Many things can be defined recursively

Recursive definition of string reversal \( \text{rev} \)
\[ \text{rev}(s) \text{- reverse string of } s \]

Example: \( \text{rev}("dina") = "anid" \)

\[
\begin{align*}
\text{if } s &= "", \text{rev}(s) = "" \\
\text{if } s &= s'.c, \text{rev}(s) &= c.\text{rev}(s')
\end{align*}
\]

\( \text{"dina" = "din".a} \)
\( \text{rev("dina") = a.rev("din") = a."nid" = "anid"} \)
\( \text{rev("din") = rev("di".n) = n.rev("di") = n."id" = "nid"} \)

Recursive definition of string length \( \text{len} \)
\[ \text{len}(s) \text{- length of string } s \]

Ex: \( \text{len}("dina") = 4 \)

\[
\begin{align*}
\text{if } s &= "", \text{len}(s)= 0 \\
\text{if } s &= s'.c \text{len}(s) &= \text{len}(s') + 1
\end{align*}
\]

\( \text{len("dina") = len("din".a) = len("din") + 1 = 4} \)
\( \text{len("din") = len("di".n) = len("di") + 1 = 3} \)

Recursive definition of set of \( k \) elements, in terms of set of \( k-1 \) elements
\[
\{x_1, x_2, \ldots x_k\} = \{x_1, x_2, \ldots x_{k-1}\} \cup \{x_k\}
\]

Recursive definition of powerset
\( P_k = \text{power set of set } \{x_1, x_2, \ldots x_k\} \)

\[
\begin{align*}
P_0 &= \text{power set of } \{\} = \{\emptyset\} \\
P_k &= \text{power set of set } \{x_1, x_2, \ldots x_k\} = \text{power set of } (\{x_1, x_2, \ldots x_{k-1}\} \cup \{x_k\}) = ??
\end{align*}
\]

Note: \( P_k \) has two types of sets -- (a) those that do not contain \( x_k \) and (b) those that do
(a) is just $P_{k-1}$
(b) is a copy of $P_k$ but $x_k$ with added to every set

Example:

$P_0 = \{\emptyset\}$
$P_1 = \{\emptyset, \{x1\}\}$
$P_2 = \{\emptyset, \{x1\}, \{x2\}, \{x1, x2\}\}$ -- bold part is same as first part but with $x_2$ added to each set
$P_3 = \{\emptyset, \{x1\}, \{x2\}, \{x1, x2\}, \{\{x3\}, \{x1,x3\}, \{x2,x3\}, \{x1, x2,x3\}\}$

So here is the recursive definition:

$P_k = \text{power set of } \{} \{} = \{\emptyset\} \quad \text{if } k = 0$

$P_k = P_{k-1} \cup \text{combine}(P_k, x_k) \quad \text{if } k > 0$

where $\text{combine}(S, x)$ is our set operation of taking a set of sets $S$ and adding element $x$ to every set in $S$

Recursive definition of EVENS

$\text{EVENS} = \{0, 2, 4, 6, 8 \ldots\}$

What are the members of Evens?

$0 \in \text{EVENS} \quad \text{// base case}$

$x \not\in \text{EVENS} \quad \text{if } x < 0$

$x \in \text{EVENS} \text{ if } x-2 \in \text{EVENS} \quad \text{//RECURSIVE, FOR } X > 0$

Is 6 in EVENS? YES

$6-2 = 4$, is 4 events? YES!!
$4-2 = 2$, is 2 in evens? YES
$2-2 = 0$, is 2 in evens? BASE CASE, YES

Is 3 in evens? NO

$3-2 = 1$, is 1 in EVENS? NO
$1-2 = -1$, is -1 in EVENS? Base case, NO

Recursive definition of Fibonacci series

$\text{fib}(0) = 1$
$\text{fib}(1) = 1$
\[ \text{fib}(k) = \text{fib}(k-1) + \text{fib}(k-2) \text{ if } k \geq 2 \]

Set of all propositions over some set of variables can also be defined recursively!! (in 3502)

