## Recursion

Recursive functions have two parts:

1) base case

2) recursive formula

ex. Fibonacci $F_i$

$$\left.\begin{array}{l} \left.\begin{array}{l} F_0 = 0 \\ F_1 = 1 \end{array}\right\} \text{base cases} \\ F_n = F_{n-1} + F_{n-2} \quad n \geq 2 \ \} \text{recursive formula} \end{array}\right\} \text{fibonacci sequence}$$

ex. summation from 0 to $n \geq 0$

$s(0) = 0$ base case

$s(n) = s(n-1) + n \quad n \geq 1$ recursive formula

Needs to recur down to base case eventually

ex. $f(x) = f(x-1) + 100$

not complete recursion bc no base case

ex. $g(x) = \begin{cases} g(x) = 0 & x = 7 \\ g(x) = g(x-1) + 2 & x > 6 \end{cases}$

invalid recursion bc $g(7)$ defined twice

Range of recursion:

1) cover every case

2) no duplicate definitions

3) base case(s)

## Unrolling

ex. Fibonacci: $F_0 = 0$, $F_1 = 1$, $F_n = F_{n-1} + F_{n-2} \quad n \geq 2$

$F_2 = F_1 + F_0 = 1 + 0 = 1$

$F_4 = F_3 + F_2 = F_2 + F_1 + F_1 + F_0 = F_1 + F_0 + F_1 + F_1 + F_0 = 1 + 0 + 1 + 1 + 0 = 3$

ex. $T: \mathbb{N} \to \mathbb{Z}$

$T(0) = 1$

$k = \underline{1}$ $\quad T(n) = 2T(n - \underline{1}) + 3 \quad n \geq 1 \qquad k =$ "level"

$k = \underline{2}$ $\quad = 2(2T(n-\underline{2}) + 3) + 3 = 2^2 T(n-2) + 2 \cdot 3 + 3$

$k = \underline{3}$ $\quad = 2^2(2T(n-3) + 3) + 2 \cdot 3 + 3 = 2^3 T(n-3) + 2^2 \cdot 3 + 2 \cdot 3 + 3$

⋮

$k=k \quad T(n) = 2^k T(n-k) + 3\sum_{i=0}^{k-1} 2^i$

At $T(0)$ (base case), $n-k=0$, $k=n$

$T(n) = 2^n T(0) + 3\sum_{i=0}^{n-1} 2^i$ final closed form

Finding closed form:

1) unroll, plug in previous iterations (start at n, go down (ie n, n-1, n-2, ... or k = 1, 2, ... k)
2) guess pattern
3) reach base case
4) plug in k value for base case

ex. $S(n) = \begin{cases} c & n=1 \\ 2S\left(\frac{n}{2}\right)+n & n \geq 2 \end{cases}$ $\quad n = 2^i \ \forall\, i \in \mathbb{N}$

$s\left(\frac{n}{2}\right) = 2s\left(\frac{n}{4}\right) + \frac{n}{2}$

$k=1 \quad S(n) = 2S\left(\frac{n}{2}\right)+n$

$k=2 \quad = 2\left(2S\left(\frac{n}{4}\right)+\frac{n}{2}\right)+n = 2^2 s\left(\frac{n}{4}\right)+2n$

$k=3 \quad = 2^2\left(2S\left(\frac{n}{8}\right)+\frac{n}{4}\right)+n+n = 2^3 S\left(\frac{n}{8}\right)+3n$

⋮

$k=k \quad = 2^k S\left(\frac{n}{2^k}\right)+kn$

base case when $n=1$, $\frac{n}{2^k}=1 \rightarrow k=\log_2 n$, plug in

$s(n) = 2^{\log_2 n} s\left(\frac{n}{2^{\log_2 n}}\right) + n\log_2 n = nS(1)+n\log_2 n = cn + n\log_2 n$ closed form

