

# Functions

## Part One

# Outline for Today

- ***What is a Function?***
  - It's more nuanced than you might expect.
- ***Domains and Codomains***
  - Where functions start, and where functions end.
- ***Defining a Function***
  - Expressing transformations compactly.
- ***Special Classes of Functions***
  - Useful types of functions you'll encounter IRL.
- ***Proofs on First-Order Definitions***
  - A key skill.

What is a function?

## High-School Algebra

## Computer Programming

### "Nice" Functions

$$f(x) = x^3 + 3x + 1$$

$$g(x) = \sin^3 x - 3$$

$$h(x) = e^{x^2} \cdot e^{e^x}$$

```
string withoutVowels(string input) {  
    string result;  
    for (char ch: input) {  
        if (!isVowel(ch)) {  
            result += ch;  
        }  
    }  
    return result;  
}
```

### "Tricksy" Functions

$$f(x) = \frac{x+2}{x+1}$$

$$g(x) = \tan x$$

$$h(x) = \ln x^3$$

```
int yikes(string input) {  
    while (true) { }  
}  
void oops(int input) {  
    cout << "Hi, mom!" << endl;  
}  
int ouch(int input) {  
    return input / 0;  
}  
double hmmm(double input) {  
    return input + randomInteger();  
}
```

“All models are wrong. Some are useful.”

- George Box

## ***High-Level Intuition:***

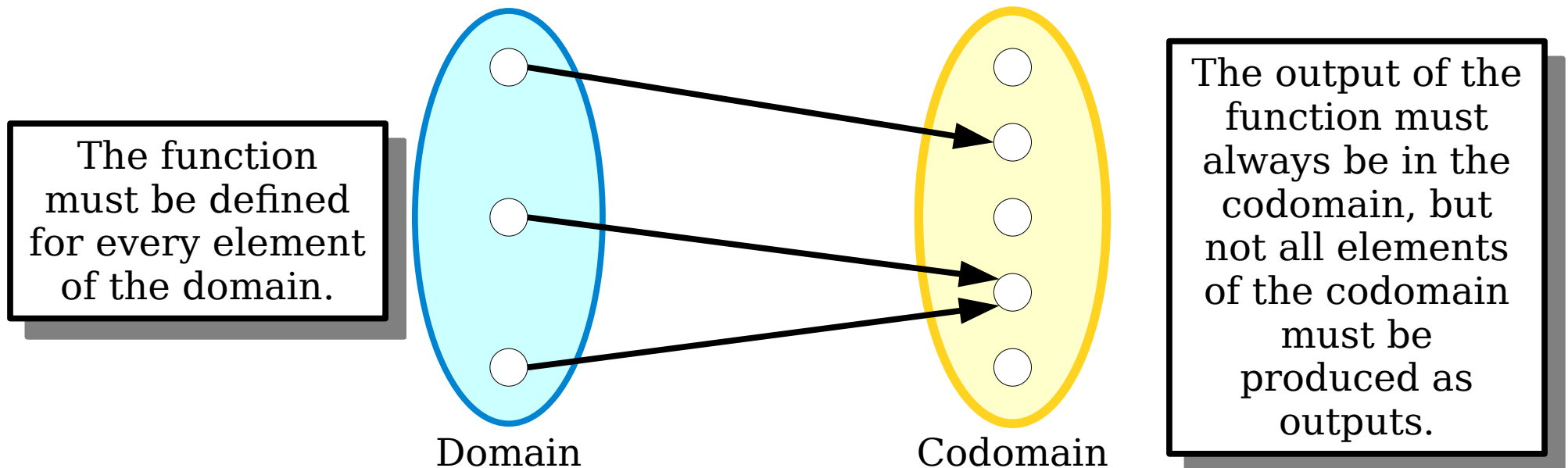
A function is an object  $f$  that takes in exactly one input  $x$  and produces exactly one output  $f(x)$ .



(This is not definition. It's just to help you build and intuition.)

# Domains and Codomains

- Every function  $f$  has two sets associated with it: its **domain** and its **codomain**.
- A function  $f$  can only be applied to elements of its domain. For any  $x$  in the domain,  $f(x)$  belongs to the codomain.



# Domains and Codomains

- Every function  $f$  has two sets associated with it: its **domain** and its **codomain**.
- A function  $f$  can only be applied to elements of its domain. For any  $x$  in the domain,  $f(x)$  belongs to the codomain.

The **domain** of this function is  $\mathbb{R}$ . Any real number can be provided as input.

The **codomain** of this function is  $\mathbb{R}$ . Everything produced is a real number, but not all real numbers can be produced.

```
double absoluteValueOf(double x) {  
    if (x >= 0) {  
        return x;  
    } else {  
        return -x;  
    }  
}
```

# Domains and Codomains

- Every function  $f$  has two sets associated with it: its **domain** and its **codomain**.
- A function  $f$  can only be applied to elements of its domain. For any  $x$  in the domain,  $f(x)$  belongs to the codomain.

