

## Ratings encoding

In the table below, each row represents a user's ratings of movies: ✓ (check) indicates the person liked the movie, ✗ (x) that they didn't, and • (dot) that they didn't rate it one way or another (neutral rating or didn't watch). Can encode these ratings numerically with 1 for ✓ (check), −1 for ✗ (x), and 0 for • (dot).

| Person | Fyre | Frozen II | Picard | Ratings written as a 3-tuple |
|--------|------|-----------|--------|------------------------------|
| $P_1$  | ✗    | •         | ✓      |                              |
| $P_2$  | ✓    | ✓         | ✗      |                              |
| $P_3$  | ✓    | ✓         | ✓      |                              |
| $P_4$  | •    | ✗         | ✓      |                              |

## Definitions

| Term                      | Notation  | Example(s) | We say in English ...   |
|---------------------------|---|------------|---|
| sequence                  | $x_1, \dots, x_n$   |            | A sequence $x_1$ to $x_n$   |
| summation                 | $\sum_{i=1}^n x_i$ 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       | $\mathbb{N}$  |            | The (set of all) natural numbers. <b>Note:</b> we use the convention that 0 is a natural number.  |
| piecewise rule definition | $f(x) = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{cases}$ |            | Define $f$ of $x$ to be $x$ when $x$ is nonnegative and to be $-x$ when $x$ is negative   |
| function application      | $f(7)$<br>$f(z)$<br>$f(g(z))$   |            | $f$ of 7 <b>or</b> $f$ applied to 7 <b>or</b> the image of 7 under $f$<br>$f$ of $z$ <b>or</b> $f$ applied to $z$ <b>or</b> the image of $z$ under $f$<br>$f$ of $g$ of $z$ <b>or</b> $f$ applied to the result of $g$ applied to $z$ |
| absolute value            | $ -3 $  |            | The absolute value of $-3$  |
| square root               | $\sqrt{9}$  |            | The non-negative square root of 9   |

# Data types

| Term   | Examples:<br>(add additional examples from class) |
|--|---|
| <b>set</b><br>unordered collection of elements<br><i>repetition doesn't matter</i><br><i>Equal sets agree on membership of all elements</i>  | $7 \in \{43, 7, 9\}$ $2 \notin \{43, 7, 9\}$      |
| <b><math>n</math>-tuple</b><br>ordered sequence of elements with $n$ “slots” ( $n > 0$ )<br><i>repetition matters, fixed length</i><br><i>Equal <math>n</math>-tuples have corresponding components equal</i>  |   |
| <b>string</b><br>ordered finite sequence of elements each from specified set<br><i>repetition matters, arbitrary finite length</i><br><i>Equal strings have same length and corresponding characters equal</i> |   |

*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  $\lambda$ .

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

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

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

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

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.

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

$$\{0, 0\}$$

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

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

$$\{\}$$

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

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

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

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

## Defining functions ratings

Recall our representation of Netflix users' ratings of movies as  $n$ -tuples, where  $n$  is the number of movies in the database. Each component of the  $n$ -tuple is  $-1$  (didn't like the movie),  $0$  (neutral rating or didn't watch the movie), or  $1$  (liked the movie).

Consider the ratings  $P_1 = (-1, 0, 1)$ ,  $P_2 = (1, 1, -1)$ ,  $P_3 = (1, 1, 1)$ ,  $P_4 = (0, -1, 1)$

Which of  $P_1$ ,  $P_2$ ,  $P_3$  has movie preferences most similar to  $P_4$ ?

One approach to answer this question: use **functions** to define distance between user preferences.

For example, consider the function  $d_0$  :

given by

→

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

*Extra example:* A new movie is released, and  $P_1$  and  $P_2$  watch it before  $P_3$ , and give it ratings;  $P_1$  gives ✓ and  $P_2$  gives ✗. Should this movie be recommended to  $P_3$ ? Why or why not?

*Extra example:* Define a new function that could be used to compare the 4-tuples of ratings encoding movie preferences now that there are four movies in the database.