

## Sets and Recursive Definitions

RNA is made up of strands of four different bases that match up in specific ways.

The bases are elements of the set  $B = \{\mathbf{A}, \mathbf{C}, \mathbf{G}, \mathbf{U}\}$ .

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

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

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

where  $sb$  is string concatenation.

Examples:

To define a set we can use the **roster method**, the **set builder notation**, and also ...

**New! Recursive Definitions of Sets:** The set  $S$  (pick a name) is defined by:

Basis Step: Specify finitely many elements of  $S$

Recursive Step: Give a rule for creating a new element of  $S$  from known values existing in  $S$  and potentially other values.

The set  $S$  then consists of all and only elements that are put in  $S$  by finitely many (a nonnegative integer number) of applications of the recursive step after the basis step.

**Definition** (Rosen p. 123) Let  $A$  and  $B$  be sets. The **Cartesian product** of  $A$  and  $B$ , denoted  $A \times B$ , is the set of all ordered pairs  $(a, b)$  where  $a \in A$  and  $b \in B$

$$A \times B = \{(a, b) \mid a \in A \text{ and } b \in B\}$$

**Definition:** Let  $A$  and  $B$  be sets of strings over the same alphabet. The **set-wise concatenation** of  $A$  and  $B$ , denoted  $A \circ B$ , is the set of all results of string concatenation  $ab$  where  $a \in A$  and  $b \in B$

$$A \circ B = \{ab \mid a \in A \text{ and } b \in B\}$$

Set	Example elements in this set:			
$B$	A	C	G	U
	(A, C)		(U, U)	
$B \times \{-1, 0, 1\}$				
$\{-1, 0, 1\} \times B$				
	(0, 0, 0)			
$\{A, C, G, U\} \circ \{A, C, G, U\}$				
	GGGG			

The bases of RNA strands are elements of the set  $B = \{\mathbf{A}, \mathbf{C}, \mathbf{G}, \mathbf{U}\}$ . Each of the sets below is described using set builder notation. Rewrite them using the roster method.

- $\{s \in S \mid s \text{ has length } 2\}$
- $\{s \in S \mid \text{the leftmost base in } s \text{ is the same as the rightmost base in } s \text{ and } s \text{ has length } 3\}$
- $\{s \in S \mid \text{there are twice as many As as Cs in } s \text{ and } s \text{ has length } 1\}$

Certain sequences of bases serve important biological functions in translating RNA to proteins. The following recursive definition gives a special set of RNA strands: The set of RNA strands  $\hat{S}$  is defined (recursively) by

Basis step:  $\mathbf{AUG} \in \hat{S}$

Basis step: If  $s \in \hat{S}$  and  $x \in R$ , then  $sx \in \hat{S}$

where  $R = \{\mathbf{UUU}, \mathbf{CUC}, \mathbf{AUC}, \mathbf{AUG}, \mathbf{GUU}, \mathbf{CCU}, \mathbf{GCU}, \mathbf{UGG}, \mathbf{GGA}\}$ .

Each of the sets below is described using set builder notation. Rewrite them using the roster method.

- $\{s \in \hat{S} \mid s \text{ has length less than or equal to } 5\}$
- $\{s \in S \mid \text{there are twice as many Cs as As in } s \text{ and } s \text{ has length } 6\}$

## Review quiz questions

1. RNA is made up of strands of four different bases that match up in specific ways. The bases are elements of the set  $B = \{\mathbf{A}, \mathbf{C}, \mathbf{G}, \mathbf{U}\}$ .

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

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

A function  $rnalen$  that computes the length of RNA strands in  $S$  is defined by:

$rnalen : S \rightarrow \mathbb{Z}^+$   
 Basis Step: If  $b \in B$  then  $rnalen(b) = 1$   
 Recursive Step: If  $s \in S$  and  $b \in B$ , then  $rnalen(sb) = 1 + rnalen(s)$

