

Algorithm redundancy

Real-life representations are often prone to corruption. Biological codes, like RNA, may mutate naturally¹ and during measurement; cosmic radiation and other ambient noise can flip bits in computer storage². One way to recover from corrupted data is to introduce or 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 nonempty bitstring)
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 nonempty bitstring 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  $a_{k-1} \cdots a_0$ 

```

Give a recursive definition of the set of outputs of the *redun3* procedure, *Out*,

Consider the message $m = 0001$ so that the sender calculates $\text{redun3}(m) = \text{redun3}(0001) = 000000000111$.

Introduce ____ errors into the message so that the signal received by the receiver is _____ but the receiver is still able to decode the original message.

Challenge: what is the biggest number of errors you can introduce?

Building a circuit for lines 3-6 in *decode* procedure: given three input bits, we need to determine whether the majority is a 0 or a 1.

c_{3i}	c_{3i+1}	c_{3i+2}	a_i	Circuit
1	1	1		
1	1	0		
1	0	1		
1	0	0		
0	1	1		
0	1	0		
0	0	1		
0	0	0		

¹Mutations of specific RNA codons have been linked to many disorders and cancers.

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

Algorithm rna mutation insertion deletion

Recall that S is defined as the set of all RNA strands, nonempty strings made of the bases in $B = \{\text{A}, \text{U}, \text{G}, \text{C}\}$. We define the functions

$$\text{mutation} : S \times \mathbb{Z}^+ \times B \rightarrow S \qquad \text{insertion} : S \times \mathbb{Z}^+ \times B \rightarrow S$$

$$\text{deletion} : \{s \in S \mid \text{rlen}(s) > 1\} \times \mathbb{Z}^+ \rightarrow S \qquad \text{with rules}$$

