Review: Strong Induction

- Base case same as regular induction, different in inductive step
- Regular induction: assume $P(k)$ holds and prove $P(k + 1)$
- Strong induction: assume $P(1), P(2), \ldots, P(k)$; prove $P(k + 1)$

A Word about Base Cases

- In all examples so far, we had only one base case
  - i.e., only proved the base case for one integer
- In some inductive proofs, there may be multiple base cases
  - i.e., prove base case for the first $k$ numbers
- In the latter case, inductive step only needs to consider numbers greater than $k$

Example

- Prove that every integer $n \geq 12$ can be written as $n = 4a + 5b$ for some non-negative integers $a, b$.
- Proof by strong induction on $n$ and consider 4 base cases
  - Base case 1 ($n=12$): $12 = 3 \cdot 4 + 0 \cdot 5$
  - Base case 2 ($n=13$): $13 = 2 \cdot 4 + 1 \cdot 5$
  - Base case 3 ($n=14$): $14 = 1 \cdot 4 + 2 \cdot 5$
  - Base case 4 ($n=15$): $15 = 0 \cdot 4 + 3 \cdot 5$

Example, cont.

- Inductive hypothesis: Suppose every $12 \leq i \leq k$ can be written as $i = 4a + 5b$.
- Inductive step: We want to show $k + 1$ can also be written this way for $k + 1 \geq 16$
  - Observe: $k + 1 = (k - 3) + 4$
  - By the inductive hypothesis, $k - 3 = 4a + 5b$ for some $a, b$ because $k - 3 \geq 12$
  - But then, $k + 1$ can be written as $4(a + 1) + 5b$

Another Example

- For $n \geq 1$, prove there exist natural numbers $a, b$ such that: $5^n = a^2 + b^2$
  - Insight: $5^{k+1} = 5^2 \cdot 5^{k-1}$
Matchstick Example

- The Matchstick game: There are two piles with same number of matches initially
- Two players take turns removing any positive number of matches from one of the two piles
- Player who removes the last match wins the game
- Prove: Second player always has a winning strategy.

Matchstick Proof

- \( P(n) \): Player 2 has winning strategy if initially \( n \) matches in each pile
- Base case:
  - Induction: Assume \( \forall j. 1 \leq j \leq k \rightarrow P(j) \); show \( P(k+1) \)
- Inductive hypothesis:
  - Prove Player 2 wins if each pile contains \( k + 1 \) matches

Matchstick Proof, cont.

- Case 1: Player 1 takes \( k + 1 \) matches from one of the piles.
  - What is winning strategy for player 2
- Case 2: Player 1 takes \( r \) matches from one pile, where \( 1 \leq r \leq k \)
  - Now, player 2 takes \( r \) matches from other pile
  - Now, the inductive hypothesis applies \( \Rightarrow \) player 2 has winning strategy for rest of the game

Recursive Definitions

- Should be familiar with recursive functions from programming:
  ```java
  public int fact(int n) {
    if(n <= 1) return 1;
    return n * fact(n - 1);
  }
  ```
  - Recursive definitions are also used in math for defining sets, functions, sequences etc.

Recursive Definitions in Math

- Consider the following sequence:
  \[ 1, 3, 9, 27, 81, \ldots \]
- This sequence can be defined recursively as follows:
  \[
  a_0 = 1 \\
  a_n = 3 \cdot a_{n-1}
  \]
  - First part called base case; second part called recursive step
  - Very similar to induction; in fact, recursive definitions sometimes also called inductive definitions

Recursively Defined Functions

- Just like sequences, functions can also be defined recursively
- Example:
  \[
  f(0) = 3 \\
  f(n+1) = 2f(n) + 3 \quad (n \geq 1)
  \]
  - What is \( f(1) \)?
  - What is \( f(2) \)?
  - What is \( f(3) \)?
Recursive Definition Examples

- Consider \( f(n) = 2n + 1 \) where \( n \) is non-negative integer
- What’s a recursive definition for \( f \)?
- Consider the sequence \( 1, 4, 9, 16, \ldots \)
- What is a recursive definition for this sequence?
- Recursive definition of function defined as \( f(n) = \sum_{i=1}^{n} i ?

