Inductive / Recursive Definitions

A sequence can be defined
① using an explicit formula, or
② by recursion (later terms defined by earlier terms)

eg. 1, 2, 4, 8, 16, ...

$a_n = 2^n$ for $n = 0, 1, 2$ [explicit formula]

$a_0 = 1$ [recursive definition]

$a_{n+1} = 2a_n$ for $n \geq 0$

using regular ind

Recursively Defined Functions

Basis Define $f(0)$

Recursive step Define $f(k+1)$ based on $f(0), ..., f(k)$.

eg. $f(0) = 3$

$f(n+1) = 2f(n) + 3 \rightarrow$ 3, 9, 21, 45, 93, ...

eg. $F(n) = n! \rightarrow F(0) = 1$

$F(n+1) = (n+1)F(n)$

eg.. $f(n) = a^n \quad \rightarrow \quad f(0) = 1$

$f(n+1) = a\,f(n)$

eg, $s(1) = 1, \quad s(n+1) = \sum_{i=1}^{n} s(i)$ ← using strong ind

1, 1, 2, 4, 8, 16, ...

<u>Def</u> The Fibonacci numbers $f_0, f_1, f_2, \ldots$ are defined by

$f_0 = 0, \quad f_1 = 1$ (base cases)

$\underbrace{f_n = f_{n-1} + f_{n-2}}_{\text{using strong ind}}, \quad n \geq 2$ (recursive step)

We can prove properties of recursively defined functions using induction.

eg. Prove that $f_1^2 + f_2^2 + \cdots + f_n^2 = f_n \cdot f_{n+1}, \quad \forall n \geq 1.$

<u>Base Case</u> $(n=1)$

Since $f_1^2 = 1$ and $f_1 \cdot f_2 = f_1(f_0 + f_1) = 1$,

it follows $f_1^2 = f_1 \cdot f_2$.

<u>Induction step</u>

Assume $f_1^2 + f_2^2 + \cdots + f_k^2 = f_k \cdot f_{k+1}$

Prove $f_1^2 + f_2^2 + \cdots + f_k^2 + f_{k+1}^2 = f_{k+1} \cdot f_{k+2}$

$$
\begin{aligned}
& (f_1^2 + f_2^2 + \cdots + f_k^2) + f_{k+1}^2 \\
&= f_k \cdot f_{k+1} + f_{k+1}^2 && \text{by IH} \\
&= f_{k+1}(f_k + f_{k+1}) \\
&= f_{k+1} \cdot f_{k+2} && \text{by Fibonacci def}
\end{aligned}
$$

QED

We can define a set inductively/
recursively

Call these recursively defined data types

<u>Basis</u> Place some elements initially in set

<u>Ind step</u> Include new elements in set, using rules based on elements already in set

* Only elements inserted in this way are included in the set.

Define the set $S \subseteq \mathbb{Z}$ as follows:

<u>Basis</u>: $3 \in S$

<u>Ind</u>: if $x \in S$ and $y \in S$
then $x + y \in S$

Basis → 3

$\Rightarrow S = \{3, 6, 9, 12, \ldots\}$ Ind → 6

$= \{3n \mid n \in \mathbb{Z}^+\}$ Ind → 9, 12

Ind → 15, 18, 21, 24

eg. Define a well-formed logical formula over T, F, propositional variables p, q, r and operators $\wedge, \vee, \neg, \rightarrow$

Basis $T, F, p, q, r \in WFF$

Ind if $A \in WFF$ and $B \in WFF$
then $[\neg A]$, $[A \wedge B]$, $[A \vee B]$, $[A \rightarrow B] \in WFF$

eg. $[[[p \wedge q] \vee r] \rightarrow [q \wedge r]] \in WFF$

Let Σ be an alphabet.
eg. $\Sigma = \{0,1\}$, $\Sigma = \{a,b\}$, $\Sigma = \{a, b, ..., z\}$

$\Sigma^* =$ set of words or strings over Σ

eg. $\Sigma = \{a,b\}$
$\Sigma^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa ...\}$
countable ↗ ↖ empty string

Define Σ^* recursively:
Basis $\varepsilon \in \Sigma^*$
Ind if $w \in \Sigma^*$, $\sigma \in \Sigma$ then $w\sigma \in \Sigma^*$

Basis $\rightarrow \varepsilon$
Ind $\rightarrow a, b$
Ind $\rightarrow aa, ab, ba, bb$
Ind $\rightarrow aaa, aab, aba, abb, baa, bab, bba, bbb$

Given an inductively defined set S

- we can define operations and functions on these sets inductively / recursively

- we can prove properties of these sets inductively / recursively (structural induction)

Define length of a string in Σ^* inductively

Basis: $\text{length}(\varepsilon) = 0$

Ind if $w \in \Sigma^*$, $\sigma \in \Sigma$
then $\text{length}(w\sigma) = \text{length}(w) + 1$

Define reversal of a string $x \in \Sigma^*$ inductively, denoted x^R.

Basis if $x = \varepsilon$ then $x^R = \varepsilon$

Ind step if $x = w\sigma$ where $w \in \Sigma^*$, $\sigma \in \Sigma$
then $x^R = \sigma w^R$

or

Basis $\varepsilon^R = \varepsilon$

Ind if $w \in \Sigma^*$, $\sigma \in \Sigma$
then $(w\sigma)^R = \sigma w^R$