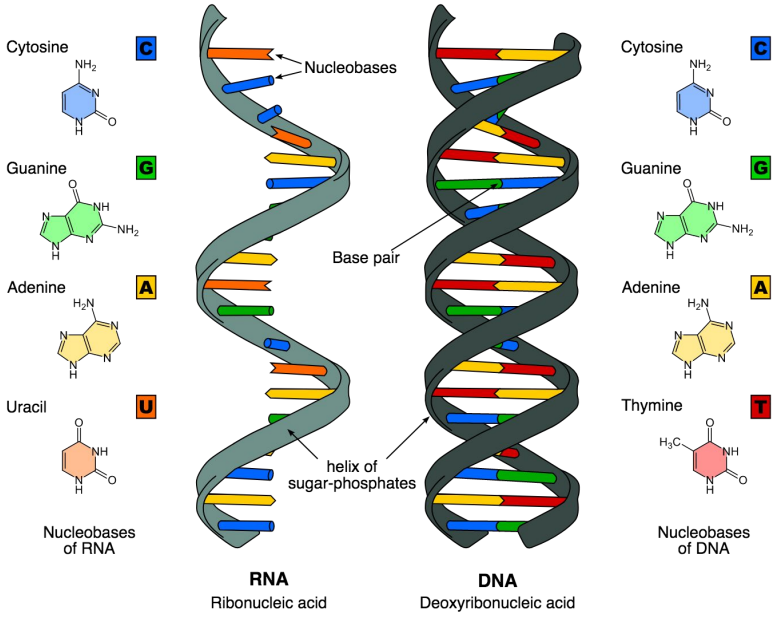
CS 40
FOUNDATIONS OF CS

Summer 2024
Week 3
Slides B

Wednesday's learning goals

- Practice with properties of recursively defined sets and functions
- Prove and disprove properties of recursively defined sets and functions with **structural induction**
- Define linked lists: a recursively defined data structure

RNA strands as strings



Each RNA strand is a **string** whose symbols are elements of the set $B = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}$.

Definition The set of RNA strands S is defined (recursively) by:

Basis Step: $\mathtt{A} \in S, \mathtt{C} \in S, \mathtt{U} \in S, \mathtt{G} \in S$

Recursive Step: If $s \in S$ and $b \in B$, then $sb \in S$

Defining functions recursively

when domain is

we use strand as string alternately

recursively defined

The function *rnalen* that computes the length of RNA strands in S is defined by:

$$rnalen : S \to \mathbb{Z}^+$$

Basis Step:	If $b \in B$ then	$rnalen(b) = 1$
Recursive Step:	If $s \in S$ and $b \in B$, then	$rnalen(sb) = 1 + rnalen(s)$

The function *basecount* that computes the number of a given base b appearing in a RNA strand s is defined recursively:

$$basecount : S \times B \to \mathbb{N}$$

Basis Step: If $b_1 \in B, b_2 \in B$

$$basecount(b_1, b_2) = \begin{cases} 1 & \text{when } b_1 = b_2 \\ 0 & \text{when } b_1 \neq b_2 \end{cases}$$

Recursive Step: If $s \in S, b_1 \in B, b_2 \in B$

$$basecount(sb_1, b_2) = \begin{cases} 1 + basecount(s, b_2) & \text{when } b_1 = b_2 \\ basecount(s, b_2) & \text{when } b_1 \neq b_2 \end{cases}$$

The function *rnalen* that computes the length of RNA strands in S is defined by:

$$rnalen : S \to \mathbb{Z}^+$$

Basis Step:	If $b \in B$ then	$rnalen(b) = 1$
Recursive Step:	If $s \in S$ and $b \in B$, then	$rnalen(sb) = 1 + rnalen(s)$

The function *basecount* that computes the number of a given base b appearing in a RNA strand s is defined recursively:

$$basecount : S \times B \to \mathbb{N}$$

Basis Step: If $b_1 \in B, b_2 \in B$

$$basecount(b_1, b_2) = \begin{cases} 1 & \text{when } b_1 = b_2 \\ 0 & \text{when } b_1 \neq b_2 \end{cases}$$

Recursive Step: If $s \in S, b_1 \in B, b_2 \in B$

$$basecount(sb_1, b_2) = \begin{cases} 1 + basecount(s, b_2) & \text{when } b_1 = b_2 \\ basecount(s, b_2) & \text{when } b_1 \neq b_2 \end{cases}$$

$\leq basecount(s) + 1$

Which of the following statements is true?