Recursive Definitions of Important Functions

- Some important functions/sequences defined recursively
- Factorial function:
  \[
  f(1) = 1 \\
  f(n) = n \cdot f(n-1) \quad (n \geq 2)
  \]
- Fibonacci numbers: \( 1, 1, 2, 3, 5, 8, 13, 21, \ldots \)
  \[
  a_1 = 1 \\
  a_2 = 1 \\
  a_n = a_{n-1} + a_{n-2} \quad (n \geq 3)
  \]
- Just like there can be multiple base cases in inductive proofs, there can be multiple base cases in recursive definitions

Inductive Proofs for Recursively Defined Structures

- Recursive definitions and inductive proofs are very similar
- Natural to use induction to prove properties about recursively defined structures (sequences, functions etc.)
- Consider the recursive definition:
  \[
  f(0) = 1 \\
  f(n) = f(n-1) + 2
  \]
- Prove that \( f(n) = 2n + 1 \)

Example

- Let \( f_n \) be \( n \)’th element in the Fibonacci sequence \((n \geq 1)\)
- Prove: For \( n \geq 3 \), \( f_n > \alpha^{n-2} \) where \( \alpha = \frac{1 + \sqrt{5}}{2} \)
  - Proof is by strong induction on \( n \) with two base cases
  - Intuition 1: Definition of \( f_n \) has two base cases
  - Intuition 2: Recursive step uses \( f_{n-1}, f_{n-2} \Rightarrow \) strong induction
  - Base case 1 \((n=3)\): \( f_3 = 2 \), and \( \alpha < 2 \), thus \( f_3 > \alpha \)
  - Base case 2 \((n=4)\): \( f_4 = 3 \) and \( \alpha^2 = \frac{(\alpha + \sqrt{5})^2}{2} < 3 \)

Example, cont.

Prove: For \( n \geq 3 \), \( f_n > \alpha^{n-2} \) where \( \alpha = \frac{1 + \sqrt{5}}{2} \)

- Inductive step: Assuming property holds for \( f_i \) where \( 3 \leq i \leq k \), need to show \( f_{k+1} > \alpha^{k-1} \)
- First, rewrite \( \alpha^{k-1} \) as \( \alpha^2 \alpha^{k-3} \)
- \( \alpha^2 \) is equal to \( 1 + \alpha \) because:
  \[
  \alpha^2 = \left( \frac{1 + \sqrt{5}}{2} \right)^2 = \frac{\sqrt{5} + 3}{2} = \alpha + 1
  \]
- Thus, \( \alpha^{k-1} = (\alpha + 1)(\alpha^{k-3}) = \alpha^{k-2} + \alpha^{k-3} \)
## Recursively Defined Sets and Structures

- We saw how to define functions and sequences recursively.
- We can also define sets and other data structures recursively.

**Example:** Consider the set $S$ defined as:

- $3 \in S$
- If $x \in S$ and $y \in S$, then $x + y \in S$

- What is the set $S$ defined as above?

---

## Strings and Alphabets

- Recursive definitions play an important role in the study of strings.
- Strings are defined over an alphabet $\Sigma$.
  - Example: $\Sigma_1 = \{a, b\}$
  - Example: $\Sigma_2 = \{0\}$

- Examples of strings over $\Sigma_1$: $a$, $b$, $aa$, $ab$, $ba$, $bb$, ...

- Set of all strings formed from $\Sigma$ forms a language called $\Sigma^*$.
  - $\Sigma_2^* = \{\epsilon, 0, 00, 000, \ldots\}$

---

## Recursive Definition of Strings

- The language $\Sigma^*$ has a natural recursive definition:
  - Base case: $\epsilon \in \Sigma^*$ (empty string)
  - Recursive step: If $w \in \Sigma^*$ and $x \in \Sigma$, then $wx \in \Sigma^*$

- Since $\epsilon$ is the empty string, $\epsilon s = s$

- Consider the alphabet $\Sigma = \{0, 1\}$

- How is the string "1" formed according to this definition?

- How is "10" formed?

---

## Another Example

- The reverse of a string $s$ is $s$ written backwards.

- Example: Reverse of "abc" is "cba"

- Give a recursive definition of the reverse($s$) operation.
  - Base case:
  - Recursive step:
Palindromes

- A palindrome is a string that reads the same forwards and backwards.
- Examples: "mom", "dad", "abba", "Madam I'm Adam", ...
- Give a recursive definition of the set $P$ of all palindromes over the alphabet $\Sigma = \{a, b\}$
  - Base cases:
  - Recursive step:

Bitstrings

- A bitstring is a string over the alphabet $\{0, 1\}$
- Give a recursive definition of the set $S$ of bitstrings that contain equal number of 0’s and 1’s.
  - Base case:
  - Recursion: