CS 2050 Worksheet 8

5.3 Recursive Definitions

1. Given that $f(n)$ is a function for all non-negative integers n, find $f(2), f(3)$, and $f(4)$ for each of the following recursive definitions:

a. $f(0) = 1$
$f(1) = 3$
$f(n+2) = f(n+1) + f(n)$

$f(0+2) = f(0+1) + f(0)$
$f(2) = f(1) + f(0) = 3 + 1 = 4$
$f(1+2) = f(2) + f(1) = 4 + 3 = 7 = f(3)$
$f(2+2) = f(3) + f(2) = 7 + 4 = 11 = f(4)$

b. $f(0) = 1$
$f(n+1) = \lfloor f(n) \rfloor + 0.5$

$f(0+1) = f(1) = \lfloor f(0) \rfloor + .5 = 1.5$
$f(1+1) = f(2) = \lfloor f(1) \rfloor + .5 = 1.5$
$f(2+1) = f(3) = \lfloor f(2) \rfloor + .5 = 1.5$
$f(4) = 1.5$

c. $f(0) = 1$
$f(1) = 2$
$f(n+2) = f(n+1)f(n)^2$

$f(2) = f(0+2) = f(1) \cdot f(0)^2 = 2 \cdot 1^2 = 2$
$f(3) = f(1+2) = f(2) \cdot f(1)^2 = 2 \cdot 2^2 = 8$
$f(4) = f(2+2) = f(3) \cdot f(2)^2 = 8 \cdot 2^2 = 32$

2. Recursively define the sequence a_n where $n \in \mathbb{Z}, n \geq 1$ for each of the following:

a. $a_n = 5n + 2$

$10+2$
a_j
$\cancel{5(n+1)+2}$
$5(n+1)+2$
$10n+5+2$

$a_1 = 5(1)+2 = 7$
$a_2 = 5(2)+2 = 12$
$a_3 = 5(3)+2 = 17$
$a_4 = 5(4)+2 = 22$
$a_5 = 5(5)+2 = 27$

base case: $a_1 = 7$

Recursive: $a_n = a_{n-1} + 5$
$n \geq 2$

b. $a_n = 3^n$

$a_1 = 3^1 = 3$
$a_2 = 3^2 = 9$
$a_3 = 3^3 = 27$
$a_4 = 3^4 = 81$

$\boxed{\begin{array}{l}\text{base case: } a_1 = 3 \\ \text{Recursive step: } a_{k+1} = 3 \cdot a_k \\ k \in \mathbb{Z} \quad k \geq 2\end{array}}$

try to use a_{k+1}, not a_{k-1}

3 9 15 21
$3 \cdot 1$ $3 \cdot 3$ $3 \cdot 5$ $3 \cdot 7$

the value we start on

Let S be a set of all positive odd multiples of 3.

3. Recursively define the set of all positive, odd multiples of 3.

Basis: $3 \in S$

Recursive: if $x \in S$, then $x + 6 \in S$

replaceX(x h y t x)

4. Recursively define the function $replaceX(s)$ that takes in a string of lowercase letters and replaces each character 'x' with the character 'y'.

base case: replaceX(λ) = λ

Recursive step: if $m \in \Sigma^*$ and $n \in \Sigma$ and $n =$ "x"

then replaceX(nm) = y . replaceX(m)

Option 1

TA method

Must define alphabet

let Σ = the set of all lowercase letters in the english alphabet.

Base case: replaceX(λ) = λ

t represents end char of string

Recursive Step: $w \in \Sigma^*$ and $t \in \Sigma$

IF t = 'x':

replaceX(wt) = replaceX(w) + 'y'

else

replaceX(wt) = replaceX(w) + t

Option 2

Base Case: replaceX(λ) = λ

replaceX('x') = 'y'

replaceX(b) = b where $b \in \Sigma$ and $b \neq$ 'x'

Recursive step: $w \in \Sigma^*$ and $t \in \Sigma$

replaceX(wt) = replaceX(w) + replaceX(t)