A. $\forall s \in S\,(rnalen(s) < basecount(s, \mathtt{A}))$

B. $\forall s \in S\,(rnalen(s) \leq basecount(s, \mathtt{A}))$

C. $\forall s \in S\,(rnalen(s) > basecount(s, \mathtt{A}))$

D. $\forall s \in S\,(rnalen(s) \geq basecount(s, \mathtt{A}))$ ✓

E. None of the above

Definition The set of RNA strands S is defined (recursively) by:

Basis Step:	$\mathtt{A} \in S, \mathtt{C} \in S, \mathtt{U} \in S, \mathtt{G} \in S$
Recursive Step:	If $s \in S$ and $b \in B$, then $sb \in S$

✓

Prove or disprove $\forall s \in S\,(rnalen(s) \geq basecount(s, \mathtt{A}))$:

Prove this by induction on S

→ Base set

Base case: Prove $\forall s \in B$ $rnalen(s) \geq basecount(s, A)$

Using definition of rnalen, basecount,

$rnalen(s) = 1$ $\quad basecount(s, A) \leq 1 = rnalen(s)$

$\therefore$ $rnalen(s) \geq basecount(s, A)$ ✱

Inductive case: Prove for $s' = sb$ where $s \in S$, $b \in B$

Assume the claim is true for s (here s is a smaller string than s')

Using defn of rnalen, basecount

$$rnalen(s') = rnalen(s) + 1 \quad [\because s' = sb]$$

$B \to$ strings of len 1
$S_2 \to$ " len 2
$S_3 \to$ " len 3
⋮
$s \in S_l \to$ " len l

By induction

$$rnalen(s) \geq basecount(s, A)$$

$$rnalen(s') = rnalen(s) + 1$$
$$\geq basecount(s, A) + 1 \quad [\text{by defn of basecount}]$$
$$\geq basecount(s', A)$$

Thus, $\boxed{rnalen(s') \geq basecount(s', A)}$ ✦

This completes the proof of the inductive case

∴ We have proved our claim by induction

To prove a universal quantification where the element comes from a recursively defined set, consider an arbitrary element and prove two cases:

1. Assume the element is one of those from the basis step and prove the conclusion

2. Assume the element is one of those built during the recursive step, **and assume that the property holds for the elements used to build it**, and prove the conclusion.

To prove a universal quantification where the element comes from a recursively defined set, consider an arbitrary element and prove two cases:

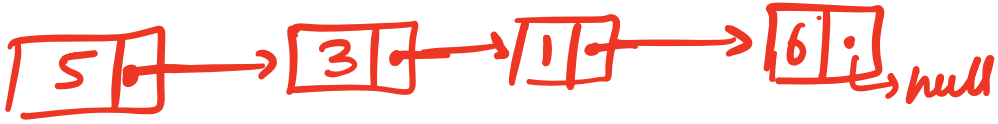
1. Assume the element is one of those from the basis step and prove the conclusion

2. Assume the element is one of those built during the recursive step, **and assume that the property holds for the elements used to build it**, and prove the conclusion.

Definition The set of linked lists of natural numbers L is defined by:

Basis Step: $[] \in L$

Recursive Step: If $l \in L$ and $n \in \mathbb{N}$, then $(n, l) \in L$



→ construct this LL as a set L

$[] \in L$

$(6, [])$ $(1, (6, []))$ $(3, (1, (6, [])))$

$(5, (3, (1, (6, []))))$

Definition The set of linked lists of natural numbers L is defined by:

Basis Step: $[] \in L$

Recursive Step: If $l \in L$ and $n \in \mathbb{N}$, then $(n, l) \in L$

Which of the following is a linked list of natural numbers?

A: ( )
B: (3, 7, 9)
C: (9, (3, 7))
D: ([])
E: None of the above

Definition The length of a linked list of natural numbers L, $len : L \to \mathbb{N}$ is defined by:

Basis Step:		$length([]) = 0$
Recursive Step:	If $l \in L$ and $n \in \mathbb{N}$, then	$length((n, l)) = ?$

$1 + length(l)$

How should we fill in the recursive step of the rule?

