This week's highlights

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

Lecture videos

Week 6 Day 1 YouTube playlist
Week 6 Day 2 YouTube playlist
Week 6 Day 3 YouTube playlist

Monday February 8

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 $sb \in S$

where sb is string concatenation.

The function *rnalen* that computes the length of RNA strands in S is defined recursively 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 by $basecount : S \times B \to \mathbb{N}$

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

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

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

Proof:

Basis case: Assume $s = \mathtt{A} \lor s = \mathtt{C} \lor s = \mathtt{U} \lor s = \mathtt{G}$. Need to show $rnalen(s) \geq basecount(s, \mathtt{A})$.

Case 1: Want to show $(s = \mathtt{A}) \to (\ rnalen(s) \geq basecount(s, \mathtt{A})\ )$.

Case 2: Want to show $(s = \mathtt{C} \lor S = \mathtt{U} \lor S = \mathtt{G}) \to (\ rnalen(s) \geq basecount(s, \mathtt{A})\ )$.

Continued next page

Proof by universal generalization: To prove that $\forall x\, P(x)$ is true, we can take an arbitrary element e from the domain and show that $P(e)$ is true, without making any assumptions about e other than that it comes from the domain.

New! Proof by Structural Induction (Rosen 5.3 p354) To prove a universal quantification over a recursively defined set:

> **Basis Step**: Show the statement holds for elements specified in the basis step of the definition.
>
> **Recursive 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 these new elements.

Recursive case: Want to show

$$\forall e \in S \; (\; \mathit{rnalen(e)} \geq \mathit{basecount}(e, \texttt{A}) \rightarrow \forall b \in B \; (\; \mathit{rnalen(eb)} \geq \mathit{basecount}(eb, \texttt{A}) \;) \;)$$

Consider arbitrary e. Assume, as the **induction hypothesis** that

$$\mathit{rnalen(e)} \geq \mathit{basecount}(e, \texttt{A})$$

Need to show

$$\forall b \in B \; (\; \mathit{rnalen(eb)} \geq \mathit{basecount}(eb, \texttt{A}) \;)$$

Consider arbitrary $b \in B$.

Case 1: Want to show $(b = \texttt{A}) \rightarrow (\; \mathit{rnalen(eb)} \geq \mathit{basecount}(eb, \texttt{A}) \;)$.

Case 2: Want to show $(b = \texttt{C} \vee b = \texttt{U} \vee b = \texttt{G}) \rightarrow (\; \mathit{rnalen(eb)} \geq \mathit{basecount}(eb, \texttt{A}) \;)$.

Wednesday February 10

Proof by Structural Induction (Rosen 5.3 p354) To prove a universal quantification over a recursively defined set:

> **Basis Step**: Show the statement holds for elements specified in the basis step of the definition.
>
> **Recursive 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 these new elements.

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)

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 $sumPow(x+1) = sumPow(x) + 2^{x+1}$.

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

Since $\mathbb{N}$ is recursively defined, 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:

$$LHS = sumPow(n+1) \overset{\text{rec def}}{=} sumPow(n) + 2^{n+1} \overset{\text{IH}}{=} (2^{n+1} - 1) + 2^{n+1}.$$

$$RHS = 2^{(n+1)+1} - 1 \overset{\text{exponent rules}}{=} 2 \cdot 2^{n+1} - 1 = \left(2^{n+1} + 2^{n+1}\right) - 1 \overset{\text{regrouping}}{=} (2^{n+1} - 1) + 2^{n+1}$$

Thus, $LHS = RHS$. The structural induction is complete and we have proved the universal generalization.

Extra example Connect the function $sumPow$ to binary expansions of positive integers.

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

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

Examples:

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

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

Examples:

Extra example: The function $prepend : L \times \mathbb{N} \to L$ that adds an element at the front of a linked list is defined:

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

Examples:

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

Proof: By structural induction on L, we have two cases:

Basis Step

1.	**To Show** $length(\ append([], 100)\ ) > length([])$	Because [] is the only element defined i step of L, we only need to prove that th 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 *lengt*
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 equivalent statements, using well-de techniques, and applying definitions.