- (a) How many distinct elements are in the set described using set builder notation as

$$\{x \in S \mid rnalen(x) = 1\} \quad ?$$

- (b) How many distinct elements are in the set described using set builder notation as

$$\{x \in S \mid rnalen(x) = 2\} \quad ?$$

- (c) How many distinct elements are in the set described using set builder notation as

$$\{rnalen(x) \mid x \in S \text{ and } rnalen(x) = 2\} \quad ?$$

- (d) How many distinct elements are in the set obtained as the result of the set-wise concatenation  $\{\mathbf{AA}, \mathbf{AC}\} \circ \{\mathbf{U}, \mathbf{AA}\}$ ?
- (e) How many distinct elements are in the set obtained as the result of the Cartesian product  $\{\mathbf{AA}, \mathbf{AC}\} \times \{\mathbf{U}, \mathbf{AA}\}$ ?
- (f) **True or False:** There is an example of an RNA strand that is both in the set obtained as the result of the set-wise concatenation  $\{\mathbf{AA}, \mathbf{AC}\} \circ \{\mathbf{U}, \mathbf{AA}\}$  and in the set obtained as the result of the Cartesian product  $\{\mathbf{AA}, \mathbf{AC}\} \times \{\mathbf{UA}, \mathbf{AA}\}$

*Bonus - not for credit: Describe each of the sets above using roster method.*

## Predicates and Quantified Statements

Recall: Each RNA strand is a string whose symbols are elements of the set  $B = \{\mathbf{A}, \mathbf{C}, \mathbf{G}, \mathbf{U}\}$ . The **set of all RNA strands** is called  $S$ . The function  $rnalen$  that computes the length of RNA strands in  $S$  is:

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

### Example predicates on $S$

$H(s) = T$	Truth set of $H$ is _____
$L_3(s) = \begin{cases} T & \text{if } rnalen(s) = 3 \\ F & \text{otherwise} \end{cases}$	Strand where $L_3$ evaluates to $T$ is e.g. Strand where $L_3$ evaluates to $F$ is e.g.
$F_{\mathbf{A}}$ is defined recursively by: Basis step: $F_{\mathbf{A}}(\mathbf{A}) = T, F_{\mathbf{A}}(\mathbf{C}) = F_{\mathbf{A}}(\mathbf{G}) = F_{\mathbf{A}}(\mathbf{U}) = F$ Recursive step: If $s \in S$ and $b \in B$ , then $F_{\mathbf{A}}(sb) = F_{\mathbf{A}}(s)$	Strand where $F_{\mathbf{A}}$ evaluates to $T$ is e.g. Strand where $F_{\mathbf{A}}$ evaluates to $F$ is e.g.
$P_{\mathbf{AUC}}$ is defined as the predicate whose truth set is the collection of RNA strands where the string <b>AUC</b> is a substring (appears inside $s$ , in order and consecutively)	Strand where $P_{\mathbf{AUC}}$ evaluates to $T$ is e.g. Strand where $P_{\mathbf{AUC}}$ evaluates to $F$ is e.g.

**Definition** (Rosen p123): The **Cartesian product** of the sets  $A$  and  $B$ ,  $A \times B$ , is the set of all ordered pairs  $(a, b)$ , where  $a \in A$  and  $b \in B$ . That is:  $A \times B = \{(a, b) \mid (a \in A) \wedge (b \in B)\}$ . The Cartesian product of the sets  $A_1, A_2, \dots, A_n$ , denoted by  $A_1 \times A_2 \times \dots \times A_n$ , is the set of ordered  $n$ -tuples  $(a_1, a_2, \dots, a_n)$ , where  $a_i$  belongs to  $A_i$  for  $i = 1, 2, \dots, n$ . That is,  $A_1 \times A_2 \times \dots \times A_n = \{(a_1, a_2, \dots, a_n) \mid a_i \in A_i \text{ for } i = 1, 2, \dots, n\}$

Recall: Each RNA strand is a string whose symbols are elements of the set  $B = \{A, C, G, U\}$ . The **set of all RNA strands** is called  $S$ . The function *rnalen* that computes the length of RNA strands in  $S$  is:

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

A function *basecount* that computes the number of a given base  $b$  appearing in a RNA strand  $s$  is:

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

$L$  with domain  $S \times \mathbb{Z}^+$  is defined by, for  $s \in S$  and  $n \in \mathbb{Z}^+$ ,

$$L(s, n) = \begin{cases} T & \text{if } \text{rnalen}(s) = n \\ F & \text{otherwise} \end{cases}$$