A: $n + 1$
B: $n + l$
C: $n + length(l)$
D: $1 + length(l)$ ✓
E: None of the above

Definition The function $append : L \times \mathbb{N} \to L$ that adds an element at the end of a linked list is defined:

Basis Step: If $m \in \mathbb{N}$ then $append([], m) = (m, [])$

Recursive Step: If $l \in L$ and $n \in \mathbb{N}$ and $m \in \mathbb{N}$, then $append((n, l), m) = (n, append(l, m))$

empty set

$L \quad \mathbb{N}$

$L \quad \mathbb{N}$

✓ $append([], 1) = (1, [])$

✓ $append((1, (2, [])), 3) = (1, (2, (3, [])))$

How should we fill in the basis step of the rule?

A: $append(m) = []$
B: $append([]) = m$
C: $append([], m) = ([], m)$
✓ D: $append([], m) = (m, [])$
E: None of the above

$(n, (1, (2, [])))$

we want

$(n, (1, (2, (m, []))))$

Definition The function $append : L \times \mathbb{N} \to L$ that adds an element at the end of a linked list is defined:

Basis Step: If $m \in \mathbb{N}$ then $append([], m) = (m, [])$

Recursive Step: If $l \in L$ and $n \in \mathbb{N}$ and $m \in \mathbb{N}$, then _____

How should we fill in the recursive step of the rule?

A: $append(l, m) = (m, l)$
B: $append(l, m) = (l, m)$
C: $append((n, l), m) = (m, (n, l))$
D: $append((n, l), m) = ((n, l), m)$
E: None of the above

Definition The set of linked lists of natural numbers L is defined by:

Basis Step:	$[] \in L$
Recursive Step:	If $l \in L$ and $n \in \mathbb{N}$, then $(n, l) \in L$

Claim: $\forall l \in L\,(\ length(\ append(l, 100)\ ) > length(l)\ )$

Proof by induction on l → structural

Base case: $(l = [])$

length (append ([], 100)) > length ([])

use definition of append & length

length ((100, [])) = 1 + length ([])

1 > 0

= 1 + 0

Thus the base case is true

Induction: Let $l = (n, l')$ → induction hypothesis

Assume by induction that:

$length(append(l', 100)) > length(l')$

T.S.T

$length(append(l, 100)) > length(l)$

we use defn of append, length

$append(l, 100) = (n, append(l', 100))$

$\therefore length(\ ,\ ) = 1 + length(\ ,\ )$ → by induction

by defn

$> 1 + length(l') = length(l)$

$\therefore length(append(l, 100)) > length(l)$

Thus, the inductive case is true

$\therefore$ The claim holds by induction

Claim: $\forall l \in L\,(\ length(\ append(l, 100)\ ) > length(l)\ )$

Analogy: unit tests in programming

To prove a universal quantification where the element comes from a recursively defined set, consider an arbitrary element and prove two cases:

1. Assume the element is one of those from the basis step and prove the conclusion

2. Assume the element is one of those built during the recursive step, **and assume that the property holds for the elements used to build it**, and prove the conclusion.

Claim: $\forall l \in L\,(\ length(\ append(l, 100)\ ) > length(l)\ )$

Analogy: unit tests in programming

Basis Step

1. **To Show** $length(\ append([], 100)\ ) > length([])$	Because [] is the only element defined in the basis step of L, we only need to prove that the property holds for [].
2. **To Show** $length(\ (100, [])\ ) > length([])$	By basis step in definition of *append*.
3. **To Show** $(1 + length([])) > length([])$	By recursive step in definition of *length*.
4. **To Show** $1 + 0 > 0$	By basis step in definition of *length*.
5. **To Show** T	By properties of integers
QED	Because we got to T only by rewriting **To Show** to equivalent statements, using well-defined proof techniques, and applying definitions.