NOTE: EVERY CLAIM TO BE PROVED RECURSIVELY INVOLVES SOME "FOR ALL"

Claim example: for all \( k \geq 0 \), \( \text{fib}(k) > \) some formula over \( k \)

\( S_k = \text{size of Powerset of } k \text{ elements} = \text{size of } P_k \)
Claim: (for all \( k \geq 0 \)) \( S_k = \text{size of } P_k = 2^k \)

Reminder of definition

\[ P_0 = \text{power set of } \emptyset = \{\emptyset\} \quad \text{if } k = 0 \]
\[ P_k = P_{k-1} \cup \text{combine}(P_{k-1}, x_k) \quad \text{if } k > 0 \]

Proof: // recursive

**Base case:** \( k = 0 \),
we want to prove that \( S_0 = 2^0 = 1 \)
\( S_k = S_0 = \text{size of } P_0 = \text{size of } \{\emptyset\} = 1 \)
QED

**INDUCTIVE ASSUMPTION (IA)**
Assume that \( S_{k-1} = \text{size of } P_{k-1} = 2^{k-1} \)

**RECURSIVE STEP**
1. Consider \( S_k = \text{size of } P_k \)
2. By definition, \( P_k = P_{k-1} \cup \text{combine}(P_{k-1}, x_k) \)
3. They are DISJOINT because no sets in \( P_{k-1} \) have \( x_k \), and all sets in \( \text{combine}(P_{k-1}, x_k) \) have \( x_k \)
4. So size of \( P_k = \text{size of } P_{k-1} + \text{size of } \text{combine}(P_{k-1}, x_k) \)
5. By IA, size of \( P_{k-1} = S_{k-1} = 2^{k-1} \)
6. Also, by construction, size of \( \text{combine}(P_{k-1}, x_k) = \text{size of } P_{k-1} = 2^{k-1} \)
7. therefore, \( S_k = \text{size of } P_k = 2^{k-1} + 2^{k-1} = 2 \cdot 2^{k-1} = 2^k \)
QED

**RECURSIVE PROOF (PROOF BY INDUCTION)** -- always based on a recursive definition
1. PROVE THE BASE CASE(S) - as many as you have base cases in your definition
TAKE YOUR CLAIM, PLUG IN THE VALUE OF THE BASE CASE, PROVE WHAT YOU GET
2. MAKE THE INDUCTIVE ASSUMPTION!! AKA RECURSIVE ASSUMPTION
THIS IS YOUR RECURSIVE CALL TO A "SMALLER CASE"
TAKE YOUR CLAIM, PLUG IN THE "SMALLER CASE"
NO NEED TO PROVE IT!!

3. INDUCTIVE STEP "PUTTING IT ALL TOGETHER"
THIS IS EXECUTING THE FUNCTION
USING THE RESULT OF THE RECURSIVE CALL
USING THE RECURSIVE DEFINITION!

TILING A $2^k$ BY $2^k$ board ($k$-board) by L-SHAPED 3-tiles

**goal:** fill up the whole board except for one corner piece!!!
If we can do this, then we have tiled the board "successfully"

smallest is 2x2
4x4, 8x8, 16x16 etc etc

NOTE: each board is just 4 smaller boards stuck together
(this is where the recursion comes in)

![Diagram of a 2x2 board tiled with L-shaped 3-tiles](image)

**CLAIM:** ANY $k$-BOARD can be tiled successfully!!

**PROOF** (recursive)
1. **BASE CASE:** 2-board
   HERE IS HOW YOU DO IT

![Diagram of a 2x2 board](image)

2. **INDUCTIVE ASSUMPTION:**
   ANY ($k$-1)-BOARD can be tiled **successfully**!!

3. **RECURSIVE STEP:**
   1. Consider a $k$-board, it consists of 4 $k$-1 boards
2. By inductive assumption, each of these boards can be tiled successfully
3. Note that we can rotate each of these boards so the untiled corner piece faces the middle
4. Now we have a 2x2 empty area in the middle, put a tile there!
5. Now we've tiled the whole thing, with just one tile left
6. can rotate the k-l board with the last empty piece so it's back in the corner
QED