1 5.3: Recursion

(1) Functions

(2) Sets

(3) Sequences (strings)

Factorials $n!,\ n \in \mathbb{Z}^{\geq 0}$

Base case: $0! = 1$

Recursive step: $(n+1)! = (n+1) \cdot n!$

Fibonacci ($n \in \mathbb{Z}^{\geq 0}$)

Base case: Fib(0) = 0

Fib(1) = 1

Recursive step: Fib(n+2) = Fib(n) + Fib(n+1)

Base case: $3 \in S$

Recursive step: if $x \in S$ and $y \in S$, then $x + y \in S$

List notation: $S = \{3, 6, 9, 12, 15, \dots\}$

Set builder: $S = \{3x |\ x \in \mathbb{Z}^{+}\}$

Σ is your alphabet. Examples: $\Sigma = \{0, 1\}$ or $\Sigma = \{A, T, C, G\}$, and so on.

Σ^* (Kleene closure) is the set of all strings that can be built from symbols in Σ. λ is an empty string.

Example: $\Sigma = \{0, 1\}$

$\Sigma^* = \{\lambda, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, \dots\}$

Recursive definition of Σ^*.

Base case: $\lambda \in \Sigma^*$

Recursive step: If $w \in \Sigma^*$ and $x \in \Sigma$, then $wx \in \Sigma^*$ (concatenation)

We will notice that the elements 0 and 1, in the previous example, are actually formed by $\lambda 0$ and $\lambda 1$

w is a string over alphabet Σ: $w \in \Sigma^*$

Define the reverse, w^R, recursively.

Base case: $\lambda^R = \lambda$

Recursive step: If $w \in \Sigma^*$ and $x \in \Sigma$, then $(wx)^R = x(w)^R$ (i.e, put the last character in the front position of the string)

Set of all palindromes over Σ. Call the set P.

Recursive definition:

Base case: $\lambda \in P$. If $x \in \Sigma, x \in P$.

Recursive step: If $w \in P$ and $x \in \Sigma$, then $xwx \in P$.