Claim: $\forall l \in L\,(\ length(\ append(l, 100)\ ) > length(l)\ )$

Analogy: unit tests in programming

Recursive Step

Consider an arbitrary: $l = (n, l')$, $l' \in L$, $n \in \mathbb{N}$, and we assume as the **induction hypothesis** that:

$$length(\ append(l', 100)\ ) > length(l')$$

Our goal is to show that $length(\ append((n, l'), 100)\ ) > length((n, l'))$ is true. We evaluate each side of the candidate inequality. Applying the recursive definition of *append*,

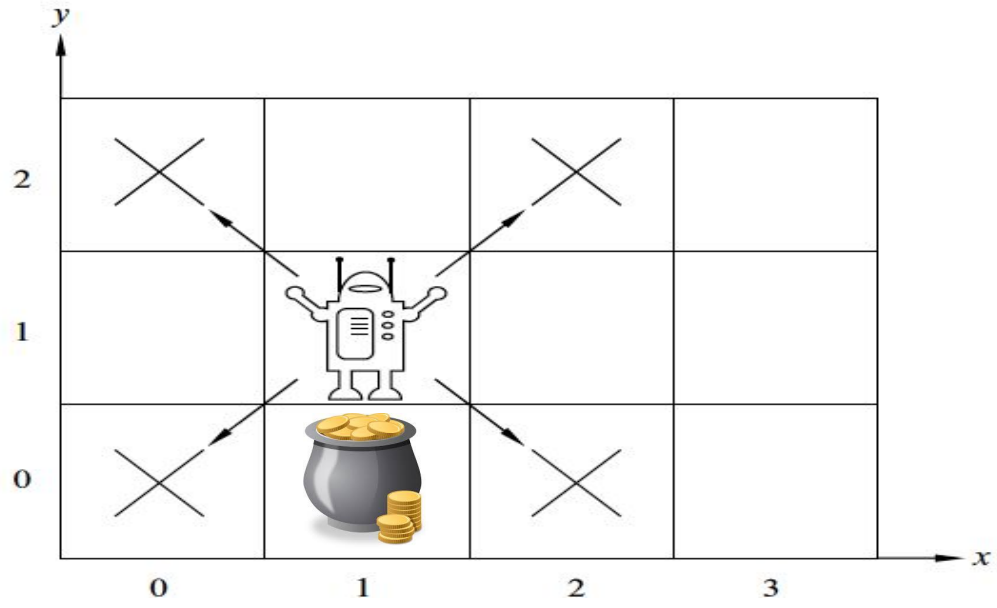
$$\begin{aligned}
LHS &= length(\ append((n, l'), 100)\ ) = length(\ (n, append(l', 100))\ ) && \text{by the recursive definition of } append \\
&= 1 + length(\ append(l', 100)\ ) && \text{by the recursive definition of } length \\
&> 1 + length(l') && \text{by the induction hypothesis} \\
&= length((n, l')) && \text{by the recursive definition of } length \\
&= RHS
\end{aligned}$$

Thursday's learning goals

- Practice proof by structural induction in a new context (robot navigation)
- Recursive definition of natural numbers
- Proof by mathematical induction

Robot

Start at origin, moves on infinite 2-dimensional integer grid.
At each step, move to diagonally adjacent grid point.



Can it ever reach (1,0)?

A. Yes
B. No
C. I'd have to think about it more
D. I don't know

* Terminology: lemmas and theorems are defined in the handout, page 6

Robot

Theorem: Robot can never reach (1, 0)

Lemma*: The sum of the coordinates of any position reachable by the robot is even, i.e. has remainder 0 upon division by 2.

Theorem ↑ Lemma ↑ claims, observations, corollary ...

Main claim you want to prove

Smaller results helpful to prove theorem

Let R be the set of points reachable by the robot

Lemma: simpler

$\forall (x,y) \in \mathbb{Z} \times \mathbb{Z}$

$(x,y) \in R \rightarrow x+y$ is even

$\forall (x,y) \in R$ $x+y$ is even

Theorem: Robot cannot reach $(1,0)$

Lemma: $\forall (x,y) \in R \quad x+y$ is even

How do we prove the Theorem assuming the Lemma?

Proof of Theorem (assuming Lemma):

We know that:

$\forall (x,y) \in \mathbb{Z} \times \mathbb{Z} \quad (x,y) \in R \longrightarrow x+y$ is even (from our Lemma)

If $(1,0) \in R \longrightarrow 1+0$ is even (by instantiation over universal quantifier)

$A \longrightarrow B$

$1+0$ is odd (fact)

$\sim B$

Modus Tollens

$\sim A: (1,0) \notin R$

Therefore proved that $(1,0)$ is not reachable by the robot.

assuming Lemma.

