Sets

- **Set:** this is a basic, undefined word in mathematics. Other things are defined in terms of it, but it is not defined in terms of other mathematical words. Maybe imagine any collection of objects, which can be physical objects, numbers, or other sets. I sometimes think of sets as paper grocery bags, with the paper bag denoted by $\{$ and $\}$.
- It helps to think of finite sets as **unordered** lists with repetitions allowed.
- **Member of a set:** $a \in B$ is the notation for "a is a member of B". This is another basic, undefined concept in mathematics.
- **Equality of Sets:** Let A and B be sets. $A = B$ if and only if they have the same members:

$$A = B \quad \Leftrightarrow \quad \text{for all } x,\ x \in A \text{ if and only if } x \in B$$

The Subset Relation

Let A and B be sets.

- A **is a subset of** B, $A \subset B$**:** every member of A is also a member of B. Other books use $A \subseteq B$ for this concept. Here $A \subset B$ even when $A = B$.
  - If A is the set of humans and B is the set of mammals, then $A \subset B$.
  - $A = B$ if and only if, $A \subset B$ and $B \subset A$. This is a common way to prove that two sets are equal—one paragraph argues $A \subset B$ and a second paragraph argues that $B \subset A$.
  - If C is a set with $A \subset B$ and $B \subset C$, then $A \subset C$. Example: A is the set of human beings, B is the set of mammals, and C is the set of vertebrates.

Computer Representation of Sets

This is a concept that computers handle badly. Computers can handle lists of elements, but computers must hold them in some specified order. So the concept of a set is difficult to code in computers because there are too many right ways to list all the elements of a finite set. All these ways are different in the computer **but all the same as sets**.

- To list 10 distinct elements without repetition, there $10! = 3,628,800$ different representations.
- For 20 distinct elements, there are more than 2.4×10^{18} ways to list them without repetition.
- For programming purposes, maybe always list elements in some “alphabetic" order and remove repetitions. This makes comparison of two sets easier, but imposes a large overhead of always sorting (and removing duplicates) before final storage.

Specifying Sets (three ways)

- **List the members:** For example, let $A = \{1, 3, 7\}$. Repetition does not matter and the order of the elements does not matter:

$$A = \{1, 1, 3, 3, 3, 7\} \quad \text{and} \quad A = \{7, 1, 3\}$$

  are both true about A.

- **Specify the members with a formula:** Suppose that A is a set and $\phi(x)$ is some sentence about x that makes sense for any member x in A. Importantly, for each $x \in A$, $\phi(x)$ is either true or false but not both. Then you can form a new set B that picks out from A the members x where $\phi(x)$ is true:

$$B = \{\, x \in A \,:\, \phi(x) \,\}$$

  For example, A could be the set of human beings and $\phi(x)$ could be "x was born in China on a Tuesday".

Specifying sets (cont'd)

Some famous sets of numbers can be used without explanation.

- $\mathbb{Z}$: this is the set of integers, including 0 and negative integers: think of the vague notation $\{\ldots, -2, -1, 0, 1, 2, \ldots\}$ This cannot be represented in a finite computer.
- $\mathbb{Q}$: the set of rational numbers. It consists of r such that, for some integers a and b with $b \neq 0$, $r = a/b$.
- $\mathbb{R}$: the set of real numbers. Truly vast. Think of all infinite decimals $x = a.d_1 d_2 d_3 \ldots$ where a is an integer and each d_i is from $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.
- $\mathbb{C}$: the set of complex numbers. It consists of numbers $r + i\,s$ for any real r and s and $i = \sqrt{-1}$.
- **Computer representation of real numbers:** usually base 2. A real x is $a.d_1 d_2 \ldots d_n$ for n around 50. Each $d_i \in \{0, 1\}$ and the place value of d_i is 2^{-i}.

Experiment with Numbers in Matlab (not a homework)

- Do this in Matlab:
  » *myID=(put here your student ID without hyphens)*
  » *format hex*
  » *myID*
  The Matlab command *f*ormat hex will display exactly what is stored, in base 16. Can you explain how every character of the hex (base 16) relates to your student ID?
- Do this in Matlab, for various values of n:
  » *testNUMBER=2∧(-n)*
  » *z=1+testNUMBER*
  » *z-1*
  What is the first positive n where $z - 1$ equals 0?