –)

Recursion defines objects in terms of
1. A base case or base cases.
2. Rules that reduce all other cases to the base case(s).

**Recursive Computations**

To compute a value $v_n$ recursively, write $v_n$ in terms of lower order terms, for instance, if

$$v_n = 2v_{n-1} + 3v_{n-2}, \quad n \geq 2$$

Then

$$v_4 = 2v_3 + 3v_2 = 2(2v_2 + 3v_1) + 3(2v_1 + 3v_0)$$

$$v_4 = 4v_2 + 12v_1 + 9v_0 = 20v_1 + 21v_0$$

This is a “top-down” evaluation process.

Recursive computations contrast with inductive computations, where values are computed from the “bottom-up.”

**Recursion**

Here are some examples of recursively defined objects.

- **Addition:** $0 + a = a$ (base case) and $(n + 1) + a = 1 + (n + a)$ for $n \geq 0$.

- **Factorials:** $0! = 1$ (base case) and $n! = n \cdot (n - 1)!$ for $n > 0$.

- **Fibonacci numbers:** $f_0 = 0$, $f_1 = 1$ (base cases) and $f_n = f_{n-1} + f_{n-2}$.

- **Binomial coefficients:** $\binom{n}{0} = 1$, $\binom{n}{n} = 1$, and $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$ for $0 < k < n$. 


Recursion

Recursion is important in computer programming and in the foundations of the computer science.

• Strings: Let $\Sigma$ be an alphabet.
  1. The empty string $\lambda$ is a string.
  2. If $s$ is a string and $c \in \Sigma$ is a character, then $cs$ is a string.

• Palindromes: Let $\Sigma$ be an alphabet
  1. The empty string $\lambda$ is a palindrome.
  2. Let $c \in \Sigma$ be a character. Then $c$ is a palindrome.
  3. If $s$ is a palindrome and $c \in \Sigma$ is a character, then $csc$ is a palindrome.