Lemma: $\forall (x,y) \in R \quad x+y$ is even

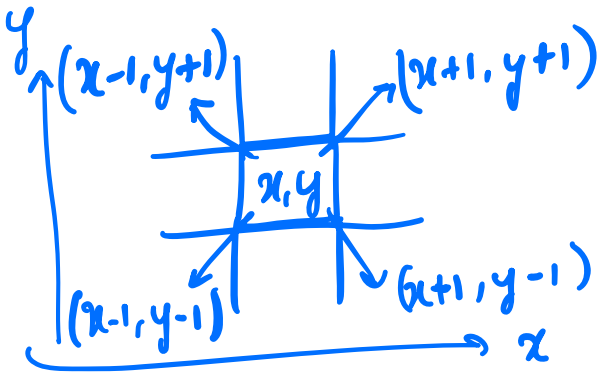
Proof of Lemma:

Proof by structural induction of R:

Recursive Definition of R:

Basis step: $(0,0) \in R$

Recursive step: $\forall (x,y) \in R$



Then,

$(x-1, y+1) \in R$

$(x+1, y+1) \in R$

$(x-1, y-1) \in R$

$(x+1, y-1) \in R$

Base case: $(0,0) \in R$

$0+0$ is even

Therefore base case is true.

Inductive case:

Need to prove for a pt $(x', y') \neq (0,0)$ $\in R$

Let (x', y') be added to R in the recursive construction due to the pt (x, y) where:

Case (i) $(x', y') = (x-1, y+1)$

(ii) $= (x+1, y+1)$

(iii) $= (x+1, y-1)$

(iv) $= (x+1, y-1)$

Assume by inductive hypothesis that the claim is true for (x, y), (i.e) $(x, y) \in R \wedge x+y$ is even

$\therefore \quad x+y = 2c$ for some $c \in \mathbb{Z}$

Now,

(i) $x-1+y+1 = x+y$

$= 2c$

(ii) $x+1+y+1 = x+y+2$

$= 2(c+1)$

(iii) $x-1+y-1 = x+y-2$

$= 2(c-1)$

(iv) $x+1+y-1 = x+y$

$= 2c$

In all cases $x'+y'$ is divisible by 2

$\therefore x'+y'$ is even

Thus,

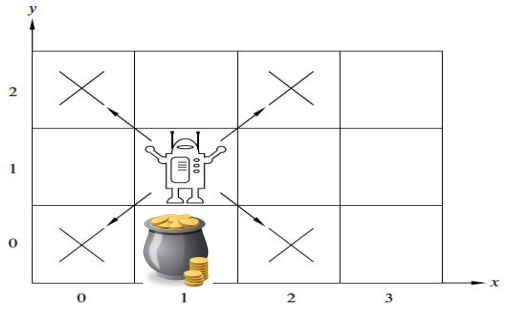
If $(x'+y') \in R$

$\longrightarrow x'+y'$ is even

This completes the proof of our inductive step

This completes proof of our Lemma

Robot



Proof of theorem using lemma: *To show is* $(1,0) \notin P$.

Rewriting the lemma to explicitly restrict the domain of the universal, we have
$\forall(x,y)\ (\ (x,y) \in P \rightarrow (x+y \text{ is an even integer})\ )$

Rewriting the lemman once more to its logically equivalent contrapositive form, we get
$\forall(x,y)\ (\ (x+y \text{ is an odd integer}) \rightarrow (x,y) \notin P\ )$ $\ldots(a)$

By universal instantiation on statement (a) *for domain element* $(1,0)$ *we get*
$(1+0 \text{ is an odd integer}) \rightarrow (1,0) \notin P$ $\ldots(b)$

Further, $1+0$ *evaluates to 1, which is an odd integer* $\ldots(c)$

Therefore, we can conclude that $(1,0) \notin P$ *by Modus Ponens on statements* (b) *and* (c) ■

* Terminology: lemmas and theorems are defined in the handout, page 6

Robot

Lemma*: The sum of the coordinates of any position reachable by the robot is even, i.e. has remainder 0 upon division by 2.

Proof of Lemma: by structural induction on the set of possible positions of the robot.

Definition The set of positions the robot can visit P is defined by:

Basis Step: $(0, 0) \in P$

Recursive Step: If $(x, y) \in P$, then  are also in P

* Terminology: lemmas and theorems are defined in the handout, page 6

Robot

Lemma*: The sum of the coordinates of any position reachable by the robot is even, i.e. has remainder 0 upon division by 2.

