Structural Induction

250H

Recursive Definitions for Functions

- Sometimes it is hard to define an object explicitly, but we can define it in terms of itself

Recursive Definitions for Functions

- Sometimes it is hard to define an object explicitly, but we can define it in terms of itself
- Fibonacci: {1, 1, 2, 3, 5, 8, 13, ...}
  - Recursive Definition:

$$F_n = F_{n-1} + F_{n-2}$$

Recursive Definitions for Functions

- Sometimes it is hard to define an object explicitly, but we can define it in terms of itself
- Fibonacci: {1, 1, 2, 3, 5, 8, 13, ...}
  - Recursive Definition:

$$F_n = F_{n-1} + F_{n-2}$$

  - Closed form:

$$F_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$$

Example

Define a function with the set of nonnegative integers as its domain:

Example

Define a function with the set of nonnegative integers as its domain:

- Base Step: Specify the value of the function at zero

Example

Define a function with the set of nonnegative integers as its domain:

- Base Step: Specify the value of the function at zero
- Recursive Step: Give a rule for finding its value at an integer from its values at smaller integers

Give a recursive definition of a^n, where a is a nonzero real number and n is a nonnegative integer.

Give a recursive definition of a^n, where a is a nonzero real number and n is a nonnegative integer.

There are two parts: the base step and the recursive step

Give a recursive definition of a^n, where a is a nonzero real number and n is a nonnegative integer.

There are two parts: the base step and the recursive step

For n = {0, 1, 2, 3, ...}

- Base step: $a^0 = 1$

Give a recursive definition of a^n, where a is a nonzero real number and n is a nonnegative integer.

There are two parts: the base step and the recursive step

For $n = \{0, 1, 2, 3, \ldots\}$

- Base step: $a^0 = 1$
- Recursive step: $a^{n+1} = a(a^n)$

Recursive Definitions for Sets and Structures

- Sets can also be defined recursively

Recursive Definitions for Sets and Structures

- Sets can also be defined recursively
- We still use the basis step and the recursive step

Recursive Definitions for Sets and Structures

- Sets can also be defined recursively
- We still use the basis step and the recursive step
  - Basis Step: initial collection of elements is specified

Recursive Definitions for Sets and Structures

- Sets can also be defined recursively
- We still use the basis step and the recursive step
    - Basis Step: initial collection of elements is specified
    - Recursive Step: rules for forming new elements in the set from those already known to be in the set are provided

Recursive Definitions for Sets and Structures

- Sets can also be defined recursively
- We still use the basis step and the recursive step
  - Basis Step: initial collection of elements is specified
  - Recursive Step: rules for forming new elements in the set from those already known to be in the set are provided
  - (Optional) Exclusion Rule: Specifies that a recursively defined set contains nothing other than those elements specified in the basis step or generated by applications of the recursive step

Proving these things

To prove results about recursively defined sets, we use what is called Structural Induction

Proving these things

To prove results about recursively defined sets, we use what is called Structural Induction

Form of Structural Induction:

- Base Case: Show that the result holds for all elements specified in the basis step of the recursive definition

Proving these things

To prove results about recursively defined sets, we use what is called Structural Induction

Form of Structural Induction:

- Base Case: Show that the result holds for all elements specified in the basis step of the recursive definition
- Inductive Hypothesis: Assume that for some element in the set, when we apply the recursive definition, we stay in the set

Proving these things

To prove results about recursively defined sets, we use what is called Structural Induction

Form of Structural Induction:

- Base Case: Show that the result holds for all elements specified in the basis step of the recursive definition
- Inductive Hypothesis: Assume that for some element in the set, when we apply the recursive definition, we stay in the set
- Inductive Step: Show that if the statement is true for each of the elements used to construct new elements in the recursive step of the definition, the result holds for the new elements

Back to Fibonacci

$$f_n = \begin{cases} 0 & n = 0 \\ 1 & n = 1 \\ f_{n-1} + f_{n-2} & n \geq 2 \end{cases}$$

Prove $f_n = \frac{1}{\sqrt{5}}(\frac{1+\sqrt{5}}{2})^n - \frac{1}{\sqrt{5}}(\frac{1-\sqrt{5}}{2})^n$.

