CSE 20
DISCRETE MATH

Fall 2020

http://cseweb.ucsd.edu/classes/fa20/cse20-a/

Learning goals

Today's goals

- Practice with properties of recursively defined sets and functions
- Prove and disprove properties of recursively defined sets and functions with structural induction

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

$$\begin{array}{ll} \text{Basis Step:} & \mathtt{A} \in S, \mathtt{C} \in S, \mathtt{U} \in S, \mathtt{G} \in S \\ \text{Recursive Step:} & \text{If } s \in S \text{ and } b \in B \text{, then } sb \in S \end{array}$$

where sb is string concatenation.

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

$$\begin{array}{lllll} & & \mathit{rnalen} : S & \to \mathbb{Z}^+ \\ \text{Basis Step:} & \text{If } b \in B \text{ then} & \mathit{rnalen}(b) & = 1 \\ \text{Recursive Step:} & \text{If } s \in S \text{ and } b \in B \text{, then} & \mathit{rnalen}(sb) & = 1 + \mathit{rnalen}(s) \end{array}$$

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

$$\begin{array}{llll} & & \mathit{basecount} : S \times B & \to \mathbb{N} \\ \text{Basis Step:} & \text{If } b_1 \in B, b_2 \in B & \mathit{basecount}(b_1, b_2) & = \begin{cases} 1 & \text{when } b_1 = b_2 \\ 0 & \text{when } b_1 \neq b_2 \end{cases} \\ \text{Recursive Step:} & \text{If } s \in S, b_1 \in B, b_2 \in B & \mathit{basecount}(sb_1, b_2) & = \begin{cases} 1 + \mathit{basecount}(s, b_2) & \text{when } b_1 = b_2 \\ \mathit{basecount}(s, b_2) & \text{when } b_1 \neq b_2 \end{cases} \end{array}$$

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

$$
\begin{array}{llll}
 & & rnalen: S & \to \mathbb{Z}^+ \\
\text{Basis Step:} & \text{If } b \in B \text{ then} & rnalen(b) & = 1 \\
\text{Recursive Step:} & \text{If } s \in S \text{ and } b \in B\text{, then} & rnalen(sb) & = 1 + rnalen(s)
\end{array}
$$

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

$$
\begin{array}{llll}
 & & basecount: S \times B & \to \mathbb{N} \\
\text{Basis Step:} & \text{If } b_1 \in B, b_2 \in B & basecount(b_1, b_2) & = \begin{cases} 1 & \text{when } b_1 = b_2 \\ 0 & \text{when } b_1 \neq b_2 \end{cases} \\
\text{Recursive Step:} & \text{If } s \in S, b_1 \in B, b_2 \in B & basecount(sb_1, b_2) & = \begin{cases} 1 + basecount(s, b_2) & \text{when } b_1 = b_2 \\ basecount(s, b_2) & \text{when } b_1 \neq b_2 \end{cases}
\end{array}
$$

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

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 $s \in S, b_1 \in B, b_2 \in B$

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

Which of the following statements is true?

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

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

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

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

E. None of the above

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

Which proof strategy is appropriate?

A. Exhaustion
B. Universal generalization
C. Witness
D. Direct proof
E. Proof by cases

New!

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

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.

New!

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

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.

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)

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

For next time

- Read website carefully

http://cseweb.ucsd.edu/classes/fa20/cse20-a/

Pre class reading for next time: Example 1 Section 5.1 p316