Today's learning goals

- Explain the steps in a proof by mathematical and/or structural induction
- Use a recursive definition of a set to determine the elements of the set
- Write a recursive definition of a given set
- Use structural induction to prove properties of a recursively defined set
- Define functions from $\mathbb{N}$ to $\mathbb{R}$ using recursive definitions
Mathematical induction

To show that some statement $P(k)$ is true about all nonnegative integers $k$,

1. Show that it’s true about 0 i.e. $P(0)$
2. Show $\forall k \ ( P(k) \rightarrow P(k + 1) )$ Hence conclude $P(1), \ldots$
Mathematical induction*

To show that some statement $P(k)$ is true about all nonnegative integers $k \geq b$,

1. Show that it’s true about $b$ i.e. $P(b)$
2. Show $\forall k \ ( P(k) \rightarrow P(k + 1) )$ Hence conclude $P(b+1),\ldots$
Sizes of (finite) sets

If $S$ is a set with exactly $n$ distinct elements, with $n$ a nonnegative integer, then $S$ is finite set and $|S| = n$.

$|\{\}\,| = 0 \quad |\{1\}\,| = 1 \quad |\{1,1,2,3,8/4\}| = 3$

For $A, B$ finite sets:

$|A \cup B| = |A| + |B| - |A \cap B|$

$|A \times B| = |A| \cdot |B|$

if $A$ has size $n$, then the power set of $A$ has size $2^n$. An infinite set is a set that is not finite.
To specify a function, we need to specify its

1. **domain** – input
2. **codomain** – type of outputs
3. **assignment / rule** – formula, table of values, induction

Rosen p. 138
Recursive definitions, part 1

For functions $f: \mathbb{N} \rightarrow X$
- define by a (closed-form) formula

$f(0)$  $f(1)$  $f(2)$  $f(3)$  $f(4)$

1. domain – $\mathbb{N}$
2. codomain – $X$
3. assignment / rule – formula, table of values, induction
Recursive definitions, part 1

For functions  \( f: \mathbb{N} \rightarrow X \)
- define by a (closed-form) formula

\[
\begin{align*}
  f(0) & \quad f(1) & \quad f(2) & \quad f(3) & \quad f(4) & \quad \ldots \\
  a_0 & \quad a_1 & \quad a_2 & \quad a_3 & \quad a_4 & \quad \ldots
\end{align*}
\]

1. domain – \( \mathbb{N} \)
2. codomain – \( X \)
3. assignment / rule – formula, table of values, induction
Recursive definitions, part 1

For functions \( f : \mathbb{N} \to X \)
- define by a (closed-form) formula or …

**Basis step:** Specify the value of the function at 0

**Recursive step:** Give a rule for finding its value at an integer from its values at smaller integers

Which of the following functions / sequences have recursive definitions?

A. \( n! = n \cdot (n-1) \cdot (n-2) \cdots 3 \cdot 2 \cdot 1 \)
B. \( 2^n \)
C. 2, -8, 32, -128, 512, …
D. \( \sum_{i=1}^{n} (i^2 + i) \)
E. All of the above.
Recursive definitions, part 2

- **Base step**
  - $0! = 1,$
  - $2^0 = 1,$
  - $a_0 = 2,$

- **Recursive step**
  - $n! = n(n-1)!$ for $n > 0.$
  - $2^n = 2(2^{n-1})$ for $n > 0.$
  - $a_n = -4a_{n-1}$ for $n > 0.$

2, -8, 32, -128, 512, ...

\[ \sum_{i=1}^{n} g(n) = g(1) + g(2) + \cdots + g(n) \]

\[ \sum_{i=1}^{0} g(i) = 0 , \quad \sum_{i=1}^{1} g(i) = g(1) , \quad \text{and} \quad \sum_{i=1}^{n} g(i) = g(n) + \sum_{i=1}^{n-1} g(i) \quad \text{for } n > 1. \]
Extra mathematical induction practice

1. \( n^2 < n! \) for all \( n \) greater than or equal to …
   
   a.k.a. \( n^2 \) is \( O(n!) \)

2. For all \( n \geq 1 \)
   
   \[ \sum_{i=1}^{n} i = \frac{n(n+1)}{2} \]
   
   Proof: By MI
   
   Basis Step: \( WTS \ for \ n=1 \)
   
   \[ \sum_{i=1}^{1} i = \frac{1(1+1)}{2} \]

3. For all \( n \geq 1 \)
   
   \[ \sum_{i=1}^{n} \frac{1}{i(i+1)} = \frac{n}{n+1} \]
   
   Proof: By MI
   
   Basis Step: \( WTS \ for \ n=1 \)
   
   \[ \sum_{i=1}^{1} \frac{1}{i(i+1)} = \frac{1}{1(1+1)} = \frac{1}{2} \]
   
   LHS = \( \frac{1}{2} \) \( \Rightarrow \) \( RHS = \frac{1}{2} \)
Recursive definitions, part 2

Define a set $S$ by

\[
\{ \ldots \} \\
\{ x \mid P(x) \}
\]

**Recursive definition:**

*Basis step* – Specify initial collection of elements.