Proof of Lemma: by structural induction on the set of possible positions of the robot.

Basis step: To show is 0+0 is an even integer.

* Terminology: lemmas and theorems are defined in the handout, page 6

Robot

Lemma*: The sum of the coordinates of any position reachable by the robot is even, i.e. has remainder 0 upon division by 2.

Proof of Lemma: by structural induction on the set of possible positions of the robot.

Recursive step: Consider arbitrary (x,y) in P. To show is

(x+y is an even integer) □ (sum of coordinates of next position is even integer)

Recursive definitions: recap

We have recursive definitions of

- the set of RNA strands
- the set of linked lists
- the set of positions visited by a robot

What else?

Set of nonnegative integers are recursively defined

Definition The set of natural numbers (aka nonnegative integers), $\mathbb{N}$, is defined (recursively) by:

Basis Step:	$0 \in \mathbb{N}$
Recursive Step:	If $n \in \mathbb{N}$ then $n+1 \in \mathbb{N}$ (where $n+1$ is integer addition)

How do we do induction for robot qs instead of structural induction

When you define R

Base set: $R_0 = \{(0,0)\}$

Recursive set: construct R_{i+1},

$\forall (x,y) \in R_i$ add the 4 corresponding pts to R_{i+1}

$R = \bigcup_{j=0}^{n} R_j$

while proving Lemma by induction on $i \in [n]$

n – # recursive construction step for R

Base ($i=0$):

$(x,y) \in R_0$

(ie) $(0,0) \in R_0$

this claim is true

Inductive case: for arbitrary $i+1$,

Assume by induction that claim is true for i

(ie) $\forall (x,y) \in R_i$ $x+y$ is even

To prove: $\forall (x,y) \in R_{i+1}$ $x+y$ is even

Let (x',y') be an arbitrary element in R_{i+1}

& let $(x,y) \in R_i$ be the one used to add (x',y') to R_{i+1}

4 cases:

.

.

.

Mathematical induction

To prove a universal quantification where the element comes from the set of integers >= 0, prove two cases:

1. Prove the property is true about the number n = 0

2. Consider an arbitrary integer n greater than or equal to 0, assume (as the **induction hypothesis**) that the property holds for n, and use this and other facts to prove that the property holds for n+1.

Structural induction

To prove a universal quantification where the element comes from a recursively defined set, prove two cases:

1. Assume the element is one of those from the basis step and prove the conclusion

2. Assume the element is one of those from the recursive step, and assume that the property holds for the elements used to build it, and prove the conclusion.

Proof by mathematical induction

Climbing an infinite ladder

Recall that the set of linked lists of natural numbers L	Recall that length of a linked list of natural numbers L, $length : L \to \mathbb{N}$ is defined by:
Basis Step: $[] \in L$ Recursive Step: If $l \in L$ and $n \in \mathbb{N}$ then $(n, l) \in L$	Basis step: $length([]) = 0$ Recursive step: If $l \in L$ and $n \in \mathbb{N}$ then $length((n, l)) = 1 + length(l)$

Prove or disprove: $\forall n \in \mathbb{N}\ \exists l \in L\ (\ length(l) = n\ )$

for each n there is some LL l s.t $len(l) = n$

Proof by induction on n

Base case ($n=0$): $l = [\,]$ has $length([\,]) = 0$

$\therefore$ the statement is true for $n=0$

Inductive case: Assume the claim is true for an arbitrary n

$$\exists l \in L \quad \text{length}(l) = n$$

We prove the claim for $n+1$:

$$\exists l \in L \quad \text{length}(l) = n+1$$

Let l be a list such that $\text{length}(l) = n$,
by our induction hypothesis,
such an l exists.

The list $l' = (100, l)$ has

$$\begin{aligned} \text{length}(l') &= 1 + \text{length}(l) \\ &= 1 + n \end{aligned}$$

$\therefore$ l' is a witness to our claim for $n+1$

Thus proving

$$\exists l \in L \quad \text{length}(l') = n+1$$

This proves the inductive case, proving our claim.

What witness should we pick?

A. []
B. I_w
C. $(1, I_w)$
D. (n, I_w)
E. None of the above

Sums of powers of 2

The function $sumPow$ with domain $\mathbb{N}$, codomain $\mathbb{N}$, and which computes, for input i, the sum of the first i powers of 2 is defined recursively by $sumPow : \mathbb{N} \to \mathbb{N}$ with

Basis step: $sumPow(0) = 1$.

