https://goo.gl/forms/1o1KOd17RMoURxjn2

CSE 20 DISCRETE MATH

Fall 2017

http://cseweb.ucsd.edu/classes/fa17/cse20-ab/

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 **N** to **R** using recursive definitions

Mathematical induction

Rosen p. 311

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),…

Mathematical induction*

Rosen p. 311

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

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 | { 1 } | = 1 | { 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

Recursive definitions, part 1

Rosen Sec 5.3

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

Rosen Sec 5.3

For functions f: **N** → X

- define by a (closed-form) formula

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

1. domain – **N**
2. codomain – X
3. assignment / rule – formula, table of values, induction

Recursive definitions, part 1

Rosen Sec 5.3

Sequences are functions too!

For functions $f: \mathbf{N} \rightarrow X$

- define by a (closed-form) formula

$f(0)$	$f(1)$	$f(2)$	$f(3)$	$f(4)$	…
a_0	a_1	a_2	a_3	a_4	…

1. domain – **N**
2. codomain – X
3. assignment / rule – formula, table of values, induction

Recursive definitions, part 1

Rosen Sec 5.3

For functions f: **N** → 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! $= n(n-1)(n-2)\cdots 3\cdot 2\cdot 1$

B. 2^n

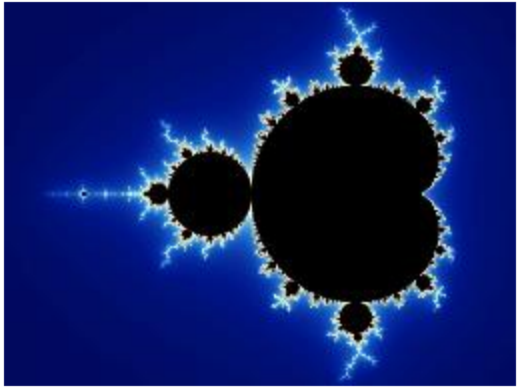
C. 2, -8, 32, -128, 512, …

D. $\sum_{i=1}^{n}(i^2+i)$

E. All of the above.

Recursive definitions, part 2

Rosen Sec 5.3



Basis Step Recursive Step

	Basis Step	Recursive Step	
* **n!**	$0! = 1,$	$n! = n\,(n-1)!$	for $n > 0$.
* $\mathbf{2^n}$	$2^0 = 1,$	$2^n = 2\,(2^{n-1})$	for $n > 0$.

2, -8, 32, -128, 512, …

$a_0 = 2,$	$a_n = \underline{-4}\, a_{n-1}$	for $n > 0$.

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

$$\sum_{i=1}^{0} g(i) = 0 \quad , \sum_{i=1}^{1} g(i) = g(1) \quad \text{, and} \quad \sum_{i=1}^{n} g(i) = g(n) + \sum_{i=1}^{n-1} g(i) \quad \text{for } n>1.$$

Extra mathematical induction practice

1

$n^2 < n!$ for all n greater than or equal to …

aka n^2 is $O(n!)$

For all $n \geq 1$

Video

2

$$\underbrace{\sum_{i=1}^{n} i = \frac{n(n+1)}{2}}_{\text{closed form}}$$

Pf By MI

Basis Step WTS at (n=1)

$\Sigma = \underline{\hspace{3em}}$

Ind Step ....

For all $n \geq 1$

3

$$\underbrace{\sum_{i=1}^{n} \frac{1}{i(i+1)} = \frac{n}{n+1}}_{\text{closed form}}$$

Pf: By M.I.

Basis Step WTS (n=1)

$\sum_{i=1}^{1} \frac{1}{i(i+1)} = \frac{1}{1+1}$

LHS $= \frac{1}{1(1+1)} = \frac{1}{2}$ RHS $= \frac{1}{2}$ ✓

Recursive definitions, part 2

Rosen Sec 5.3

Define a set S by

ex: $\mathbb{N} = \{0, 1, 2, 3, 4, \ldots\}$

{ … }

{ x | P(x) }

Recursive definition:

Basis step – Specify initial collection of elements. 0

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

if $n \in \mathbb{N}$ then $n+1 \in \mathbb{N}$

$S =$

Example: { x | x is an int, x > 3}

Basis Step $4 \in S$

Rec Step If $n \in S$, $n+1 \in S$

Recursive definitions, part 2

Rosen Sec 5.3

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

Basis step: $1 \in S$

Recursive step: If $a \in S$, then $a + 2 \in S$.

$\{1, 3, 5, 7, \ldots\}$

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.

Recursive definitions, part 2

Rosen Sec 5.3 page 349

S

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

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

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

$w=\lambda$ $w0$ $\quad$ $w=\lambda$ $w1$ $\quad$ $w=0$ $w0$ $\quad$ $w=0$ $w1$

$\{ \lambda, 0, 1, 00, 01,$

$w=1$ $w0$ $\quad$ $w=1$ $w1$

$10, 11, \ldots \}$

Also known as **binary sequences**

Recursive definitions, part 2

Rosen Sec 5.3 page 349

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

Basis step: $\lambda \in \{0,1\}^*$ where λ 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. λ

B. 1 $= \lambda 1$ so is result of applying to basis step

C. 000 $= (00)0 = (0 \cdot 0)0 = ((\lambda 0)0)0$

D. 010101

E. 1011001110001111000011111000000111…

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

Rosen p. 354

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.

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 λ 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 *Rosen Sec 5.3*

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

Basis step: $\lambda \in S$ where λ 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 WTS λ has an equal # of 0s & 1s

Pf: λ has 0 0s and 0 1s, so equal #

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 λ 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** → 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

Recursive definitions, part 3

Rosen Sec 5.3

Define a function f: **S** → 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.

λ

$w0$, $w1$

Example: how do you define the length function l: {0,1}* → N ?

Basis Step $\quad l(\lambda) = 0$

Recursive Step $\quad$ If $w \in \{0,1\}^*$, $l(w0) = l(w) + 1$

$l(w1) = l(w) + 1$

Reminders

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

Next class: more applications + variants of induction