*Recursive step* – Provide rules for forming new elements in the set from those already known to be in the set.

\[
e \in \mathbb{N} = \{0, 1, 2, 3, 4, \ldots \}
\]

\[
\text{if } n \in \mathbb{N} \text{ then } n+1 \in \mathbb{N}
\]
Example: \( S^\infty = \{ x \mid x \text{ is an int, } x > 3 \} \)

**Basis Step** \( 4 \in S \)

**Rec Step** \( \text{if } n \in S, \ n+1 \in S \)
Recursive definitions, part 2

Let S be the subset of the set of integers defined recursively by

\[\begin{align*}
\text{Basis step:} & \quad 1 \in S \\
\text{Recursive step:} & \quad \text{If } a \in S, \text{ then } a + 2 \in S.
\end{align*}\]

What's an equivalent description of this set?

A. All positive multiples of 2.
B. All positive even integers.
C. All positive odd integers.
D. All integers.
E. None of the above.

\[\{1, 3, 5, 7, \ldots\}\]
Recursive definitions, part 2

The set of **bit strings** \( \{0, 1\}^* \) is defined recursively by

**Basis step:** \( \lambda \in \{0, 1\}^* \), where \( \lambda \) is the empty string.

**Recursive step:** If \( w \in \{0, 1\}^* \), then \( w0 \in \{0, 1\}^* \) and \( w1 \in \{0, 1\}^* \).

Also known as **binary sequences**
The set of bit strings \( \{0,1\}^* \) is defined recursively by

**Basis step:** \( \lambda \in \{0,1\}^* \) where \( \lambda \) is the empty string.

**Recursive step:** If \( w \in \{0,1\}^* \), then \( w0 \in \{0,1\}^* \) and \( w1 \in \{0,1\}^* \).

Which of the following are not bit strings?

A. \( \lambda \)
B. 1
C. 000
D. 010101
E. 101100111000111100001111100000111…

Also known as binary sequences.
Recursive definitions in CS

• Strings – encode integers, everything else
• Data structures – linked lists, trees, graphs

How do we prove a fact is true about all strings?
Structural induction

To show that some statement P(k) is true about all elements of a recursively defined set S,

1. Show that it's true for each element specified in the basis step to be part of S.
2. Show that if it's true for each of the elements used to construct new elements in recursive step, then it holds for these new elements.
Define the subset $S$ of the set of all bit strings, $\{0,1\}^*$, by

**Basis step:** $\lambda \in S$ where $\lambda$ is the empty string.

**Recursive step:** If $w \in S$, then each of $10w \in S, 01w \in S$

$S = \{ \lambda, 10, 01, 1010, 0110, 1001, 0101, \ldots \}$
Structural induction, example  \(\text{Rosen Sec 5.3}\)

Define the **subset** \(S\) of the set of all bit strings, \(\{0,1\}^*\), by

- **Basis step**: \(\lambda \in S\) where \(\lambda\) is the empty string.
- **Recursive step**: If \(w \in S\), then each of \(10w \in S, 01w \in S\)

**Claim**: Every element in \(S\) has an equal number of 0s and 1s.

**Pf by structural induction**

- **Basis Step**: \(\lambda\) has 0 0s and 0 1s, so equal. \(\checkmark\)
Structural induction, example  

Rosen Sec 5.3

Define the **subset** $S$ of the set of all bit strings, $\{0,1\}^*$, by

**Basis step:** $\lambda \in S$ where $\lambda$ is the empty string.

**Recursive step:** If $w \in S$, then each of $10w \in S, 01w \in S$

**Claim:** Every element in $S$ has an equal number of 0s and 1s.

**Proof:**

*Basis step* – **WTS** that empty string has equal # of 0s and 1s

*Recursive step* – Let $w$ be an arbitrary element of $S$.

Assume, as the **IH** that $w$ has equal # of 0s and 1s.

**WTS** that $10w, 01w$ each have equal # of 0s, 1s.
Recursive definitions, part 3

Rosen Sec 5.3

Define a function $f : S \rightarrow X$ by

- specifying $f(a)$ when $a$ is in the basis collection of elements of $S$
- giving a formula for computing $f(x)$ when $x$ is built from other elements $a_1, a_2$ etc. based on values of $f(a_1), f(a_2)$, etc.

1. domain – $S$, a recursively defined set
2. codomain – $X$
3. assignment / rule – recursively defined
Define a function $f: S \rightarrow X$ by

- specifying $f(a)$ when $a$ is in the basis collection of elements of $S$
- giving a formula for computing $f(x)$ when $x$ is built from other elements $a_1, a_2$ etc. based on values of $f(a_1), f(a_2)$, etc.

Example: how do you define the length function $l: \{0,1\}^* \rightarrow \mathbb{N}$?

- **Basis Step**: $l(\lambda) = 0$
- **Recursive Step**: If $w\in \{0,1\}^*$, $l(wo) = l(w) + 1$, $l(w1) = l(w) + 1$
Reminders

• Office hours may be shifted – check Google calendar
• HW5 due Saturday night

**Next class**: more applications + variants of induction