Recursive step: If $x \in \mathbb{N}$ then $sumPow(x + 1) = sumPow(x) + 2^{x+1}$.

Prove or disprove $\forall n \in \mathbb{N}\,(sumPow(n) = 2^{n+1} - 1)$:

Prove or disprove $\forall n \in \mathbb{N}\,(sumPow(n) = 2^{n+1} - 1)$:

Fill in the blanks in the following proof of $\forall n \in \mathbb{N}\,(sumPow(n) = 2^{n+1} - 1)$*:*

Since the domain is the set of natural numbers, we proceed by _____.

Basis case *We need to show that* _____.

Evaluating each side:

$LHS = sumPow(0) = 1$ *by the basis case in the recursive definition of* $sumPow$*;*

$RHS = 2^{0+1} - 1 = 2^1 - 1 = 2 - 1 = 1.$

Since $1 = 1$*, the equality holds.*

Recursive step *Consider arbitrary natural number* n *and assume, as the* _____ *that* $sumPow(n) = 2^{n+1} - 1$*. We need to show that* _____.

Evaluating each side:

Mathematical induction

To prove a universal quantification where the element comes from the set of integers >= b, prove two cases:

1. Prove the property is true about the number b

2. Consider an arbitrary integer n greater than or equal to b, assume (as the **induction hypothesis**) that the property holds for n, and use this and other facts to prove that the property holds for n+1.

Structural induction

To prove a universal quantification where the element comes from a recursively defined set, prove two cases:

1. Assume the element is one of those from the basis step and prove the conclusion

2. Assume the element is one of those from the recursive step, and assume that the property holds for the elements used to build it, and prove the conclusion.

Growth of functions

Rosen p. 320

Definition The exponent function on $\mathbb{Z}^+$ is defined by:

Basis Step:	If $n = 1$ then	2^n	$= 2$
Recursive Step:	If $n \in \mathbb{Z}^+$, then	2^{n+1}	$= 2 \cdot 2^n$

Definition The factorial function on $\mathbb{Z}^+$ is defined by:

Basis Step:	If $n = 1$ then	$n!$	$= 1$
Recursive Step:	If $n \in \mathbb{Z}^+$, then	$(n+1)!$	$= (n+1) \cdot n!$

Which of the following is true?

A: $2^0 < 0!$
B: $2^1 < 1!$
C: $2^2 < 2!$
D: $2^3 < 3!$
E: $2^{30} < 30!$

Prove or disprove: $\forall n \in \mathbb{Z}^{\geq 4}\ (2^n < n!)$

Proof: By mathematical induction on $n \in \mathbb{Z}^{\geq 4}$

Basis step:

Inductive step: Consider and assume as **inductive hypothesis** that

Want to show (WTS)

Use mathematical induction to prove that $2^n < n!$ for every integer n with $n \geq 4$. (Note that this inequality is false for $n = 1$, 2, and 3.)

Solution: Let $P(n)$ be the proposition that $2^n < n!$.

BASIS STEP: To prove the inequality for $n \geq 4$ requires that the basis step be $P(4)$. Note that $P(4)$ is true, because $2^4 = 16 < 24 = 4!$

INDUCTIVE STEP: For the inductive step, we assume that $P(k)$ is true for an arbitrary integer k with $k \geq 4$. That is, we assume that $2^k < k!$ for the positive integer k with $k \geq 4$. We must show that under this hypothesis, $P(k+1)$ is also true. That is, we must show that if $2^k < k!$ for an arbitrary positive integer k where $k \geq 4$, then $2^{k+1} < (k+1)!$. We have

$$
\begin{aligned}
2^{k+1} &= 2 \cdot 2^k && \text{by definition of exponent} \\
&< 2 \cdot k! && \text{by the inductive hypothesis} \\
&< (k+1)k! && \text{because } 2 < k+1 \\
&= (k+1)! && \text{by definition of factorial function.}
\end{aligned}
$$

This shows that $P(k+1)$ is true when $P(k)$ is true. This completes the inductive step of the proof.

We have completed the basis step and the inductive step. Hence, by mathematical induction $P(n)$ is true for all integers n with $n \geq 4$. That is, we have proved that $2^n < n!$ is true for all integers n with $n \geq 4$. ◀