Fibonacci Induction Proof

Base Case: Let $n = 0$,

$$f_0 = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^0 - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^0$$

$$= \frac{1}{\sqrt{5}} - \frac{1}{\sqrt{5}} = 0$$

Fibonacci Induction Proof

Let $n = 1$,

$$f_1 = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^1 - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^1$$

$$\frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right) - \left(\frac{1-\sqrt{5}}{2}\right)\right)$$

$$\frac{1}{\sqrt{5}}\left(\frac{2\sqrt{5}}{2}\right) = 1$$

So our base cases holds.

Fibonacci Induction Proof

Inductive Hypothesis: Assume for some $n \geq 1$ that $f_n = \frac{1}{\sqrt{5}}(\frac{1+\sqrt{5}}{2})^n - \frac{1}{\sqrt{5}}(\frac{1-\sqrt{5}}{2})^n$.

Fibonacci Induction Proof

Inductive Step: Consider, $f_{n+1} = f_n + f_{n-1}$. By our inductive hypothesis we have

$$\frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n + \left(\frac{1+\sqrt{5}}{2}\right)^{n-1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n-1}\right]$$

Let $\alpha = \frac{1+\sqrt{5}}{2}$ and $\beta = \frac{1-\sqrt{5}}{2}$. So we have,

$$\frac{1}{\sqrt{5}}(\alpha^n - \beta^n + \alpha^{n-1} - \beta^{n-1})$$

$$= \frac{1}{\sqrt{5}}(\alpha^{n-1}(\alpha+1) - \beta^{n-1}(\beta+1))$$

Note that $\alpha^2 = 1 + \alpha$ and $\beta^2 = 1 + \beta$. This comes from the fact that α and β are roots of $x^2 - x - 1$. Now we have,

$$= \frac{1}{\sqrt{5}}(\alpha^{n-1}(\alpha^2) - \beta^{n-1}(\beta^2))$$

$$= \frac{1}{\sqrt{5}}(\alpha^{n+1} - \beta^{n+1})$$

Example 1

Consider the following:

$$a_n = \begin{cases} 3 & n = 0 \\ 5 & n = 1 \\ 3a_{n-1}a_{n-2} + 4 & n \geq 2 \end{cases}$$

Prove $\forall n,\ a_n^2 \equiv 1 \mod 8$.

Example 1

Base Case:

Let n=0. Then $a_0^2 = 3^2 \equiv 1 \mod 8$.

Let n=1. Then $a_1^2 = 5^2 \equiv 1 \mod 8$.

So our base cases holds.

Example 1

Base Case:

Let n=0. Then $a_0^2 = 3^2 \equiv 1 \mod 8$.

Let n=1. Then $a_1^2 = 5^2 \equiv 1 \mod 8$.

So our base cases holds.

Inductive Hypothesis: Assume for some $n \geq 1$ that $a_n^2 \equiv 1 \mod 8$.

Example 1

Inductive Step: Consider $a_{n+1}^2 = (3a_na_{n-1} + 4)^2$. Simplifying that we get,

$$
\begin{aligned}
a_{n+1}^2 &= (3a_na_{n-1} + 4)^2 \\
&= 9a_n^2a_{n-1}^2 + 24a_na_{n-1} + 16 \\
&\equiv 1a_n^2a_{n-1}^2 + 0a_na_{n-1} + 0 \mod 8 \\
&\equiv 1(1)(1) \mod 8 \text{ \textit{by the Inductive Hypothesis}} \\
&\equiv 1 \mod 8
\end{aligned}
$$

Thus, $\forall n$, $a_n^2 \equiv 1 \mod 8$.

Strings

- An alphabet Σ is a finite set of symbols
  - Binary alphabet = {0,1}

Strings

- An alphabet Σ is a finite set of symbols
  - Binary alphabet = {0,1}
- A string is a finite sequence of symbols from some alphabet
  - Ex: 0001101, 1010101

Strings

- An alphabet Σ is a finite set of symbols
  - Binary alphabet = {0,1}
- A string is a finite sequence of symbols from some alphabet
  - Ex: 0001101, 1010101
- ε is the empty string

Strings

- An alphabet Σ is a finite set of symbols
  - Binary alphabet = {0,1}
- A string is a finite sequence of symbols from some alphabet
  - Ex: 0001101, 1010101
- ε is the empty string
- A language is a set of strings over an alphabet

Example 2

Define a language as follows:

- $\epsilon \in \mathcal{L}$
- $\forall \sigma \in \mathcal{L},\ a\sigma a \in \mathcal{L}$
- $\forall \sigma \in \mathcal{L},\ \sigma b \sigma b \in \mathcal{L}$
- $\forall \sigma \in \mathcal{L},\ c\sigma c\sigma \in \mathcal{L}$

Prove that all strings in the language contain an even number of each character $(a, b, c \in \Sigma)$.