Discussion Problems - Invalid Recursion, 12.2bc

## Invalid recursion

Three of the following function definitions are not valid, each for a different reason. Which one is valid, and what's wrong with each of the others? (Domains and codomains are not provided, but assume they're always something sensible.)

$$f(n) = \begin{cases} 8 & \text{when } n < 9 \\ n + f(n-2) & \text{when } n \geq 9 \end{cases}$$ valid

$$g(n) = \begin{cases} -9 & \text{when } n = 6 \\ n + g(n-1) & \text{when } n > 7 \end{cases}$$ invalid, doesn't cover g(7)

$$h(n) = \begin{cases} 3 & \text{when } n = 6 \\ n + h(n+1) & \text{when } n \geq 7 \end{cases}$$ invalid, never reaches a base case

$$s(n) = \begin{cases} 2 & \text{when } n \leq 7 \\ n + s(n-1) & \text{when } n > 6 \end{cases}$$ invalid, two definitions for s(7)

## 12.2 Unrolling

Find closed forms for the following recursive definitions using unrolling. Specifically, show at least two steps of unrolling, a summation whose value is equal to $T(n)$, and finally a closed-form expression (i.e. containing no recursion or summations) equal to $T(n)$.

(a) $T : \mathbb{Z}^+ \to \mathbb{Z}^+$ defined by

$T(1) = 1$
$T(n) = 2T(n-1) + 3$

(b) $f : \mathbb{N}+ \to \mathbb{N}$ defined by

$f(0) = 0$
$f(n) = 5f(n-1) + 1$

(c) $T : \mathbb{Z}^+ \to \mathbb{Z}^+$ defined (powers of 3 only) by

$T(1) = 47$
$T(n) = 3T(n/3) + 13n$

b) $f : \mathbb{N}^+ \to \mathbb{N}$

$f(0) = 0$

$f(n) = 5f(n-1) + 1$

$k=1$ $\quad f(n) = 5f(n-1) + 1$

$k=2$ $\quad = 5(5f(n-2)+1) + 1 = 5^2 f(n-2) + 5\cdot 1 + 1$

$k=3$ $\quad = 5^2(5f(n-3)+1) + 5\cdot 1 + 1 = 5^3 f(n-3) + 5^2\cdot 1 + 5\cdot 1 + 5$

$\vdots$

$k=k$ $\quad = 5^k f(n-k) + 5^{k-1} + 5^{k-2} + \ldots + 5^1 + 5^0 = 5^k f(n-k) + \sum_{i=0}^{k-1} 5^i$

base case when $n-k=0$, $k=n$

$k=n$ $\quad = 5^n f(0) + \sum_{i=0}^{n-1} 5^i = \sum_{i=0}^{n-1} 5^i = \frac{5^n - 1}{4}$ closed form

c) $T : \mathbb{Z}^+ \to \mathbb{Z}^+$

$T(1) = 47$

$T(n) = 3T\left(\frac{n}{3}\right) + 13n$

$k=1$ $\quad T(n) = 3T\left(\frac{n}{3}\right) + 13n$

$k=2$ $$= 3\left(3T\left(\frac{n}{9}\right) + \frac{13}{3}n\right) + 13n = 3^2\,T\left(\frac{n}{9}\right) + 13n + 13n$$

$k=3$ $$= 3^2\left(3T\left(\frac{n}{27}\right) + \frac{13}{9}n\right) + 13n + 13n = 3^3\,T\left(\frac{n}{27}\right) + 13n + 13n + 13n$$

$\vdots$

$k=k$ $$= 3^k\,T\left(\frac{n}{3^k}\right) + 13kn$$

for base case, $\frac{n}{3^k} = 1 \Rightarrow n = 3^k \Rightarrow k = \log_3 n$

$k=\log_3 n$ $$= 3^{\log_3 n}\,T\left(\frac{n}{3^{\log_3 n}}\right) + 13n\log_3 n = nT(1) + 13n\log_3 n = 47n + 13n\log_3 n$$ closed form