Element where  $L$  evaluates to  $T$ : \_\_\_\_\_

Element where  $L$  evaluates to  $F$ : \_\_\_\_\_

$BC$  with domain \_\_\_\_\_ is defined  
  $S$  and  $b \in B$  and  $n \in \mathbb{N}$ ,

$$BC(s, b, n) = \begin{cases} T & \text{if } \text{basecount}(s, b) = n \\ F & \text{otherwise} \end{cases}$$

Element where  $BC$  evaluates to  $T$ : \_\_\_\_\_

Element where  $BC$  evaluates to  $F$ : \_\_\_\_\_

**Notation:** for a predicate  $P$  with domain  $X_1 \times \dots \times X_n$  and a  $n$ -tuple  $(x_1, \dots, x_n)$  with each  $x_i \in X$ , we write  $P(x_1, \dots, x_n)$  to mean  $P((x_1, \dots, x_n))$ .

$\exists t \, BC(t)$       In English: \_\_\_\_\_  
 Witness that proves this existential quantification is true: \_\_\_\_\_  
 $\forall (s, b, n) ( \, BC(s, b, n) \, )$       In English: \_\_\_\_\_  
 Counterexample that proves this universal quantification is false: \_\_\_\_\_  
**New predicates from old**       $BC(s, b, n)$  means *basecount*( $s, b$ ) =  $n$ .

Predicate	Domain	Example domain element where predicate is $T$
$basecount(s, b) = 3$		
$basecount(s, A) = n$		
$\exists n \in \mathbb{N} (basecount(s, b) = n)$		
$\forall b \in B (basecount(s, b) = 1)$		

### Alternating quantifiers

$$\forall s \exists n \, BC(s, A, n)$$

In English: \_\_\_\_\_

$$\exists n \forall s \, BC(s, U, n)$$

In English: \_\_\_\_\_

## Review quiz questions

1. Real-life representations are often prone to corruptions. Biological codes, like RNA, may mutate naturally<sup>1</sup> and during measurement; cosmic radiation and other ambient noise can flip bits in computer storage<sup>2</sup>. One way to recover from corrupted data is to exploit redundancy. Consider the following algorithm to introduce redundancy in a string of 0s and 1s.

Create redundancy by repeating each bit three times

---

```
1 procedure redun3( $a_{k-1} \cdots a_0$ : a binary string)
2 for  $i := 0$  to  $k-1$ 
3    $c_{3i} := a_i$ 
4    $c_{3i+1} := a_i$ 
5    $c_{3i+2} := a_i$ 
6 return  $c_{3k-1} \cdots c_0$ 
```

---

Decode sequence of bits using majority rule on consecutive three bit sequences

---

```
1 procedure decode3( $c_{3k-1} \cdots c_0$ : a binary string whose length is an integer multiple of 3)
2 for  $i := 0$  to  $k-1$ 
3   if exactly two or three of  $c_{3i}, c_{3i+1}, c_{3i+2}$  are set to 1
4      $a_i := 1$ 
5   else
6      $a_i := 0$ 
7 return  $a_{k-1} \cdots a_0$ 
```

---

For each of the following, type in your answers precisely including all notational punctuation.

- (a) Give the output of *redun3*(100).
- (b) If the output of running *redun3* is 000000111000111, what was its input?
- (c) Give the output of *decode3*(100).
- (d) How many distinct possible inputs to *decode3* give the output 01?

---

<sup>1</sup>Mutations of specific RNA codons have been linked to many disorders and cancers.

<sup>2</sup>This RadioLab podcast episode goes into more detail on bit flips: <https://www.wnycstudios.org/story/bit-flip>



2. Recall that  $S$  is defined as the set of all RNA strands, strings made of the bases in  $B = \{\mathbf{A}, \mathbf{U}, \mathbf{G}, \mathbf{C}\}$ . Define the functions *mutation*, *insertion*, and *deletion* as described by the pseudocode below:

---

```

1  procedure mutation( $b_1 \cdots b_n$ : a RNA strand,  $k$ : a positive integer,  $b$ : an element of  $B$ )
2  for  $i := 1$  to  $n$ 
3      if  $i = k$ 
4           $c_i := b$ 
5      else
6           $c_i := b_i$ 
7  return  $c_1 \cdots c_n$  {The return value is a RNA strand made of the  $c_i$  values}