Example 2

Base case:

$\epsilon \in L$: ϵ has 0 a, b, c and 0 is even.

So our base case holds.

Example 2

Base case:

$\epsilon \in L$: ϵ has 0 a, b, c and 0 is even.

So our base case holds.

Inductive hypothesis: Assume for some $x \in L$, x has an even number of each character a, b, c.

Example 2

Base case:

$\epsilon \in L$: ϵ has 0 a, b, c and 0 is even.

So our base case holds.

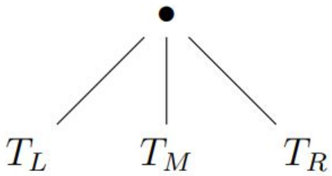
Inductive hypothesis: Assume for some $x \in L$, x has an even number of each character a, b, c.

Inductive step: Consider $i \in L$. Then $aia, ibib, cici \in L$. From our inductive hypothesis, i has an even number of a, b, c in it. Since $aia, ibib, cici$ all have 2 charecters for a, b, c, $aia, ibib, cici$ also have an even number of characters of a, b, c in them.

Example 3

Suppose we define a three-tree recursively as follows:

- A single node • is a three-tree
- If T_L, T_M, T_R are three-trees, then



  is a three-tree

Denote $N(T)$ = the number of nodes in the three-tree T. Define $h(T)$ = the height of the three-tree T recursively as:

- 0 if T = a single node
- $1 + \max\{T_L, T_M, T_R\}$ if T = a node with three children T_L, T_M, T_R

Prove that $N(T) \leq \frac{3^{h(T)+1}-1}{2}$ for all three-trees T. Further prove that $3^{h(T)}+1 \leq N(T)$. $\geq$

Base case: Consider a 1 node. Notice $N(T) = 1$ and $h(T) = 0$

$$\frac{3^{0+1} - 1}{2} = \frac{2}{2} = 1$$

$$N(T) = 1 \leq 1$$

$$3^0 + 1 = 2$$

$$2 \geq N(T) = 1$$

So our base cases hold.

Inductive hypothesis: Assume that for some three tree T, $N(T) \leq \frac{3^{h(T)+1}-1}{2}$ and $3^{h(T)} + 1 \geq N(T)$

Inductive step: Consider T'. $h(T') = 1 + max\{T'_L, T'_M, T'_R\}$.

$$\frac{3^{h(T')+1} - 1}{2}$$

$$\frac{3^{1+max\{T'_L, T'_M, T'_R\}+1} - 1}{2}$$

$$\frac{(3)3^{max\{T'_L, T'_M, T'_R\}+1} - 3 + 2}{2}$$

$$\frac{3(3^{max\{T'_L, T'_M, T'_R\}+1} - 1)}{2} + 1$$

From our inductive hypothesis,

$$\frac{3(3^{max\{T'_L, T'_M, T'_R\}+1} - 1)}{2} + 1 \geq 3N(max\{T'_L, T'_M, T'_R\}) + 1$$

$$\frac{3(3^{max\{T'_L, T'_M, T'_R\}+1} - 1)}{2} + 1 \geq N(T')$$

Consider T'. $h(T') = 1 + max\{T'_L, T'_M, T'_R\}$.

$$3^{h(T')} + 1$$

$$3^{1+max\{T'_L, T'_M, T'_R\}} + 1$$

$$3(3^{1+max\{T'_L, T'_M, T'_R\}}) + 1$$

By our inductive hypothesis,

$$3(3^{max\{T'_L, T'_M, T'_R\}}) + 1 \geq 3N(max\{T'_L, T'_M, T'_R\}) + 1$$

$$3(3^{max\{T'_L, T'_M, T'_R\}}) + 1 \geq N(T')$$