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 also true. We evaluate each side of the candidate inequality:

$LHS = length(\ append((n, l'), 100)\ ) = length(\ (n, append(l', 100))\ )$ by the recursive definitio

$= 1 + length(\ append(l', 100)\ )$ by the recursive definition of *length*

$> 1 + length(l')$ by the induction hypothesis

$= length((n, l'))$ by the recursive definition of *length*

$= RHS$

Friday February 12

Invariant: A property that is true about our algorithm no matter what. Rosen p375

Theorem: Statement that can be shown to be true, usually an important one. Rosen p81

> Less important theorems can be called **proposition**, **fact**, **result**.
>
> A less important theorem that is useful in proving a theorem is called a **lemma**.
>
> A theorem that can be proved directly after another one has been proved is called a **corollary**

Theorem: A robot on an infinite 2-dimensional integer grid starts at $(0,0)$ and at each step moves to diagonally adjacent grid point. This robot can / cannot (*circle one*) reach $(1,0)$.

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

Lemma: $\forall (x,y) \in P ((x+y$ is an even integer$)\ )$

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 $\forall (x,y)\ (\ (x,y) \in P \to (x+y$ is an even integer$)\ )$. Since the universal is true, $(\ (1,0) \in P \to (1+0$ is an even integer$)\ )$ is a true statement. Evaluating the conclusion of this conditional statement: By definition of long division, since $1 = 0 \cdot 2 + 1$ (where $0 \in \mathbb{Z}$ and $1 \in \mathbb{Z}$ and $0 \leq 1 < 2$ mean that 0 is the quotient and 1 is the remainder), $1 \textbf{ mod } 2 = 1$ which is not 0 so the conclusion is false. A true conditional with a false conclusion must have a false hypothesis. Thus, $(1,0) \notin P$, QED. □

Proof of lemma by structural induction:

Basis Step

Recursive Step. Consider arbitrary $(x,y) \in P$. To show is:

$(x+y$ is an even integer$) \to ($sum of coordinates of next position is even integer$)$

Assume **as the induction hypothesis, IH** that:

"New"! Proof by Mathematical Induction (Rosen 5.1 p329)
To prove a universal quantification over the set of all integers greater than or equals some base integer b:

> **Basis Step**: Show the statement holds for b.
>
> **Recursive Step**: 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 $n+1$.

Recall that the set of linked lists of natural numbers L

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

Recall that length of a linked list of natural n
bers L, $length : L \to \mathbb{N}$ is defined by:

> Basis step: $length([]) = 0$
> Recursive step: If $l \in L$ and $n \in \mathbb{N}$ t
> $length((n, l)) = 1 + length(l)$

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

Review quiz questions

1. **Monday** The function *rnalen* that computes the length of RNA strands in S is defined recursively 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 by $basecount : S \times B \to \mathbb{N}$

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

   (a) Select all and only options that give a witness for the existential quantification

   $$\exists s \in S \; (\; rnalen(s) = basecount(s, \mathtt{U}) \;)$$

      i. `A`
      ii. `UU`
      iii. `CU`
      iv. $(\mathtt{U}, 1)$

   (b) Select all and only options that give a counterexample for the universal quantification

   $$\forall s \in S \; (\; rnalen(s) > basecount(s, \mathtt{G}) \;)$$

      i. `U`
      ii. `GG`
      iii. `AG`
      iv. `CUG`

(c) Select all and only the true statements

i. $\forall s \in S\ \exists b \in B\ \ (\ rnalen(s) = basecount(s, b)\ )$

ii. $\exists s \in S\ \forall b \in B\ \ (\ rnalen(s) = basecount(s, b)\ )$

iii.

$$\forall s_1 \in S\ \forall s_2 \in S\ \forall b \in B\ \Big(\ \big(rnalen(s_1) = basecount(s_1, b) \\ \land\ rnalen(s_2) = basecount(s_2, b) \land rnalen(s_1) = rnalen(s_2)\big) \to s_1$$

2. **Wednesday** 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 $sumPow(x+1) = sumPow(x)+ 2^{x+1}$.