```

---

```

1  procedure insertion( $b_1 \cdots b_n$ : a RNA strand,  $k$ : a positive integer,  $b$ : an element of  $B$ )
2  if  $k > n$ 
3      for  $i := 1$  to  $n$ 
4           $c_i := b_i$ 
5       $c_{n+1} := b$ 
6  else
7      for  $i := 1$  to  $k - 1$ 
8           $c_i := b_i$ 
9       $c_k := b$ 
10     for  $i := k + 1$  to  $n + 1$ 
11          $c_i := b_{i-1}$ 
12 return  $c_1 \cdots c_{n+1}$  {The return value is a RNA strand made of the  $c_i$  values}

```

---

```

1  procedure deletion( $b_1 \cdots b_n$ : a RNA strand,  $k$ : a positive integer)
2  if  $k > n$ 
3       $m := n$ 
4      for  $i := 1$  to  $n$ 
5           $c_i := b_i$ 
6  else
7       $m := n - 1$ 
8      for  $i := 1$  to  $k - 1$ 
9           $c_i := b_i$ 
10     for  $i := k$  to  $n - 1$ 
11          $c_i := b_{i+1}$ 
12 return  $c_1 \cdots c_m$  {The return value is a RNA strand made of the  $c_i$  values}

```

---

For this question, we will use the following predicates.

$F_{\mathbf{A}}$  with domain  $S$  is defined recursively by:

Basis step:  $F_{\mathbf{A}}(\mathbf{A}) = T$ ,  $F_{\mathbf{A}}(\mathbf{C}) = F_{\mathbf{A}}(\mathbf{G}) = F_{\mathbf{A}}(\mathbf{U}) = F$

Recursive step: If  $s \in S$  and  $b \in B$ , then  $F_{\mathbf{A}}(sb) = F_{\mathbf{A}}(s)$

$P_{\text{AUC}}$  with domain  $S$  is defined as the predicate whose truth set is the collection of RNA strands where the string **AUC** is a substring (appears inside  $s$ , in order and consecutively)

$L$  with domain  $S \times \mathbb{Z}^+$  is defined by, for  $s \in S$  and  $n \in \mathbb{Z}^+$ ,

$$L(s, n) = \begin{cases} T & \text{if } rna len(s) = n \\ F & \text{otherwise} \end{cases}$$

$Mut$  with domain  $S \times S$  is defined by, for  $s_1 \in S$  and  $s_2 \in S$ ,

$$Mut(s_1, s_2) = \exists k \in \mathbb{Z}^+ \exists b \in B( \text{mutation}(s_1, k, b) = s_2 )$$

$Ins$  with domain  $S \times S$  is defined by, for  $s_1 \in S$  and  $s_2 \in S$ ,

$$Ins(s_1, s_2) = \exists k \in \mathbb{Z}^+ \exists b \in B( \text{insertion}(s_1, k, b) = s_2 )$$

$Del$  with domain  $S \times S$  is defined by, for  $s_1 \in S$  and  $s_2 \in S$ ,

$$Del(s_1, s_2) = \exists k \in \mathbb{Z}^+ ( \text{deletion}(s_1, k) = s_2 )$$

(a) Which of the following is true? (Select all and only that apply.)

- i.  $F_A(\mathbf{AA})$
- ii.  $F_A(\mathbf{AC})$
- iii.  $F_A(\mathbf{AG})$
- iv.  $F_A(\mathbf{AU})$
- v.  $F_A(\mathbf{CA})$
- vi.  $F_A(\mathbf{CC})$
- vii.  $F_A(\mathbf{CG})$
- viii.  $F_A(\mathbf{CU})$

(b) Which of the following is true? (Select all and only that apply.)

- i.  $\exists s \in S \exists n \in \mathbb{Z}^+ (L(s, n))$
- ii.  $\exists s \in S \forall n \in \mathbb{Z}^+ (L(s, n))$
- iii.  $\forall n \in \mathbb{Z}^+ \exists s \in S (L(s, n))$
- iv.  $\forall s \in S \exists n \in \mathbb{Z}^+ (L(s, n))$
- v.  $\exists n \in \mathbb{Z}^+ \forall s \in S (L(s, n))$
- vi.  $\forall s \in S \forall n \in \mathbb{Z}^+ (L(s, n))$

(c) Which of the following is true? (Select all and only that apply.)

