Definitions

Term	Notation Example(s)	We say in English
sequence	x_1, \ldots, x_n	A sequence x_1 to x_n
summation	x_1, \dots, x_n $\sum_{i=1}^n x_i \text{ or } \sum_{i=1}^n x_i$	The sum of the terms of the sequence x_1 to x_n
all reals	\mathbb{R}	The (set of all) real numbers (numbers on the number line)
all integers	\mathbb{Z}	The (set of all) integers (whole numbers including negatives, zero, and positives)
all positive integers	\mathbb{Z}^+	The (set of all) strictly positive integers
all natural numbers	N	The (set of all) natural numbers. Note : we use the convention that 0 is a natural number.
piecewise rule definition function application	$f(x) = \begin{cases} x & \text{if } x \ge 0 \\ -x & \text{if } x < 0 \end{cases}$ $f(7)$ $f(z)$ $f(g(z))$	Define f of x to be x when x is nonnegative and to be $-x$ when x is negative f of f or f applied to f or the image of f under f of f or f applied to f or the image of f under f of f of f of f of f of f applied to the result of f applied to f
absolute value square root	$\begin{array}{c} -3 \\ \sqrt{9} \end{array}$	The absolute value of -3 The non-negative square root of 9

Data types

Term	Examples:		
	(add additional	examples from class)	
set	$7 \in \{43, 7, 9\}$	$2 \notin \{43, 7, 9\}$	
unordered collection of elements			
repetition doesn't matter			
Equal sets agree on membership of all elements			
n-tuple			
ordered sequence of elements with n "slots" $(n > 0)$			
repetition matters, fixed length			
Equal n-tuples have corresponding components equal			

string

ordered finite sequence of elements each from specified set repetition matters, arbitrary finite length Equal strings have same length and corresponding characters equal

$Special\ cases:$

When n = 2, the 2-tuple is called an **ordered pair**.

A string of length 0 is called the **empty string** and is denoted λ .

A set with no elements is called the **empty set** and is denoted $\{\}$ or \emptyset .

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.

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 X and Y be sets. The **Cartesian product** of X and Y, denoted $X \times Y$, is the set of all ordered pairs (x, y) where $x \in X$ and $y \in Y$

$$X \times Y = \{(x, y) \mid x \in X \text{ and } y \in Y\}$$

Definition: Let X and Y be sets of strings over the same alphabet. The **set-wise concatenation** of X and Y, denoted $X \circ Y$, is the set of all results of string concatenation xy where $x \in X$ and $y \in Y$

$$X \circ Y = \{xy \mid x \in X \text{ and } y \in Y\}$$

Pro-tip: the meaning of writing one element next to another like xy depends on the data-types of x and y. When x and y are strings, the convention is that xy is the result of string concatenation. When x and y are numbers, the convention is that xy is the result of multiplication. This is (one of the many reasons) why is it very important to declare the data-type of variables before we use them.

Fill in the missing entries in the table:

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)		
$\{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{U}\}\circ\{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{U}\}$			
	GGGG		