   (a) Calculate $sumPow(0)$
   (b) Calculate $sumPow(1)$
   (c) Calculate $sumPow(2)$

3. **Wednesday 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 $(n, l) \in L$

Definition The function $length : L \to \mathbb{N}$ that computes the length of a list is:

		$length : L$	$\to \mathbb{N}$
Basis Step:		$length([])$	$= 0$
Recursive Step:	If $l \in L$ and $n \in \mathbb{N}$, then	$length((n, l))$	$= 1 + length(l)$

Consider this (incomplete) definition:

Definition The function $increment :$ ____________ that adds 1 to each element of a linked list is defined by:

		$increment :$ ______	$\to$ ______
Basis Step:		$increment([])$	$= []$
Recursive Step:	If $l \in L, n \in \mathbb{N}$	$increment((n, l))$	$= (1 + n, increment(l))$

Consider this (incomplete) definition:

Definition The function $sum : L \to \mathbb{N}$ that adds together all the elements of the list is defined by:

		$sum : L$	$\to \mathbb{N}$
Basis Step:		$sum([])$	$= 0$
Recursive Step:	If $l \in L, n \in \mathbb{N}$	$sum((n, l))$	$=$ ______________

(a) (You will compute a sample function application and then fill in the blanks for the domain and codomain) Based on the definition, what is the result of $increment((4,(2,(7,[]))))$? Write your answer directly with no spaces.

(b) Which of the following describes the domain and codomain of $increment$?

i. $L \to \mathbb{N}$

ii. $L \to \mathbb{N} \times L$

iii. $L \times \mathbb{N} \to L$

iv. $L \times \mathbb{N} \to \mathbb{N}$

v. $L \to L$

vi. None of the above

(c) Assuming we would like $sum((5,(6,[])))$ to evaluate to 11 and $sum((3,(1,(8,[]))))$ to evaluate to 12, which of the following could be used to fill in the definition of the recursive case of sum?

i. $\begin{cases} 1 + sum(l) & \text{when } n \neq 0 \\ sum(l) & \text{when } n = 0 \end{cases}$

ii. $1 + sum(l)$

iii. $n + increment(l)$

iv. $n + sum(l)$

v. None of the above

(d) Choose only and all of the following statements that are **well-defined**; that is, they correctly reflect the domains and codomains of the functions and quantifiers, and respect the notational conventions we use in this class. Note that a well-defined statement may be true or false.

i. $\forall l \in L\,(sum(l))$

ii. $\exists l \in L\,(sum(l) \wedge length(l))$

iii. $\forall l \in L\,(sum(increment(l)) = 10)$

iv. $\exists l \in L\,(sum(increment(l)) = 10)$

v. $\forall l \in L\, \forall n \in \mathbb{N}\,((n \times l) \subseteq L)$

vi. $\forall l_1 \in L\, \exists l_2 \in$

$L\,(increment(sum(l_1)) = l_2)$ vii. $\forall l \in L\,(length(increment(l)) = length(l))$

(e) Choose only and all of the statements in the previous part that are both well-defined and true.

4. **Friday** Recall the set P defined by the recursive definition

Basis Step:	$(0,0) \in P$
Recursive Step:	If $(x,y) \in P$ then $(x+1,y+1) \in P$ and $(x+1,y-1) \in P$ and $(x-1,y-1) \in P$ and $(x-1,y+1) \in P$

(a) Select all and only the ordered pairs below that are elements of P

i. $(0,0)$
ii. $(4,0)$
iii. $(1,1)$
iv. $(1.5,2.5)$
v. $(0,-2)$

(b) What is another description of the set P ? (Select all and only the true descriptions.)

i. $\mathbb{Z} \times \mathbb{Z}$
ii. $\{(n,n) \mid n \in \mathbb{Z}\}$
iii. $\{(a,b) \in \mathbb{Z} \times \mathbb{Z} \mid (a+b) \textbf{ mod } 2 = 0\}$