```

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 with  $n > 1$ ,  $k$ : 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}

```

Alternating quantifiers order rna examples

Alternating nested quantifiers

$$\forall s \in S \ \exists n \in \mathbb{N} \ (\textit{basecount}(\ (s, \mathsf{U}) \) = n \)$$

In English: For each strand, there is a nonnegative integer that counts the number of occurrences of U in that strand.

$$\exists n \in \mathbb{N} \ \forall s \in S \ (\textit{basecount}(\ (s, \mathsf{U}) \) = n \)$$

In English: There is a nonnegative integer that counts the number of occurrences of U in every strand.

Are these statements true or false?

$$\forall s \in S \exists b \in B (\text{basecount}(s, b) = 3)$$

In English: For each RNA strand there is a base that occurs 3 times in this strand.

Write the negation and use De Morgan's law to find a logically equivalent version where the negation is applied only to the BC predicate (not next to a quantifier).

Is the original statement **True** or **False**?

Predicate rna example

Example predicates on S , the set of RNA strands (an infinite set)

$L_A : S \rightarrow \{T, F\}$ defined recursively by:

Basis step: $L_A(A) = T$, $L_A(C) = L_A(G) = L_A(U) = F$

Recursive step: If $s \in S$ and $b \in B$, then $L_A(sb) = L_A(s)$.

Example where L_A evaluates to T is _____

Example where L_A evaluates to F is _____

$H : S \rightarrow \{T, F\}$ where $H(s) = T$ for all s .

Truth set of H is _____

Predicates example rnalen basecount

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

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

In other words, $L((s, n))$ means $rnalen(s) = n$

BC with domain $S \times B \times \mathbb{N}$ is defined by, for $s \in S$ and $b \in B$ and $n \in \mathbb{N}$,

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

In other words, $BC((s, b, n))$ means $basecount((s, b)) = n$

Example where L evaluates to T : _____ Why?

Example where BC evaluates to T : _____ Why?

Example where L evaluates to F : _____ Why?

Example where BC evaluates to F : _____ Why?

$$\exists t \ BC(t) \qquad \exists (s, b, n) \in S \times B \times \mathbb{N} \ (basecount((s, b)) = n)$$

In English:

Witness that proves this existential quantification is true:

$$\forall t \ BC(t) \qquad \forall (s, b, n) \in S \times B \times \mathbb{N} \ (basecount((s, b)) = n)$$

In English:

Counterexample that proves this universal quantification is false:

Alternating quantifiers strategies rna examples

Alternating nested quantifiers

$$\forall s \in S \exists b \in B (\text{basecount}(s, b) = 3)$$

In English: For each RNA strand there is a base that occurs 3 times in this strand.

Write the negation and use De Morgan's law to find a logically equivalent version where the negation is applied only to the BC predicate (not next to a quantifier).

Is the original statement **True** or **False**?

$$\exists s \in S \forall b \in B \exists n \in \mathbb{N} (\text{basecount}(s, b) = n)$$

In English: There is an RNA strand so that for each base there is some nonnegative integer that counts the number of occurrences of that base in this strand.

Write the negation and use De Morgan's law to find a logically equivalent version where the negation is applied only to the BC predicate (not next to a quantifier).

Is the original statement **True** or **False**?

Alternating quantifiers proofs rna examples

Which proof strategies could be used to prove each of the following statements?

Hint: first translate the statements to English and identify the main logical structure.

$$\forall b \in B \exists s \in S (\text{basecount}(s, b) > 0)$$

$$\exists s \in S \forall b \in B (\text{basecount}(s, b) > 0)$$

$$\exists s \in S (\text{rinalen}(s) = \text{basecount}(s, \mathbf{A}))$$

$$\forall s \in S \exists b \in B (\text{basecount}(s, b) > 0)$$

$$\forall s \in S (\text{rinalen}(s) \geq \text{basecount}(s, \mathbf{A}))$$

$$\forall s \in S (\text{rinalen}(s) > 0)$$

Defining functions more examples

Let's practice with functions related to some of our applications so far.

Recall: We model the collection of user ratings of the four movies Superman, Wicked, KPop Demon Hunters, A Minecraft Movie as the set $\{-1, 0, 1\}^4$. One function that compares pairs of ratings is

$$d_0 : \{-1, 0, 1\}^4 \times \{-1, 0, 1\}^4 \rightarrow \mathbb{R}$$

given by

$$d_0((x_1, x_2, x_3, x_4), (y_1, y_2, y_3, y_4)) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 + (x_4 - y_4)^2}$$

Notice: any ordered pair of ratings is an okay input to d_0 .

Notice: there are (at most)

$$(3 \cdot 3 \cdot 3 \cdot 3) \cdot (3 \cdot 3 \cdot 3 \cdot 3) = 3^8 = 6561$$

many pairs of ratings. There are therefore lots and lots of real numbers that are not the output of d_0 .

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

Pro-tip: informal definitions sometime use $\dots$ to indicate “continue the pattern”. Often, to make this pattern precise we use recursive definitions.

Name	Domain	Codomain	Rule	Example
<i>rnalen</i>	S	$\mathbb{Z}^+$	Basis Step: If $b \in B$ then $rnalen(b) = 1$ Recursive Step: If $s \in S$ and $b \in B$, then $rnalen(sb) = 1 + rnalen(s)$	$rnalen(\mathbf{AC}) \stackrel{\text{rec step}}{=} 1 + rnalen(\mathbf{A})$ $\stackrel{\text{basis step}}{=} 1 + 1 = 2$
<i>basecount</i>	$S \times B$	$\mathbb{N}$	Basis Step: If $b_1 \in B, b_2 \in B$ then $basecount((b_1, b_2)) =$ $\begin{cases} 1 & \text{when } b_1 = b_2 \\ 0 & \text{when } b_1 \neq b_2 \end{cases}$ Recursive Step: If $s \in S, b_1 \in B, b_2 \in B$ $basecount((sb_1, b_2)) =$ $\begin{cases} 1 + basecount((s, b_2)) & \text{when } b_1 = b_2 \\ basecount((s, b_2)) & \text{when } b_1 \neq b_2 \end{cases}$	$basecount((\mathbf{ACU}, \mathbf{C})) =$
“2 to the power of”	$\mathbb{N}$	$\mathbb{N}$	Basis Step: $2^0 = 1$ Recursive Step: If $n \in \mathbb{N}, 2^{n+1} =$	
“ b to the power of i ”	$\mathbb{Z}^+ \times \mathbb{N}$	$\mathbb{N}$	Basis Step: $b^0 = 1$ Recursive Step: If $i \in \mathbb{N}, b^{i+1} = b \cdot b^i$	

$2^0 = 1$	$2^1 = 2$	$2^2 = 4$	$2^3 = 8$	$2^4 = 16$	$2^5 = 32$	$2^6 = 64$	$2^7 = 128$	$2^8 = 256$	$2^9 = 512$	$2^{10} = 1024$
-----------	-----------	-----------	-----------	------------	------------	------------	-------------	-------------	-------------	-----------------

Defining sets

To define sets:

To define a set using **roster method**, explicitly list its elements. That is, start with $\{$ then list elements of the set separated by commas and close with $\}$.

To define a set using **set builder definition**, either form “The set of all x from the universe U such that x is ...” by writing

$$\{x \in U \mid ...x...\}$$

or form “the collection of all outputs of some operation when the input ranges over the universe U ” by writing

$$\{...x... \mid x \in U\}$$

We use the symbol $\in$ as “is an element of” to indicate membership in a set.

Example sets: For each of the following, identify whether it's defined using the roster method or set builder notation and give an example element.

Can we infer the data type of the example element from the notation?

$$\{-1, 1\}$$

$$\{0, 0\}$$

$$\{-1, 0, 1\}$$

$$\{(x, x, x) \mid x \in \{-1, 0, 1\}\}$$

$$\{\}$$

$$\{x \in \mathbb{Z} \mid x \geq 0\}$$

$$\{x \in \mathbb{Z} \mid x > 0\}$$

$$\{\smile, \odot\}$$

$$\{\mathbf{A}, \mathbf{C}, \mathbf{U}, \mathbf{G}\}$$

$$\{\mathbf{AUG}, \mathbf{UAG}, \mathbf{UGA}, \mathbf{UAA}\}$$

Rna motivation

RNA is made up of strands of four different bases that encode genomic information in specific ways. The bases are elements of the set $B = \{\mathbf{A}, \mathbf{C}, \mathbf{U}, \mathbf{G}\}$. Strands are ordered nonempty finite sequences of bases.

Formally, to define the set of all RNA strands, we need more than roster method or set builder descriptions.

Set recursive examples

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

Basis Step:

Recursive Step:

Examples:

Definition The set of all integers $\mathbb{Z}$ is defined (recursively) by:

Basis Step:

Recursive Step:

Examples:

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:

Definition The set of bitstrings (strings of 0s and 1s) is defined (recursively) by:

Basis Step:

Recursive Step:

Notation: We call the set of bitstrings $\{0, 1\}^*$ and we say this is the set of all strings over $\{0, 1\}$.

Examples: