Netflix intro

What data should we encode about each Netflix account holder to help us make effective recommendations?

In machine learning, clustering can be used to group similar data for prediction and recommendation. For example, each Netflix user's viewing history can be represented as a n-tuple indicating their preferences about movies in the database, where n is the number of movies in the database. People with similar tastes in movies can then be clustered to provide recommendations of movies for one another. Mathematically, clustering is based on a notion of distance between pairs of n-tuples.

Set operations

To define a set we can use the roster method, set builder notation, a recursive definition, and also we can apply a set operation to other sets.

New! Cartesian product of sets and **set-wise concatenation of sets of strings**

Definition: 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 $b \in B$

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

Fill in the missing entries in the table:

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

Defining functions

New! Defining functions A function is defined by its (1) domain, (2) codomain, and (3) rule assigning each element in the domain exactly one element in the codomain.

The domain and codomain are nonempty sets.
The rule can be depicted as a table, formula, or English description.

Examples:

Definition (Of a function, recursively) A 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 domain of *rnalen* is __________________. The codomain of *rnalen* is ____________________.

$rnalen(\texttt{ACU}) =$ ____

Extra example: A function *basecount* that computes the number of a given base b appearing in a RNA strand s is defined recursively: *fill in codomain and sample function applications*

$$basecount : S \times B \quad \to$$

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

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

$basecount(\texttt{ACU}, \texttt{A}) =$ ____

$basecount(\texttt{ACU}, \texttt{G}) =$ ____

Defining functions recursively

Definition (Of a function, recursively) A 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 domain of *rnalen* is ____________________. The codomain of *rnalen* is _____________________.

$rnalen(\texttt{ACU}) =$ ____

Extra example: A function *basecount* that computes the number of a given base b appearing in a RNA strand s is defined recursively: *fill in codomain and sample function applications*

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

$basecount(\texttt{ACU}, \texttt{A}) =$ ____

$basecount(\texttt{ACU}, \texttt{G}) =$ ____