- i.  $\exists s \in S Mut(s, s)$

- ii.  $\forall s \in S \text{ } Mut(s, s)$
- iii.  $\exists s \in S \text{ } Ins(s, \mathbf{A})$
- iv.  $\exists s \in S \text{ } Ins(\mathbf{A}, s)$
- v.  $\exists s \in S \text{ } Del(s, \mathbf{A})$
- vi.  $\forall s \in S \text{ } Del(s, \mathbf{A})$

# Induction

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

Basis Step:  $\mathbf{A} \in S, \mathbf{C} \in S, \mathbf{U} \in S, \mathbf{G} \in S$   
 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  $rnalen(b) = 1$

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

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  $\forall s \in S (rnalen(s) \geq basecount(s, \mathbf{A}))$ :

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

**Proof:**

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

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

Case 2: Want to show  $(s = \mathbf{C} \vee s = \mathbf{U} \vee s = \mathbf{G}) \rightarrow (rnalen(s) \geq basecount(s, \mathbf{A}))$ .

*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

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

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

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

Need to show

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

Consider arbitrary  $b \in B$ .

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

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

## Review quiz questions

1. The function *rnalen* that computes the length of RNA strands in  $S$  is defined recursively by  $rnalen : S \rightarrow \mathbb{Z}^+$

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

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

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

- i. A
- ii. UU
- iii. CU
- iv. (U, 1)

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

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

- 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.

$\forall s_1 \in S \forall s_2 \in S \forall b \in B \ ( \ (rnalen(s_1) = basecount(s_1, b)$   
 $\wedge rnalen(s_2) = basecount(s_2, b) \wedge rnalen(s_1) = rnalen(s_2)) \rightarrow s_1$

## Cardinality

Let  $S_2$  be the set of RNA strands of length 2.

Statement	True/False , justification
$ \{A, U, G, C\}  \leq  S_2 $	
$ \{A, U, G, C\} \times \{A, U, G, C\}  \leq  S_2 $	

Let  $S_2$  be the set of RNA strands of length 2.

Statement	True/False , justification
$ S_2  \geq  \{A, U, G, C\} $	
$ S_2  \geq  \{A, U, G, C\} \times \{A, U, G, C\} $	



## Review quiz questions

- Recall the definitions from previous assignments and class: The bases of RNA are elements of the set  $B = \{\mathbf{A}, \mathbf{C}, \mathbf{G}, \mathbf{U}\}$ . The set of RNA strands  $S$  is defined (recursively) by:

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

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

For  $b$  an integer greater than 1 and  $n$  a positive integer, the **base  $b$  expansion of  $n$**  is

$$(a_{k-1} \cdots a_1 a_0)_b$$

where  $k$  is a positive integer,  $a_0, a_1, \dots, a_{k-1}$  are nonnegative integers less than  $b$ ,  $a_{k-1} \neq 0$ , and

$$n = a_{k-1}b^{k-1} + \cdots + a_1b + a_0$$

For  $b$  an integer greater than 1,  $w$  a positive integer, and  $n$  a nonnegative integer with  $n < b^w$ , the **base  $b$  fixed-width  $w$  expansion of  $n$**  is

$$(a_{w-1} \cdots a_1 a_0)_{b,w}$$

where  $a_0, a_1, \dots, a_{w-1}$  are nonnegative integers less than  $b$  and

$$n = a_{w-1}b^{w-1} + \cdots + a_1b + a_0$$

For  $b$  an integer greater than 1,  $w$  a positive integer,  $w'$  a positive integer, and  $x$  a real number the **base  $b$  fixed-width expansion of  $x$  with integer part width  $w$  and fractional part width  $w'$**  is

$$(a_{w-1} \cdots a_1 a_0 . c_1 \cdots c_{w'})_{b,w,w'}$$

where  $a_0, a_1, \dots, a_{w-1}, c_1, \dots, c_{w'}$  are nonnegative integers less than  $b$  and

$$x \geq a_{w-1}b^{w-1} + \cdots + a_1b + a_0 + c_1b^{-1} + \cdots + c_{w'}b^{-w'}$$

and

$$x < a_{w-1}b^{w-1} + \cdots + a_1b + a_0 + c_1b^{-1} + \cdots + (c_{w'} + 1)b^{-w'}$$

For each set below, determine if it is empty, nonempty and finite, countably infinite, or uncountable.

*Challenge - not to hand in:* how would you prove this?

- (a)  $B$
- (b)  $S$
- (c)  $\{x \in \mathbb{N} \mid x = (4102)_3\}$
- (d)  $\{x \in \mathbb{R} \mid x \text{ has a binary fixed-width 5 expansion}\}$
- (e)  $\{x \in \mathbb{R} \mid x = (0.10)_{(2,1,2)}\}$

## (Equivalence) Relations

*Example:* Recall that  $S$  is defined as the set of all RNA strands, strings made of the bases in  $B = \{\mathbf{A}, \mathbf{U}, \mathbf{G}, \mathbf{C}\}$ . Define the functions *mutation*, *insertion*, and *deletion* as described by the pseudocode below:

---

```

1 procedure mutation( $b_1 \dots b_n$ : a RNA strand,  $k$ : a positive integer,  $b$ : an element of  $B$ )
2   for  $i := 1$  to  $n$ 
3     if  $i = k$ 
4        $c_i := b$ 
5     else
6        $c_i := b_i$ 
7   return  $c_1 \dots c_n$  {The return value is a RNA strand made of the  $c_i$  values}

```

---

```

1 procedure insertion( $b_1 \dots b_n$ : a RNA strand,  $k$ : a positive integer,  $b$ : an element of  $B$ )
2   if  $k > n$ 
3     for  $i := 1$  to  $n$ 
4        $c_i := b_i$ 
5      $c_{n+1} := b$ 
6   else
7     for  $i := 1$  to  $k-1$ 
8        $c_i := b_i$ 
9      $c_k := b$ 
10    for  $i := k+1$  to  $n+1$ 
11       $c_i := b_{i-1}$ 
12  return  $c_1 \dots c_{n+1}$  {The return value is a RNA strand made of the  $c_i$  values}

```

---

```

1 procedure deletion( $b_1 \dots b_n$ : a RNA strand,  $k$ : a positive integer)
2   if  $k > n$ 
3      $m := n$ 
4     for  $i := 1$  to  $n$ 
5        $c_i := b_i$ 
6   else
7      $m := n-1$ 
8     for  $i := 1$  to  $k-1$ 
9        $c_i := b_i$ 
10    for  $i := k$  to  $n-1$ 
11       $c_i := b_{i+1}$ 
12  return  $c_1 \dots c_m$  {The return value is a RNA strand made of the  $c_i$  values}

```

---

$Mut$  with domain  $S \times S$  is defined by, for  $s_1 \in S$  and  $s_2 \in S$ ,

$$Mut(s_1, s_2) = \exists k \in \mathbb{Z}^+ \exists b \in B( \text{mutation}(s_1, k, b) = s_2 )$$

$Ins$  with domain  $S \times S$  is defined by, for  $s_1 \in S$  and  $s_2 \in S$ ,

$$Ins(s_1, s_2) = \exists k \in \mathbb{Z}^+ \exists b \in B( \text{insertion}(s_1, k, b) = s_2 )$$

$Del$  with domain  $S \times S$  is defined by, for  $s_1 \in S$  and  $s_2 \in S$ ,

$$Del(s_1, s_2) = \exists k \in \mathbb{Z}^+ ( \text{deletion}(s_1, k) = s_2 )$$

**Definition:** We say that a RNA strand  $s_1$  is “within one edit” of a RNA strand  $s_2$  to mean

$$Mut(s_1, s_2) \vee Mut(s_2, s_1) \vee Ins(s_1, s_2) \vee Ins(s_2, s_1) \vee Del(s_1, s_2) \vee Del(s_2, s_1)$$

$$within1_{TF} : \underline{\hspace{2cm}} \rightarrow \underline{\hspace{2cm}}$$

$$within1_{\mathcal{P}} : \underline{\hspace{1cm}} \rightarrow \underline{\hspace{2cm}}$$

$$within1_{TF}(s_1, s_2) = \underline{\hspace{4cm}}$$

$$within1_{\mathcal{P}}(s_1) = \underline{\hspace{4cm}}$$

$$W_1 = \{ \underline{\hspace{4cm}} \}$$