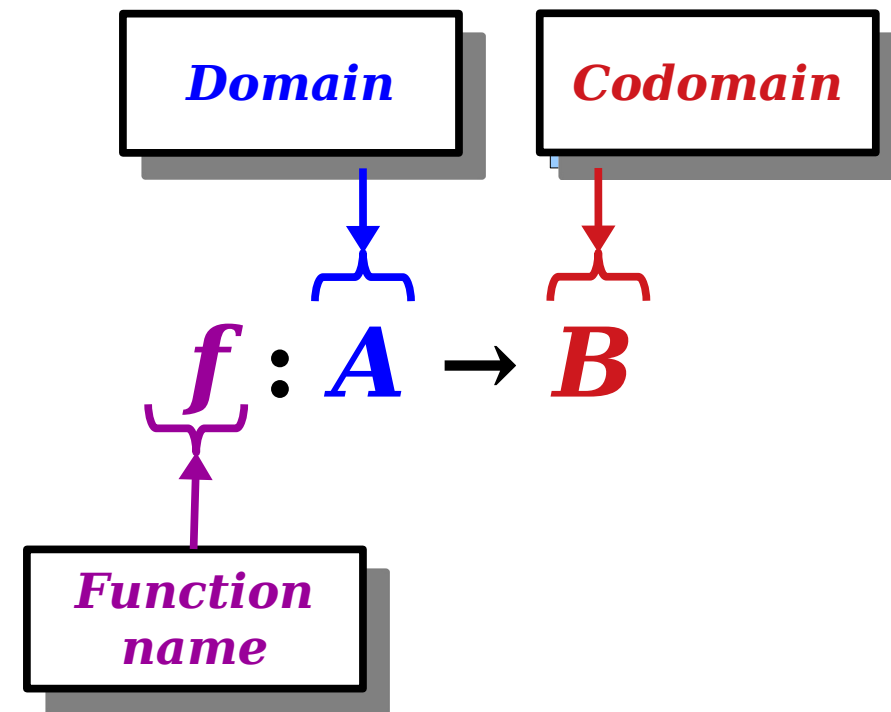
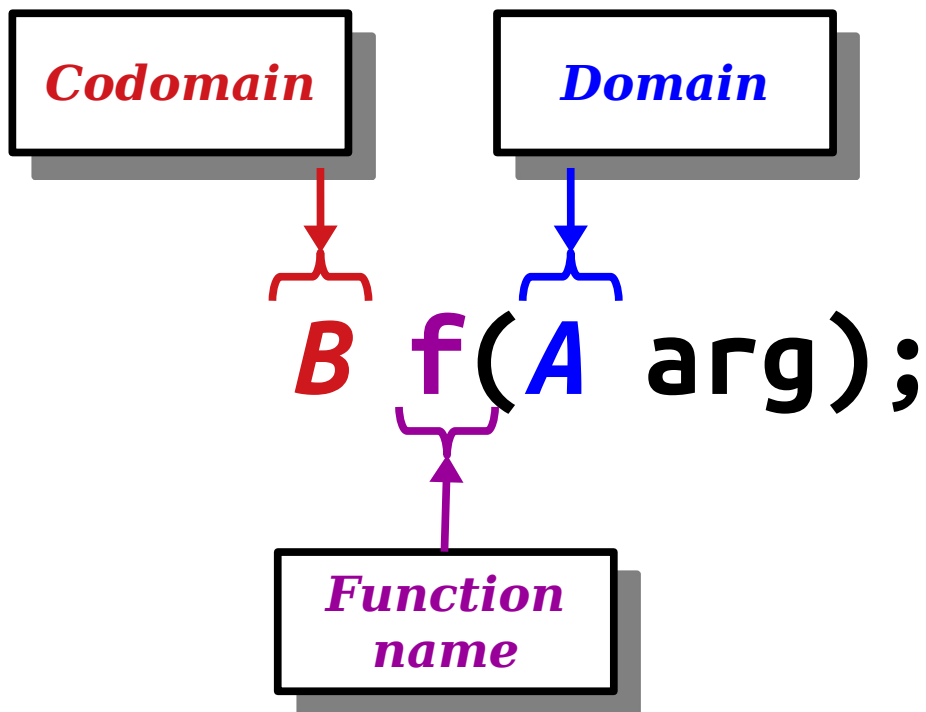
The **domain** of this function is the set of all strings. Any string can be an input.

The **codomain** of this function is the set of all strings. Every output of this function is a string, but not all strings can be produced.

```
string withoutVowels(string input) {  
    string result;  
    for (char ch: input) {  
        if (!isVowel(ch)) {  
            result += ch;  
        }  
    }  
    return result;  
}
```

# Domains and Codomains

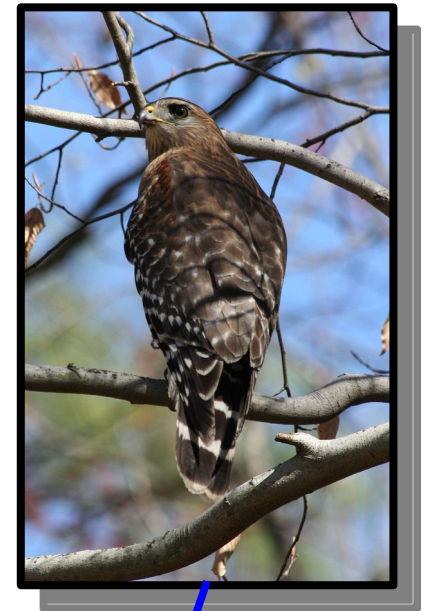
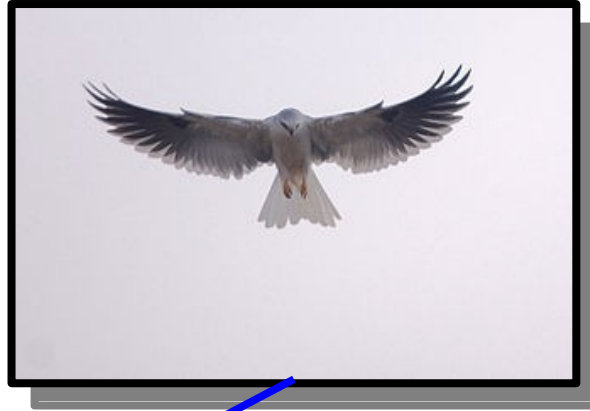
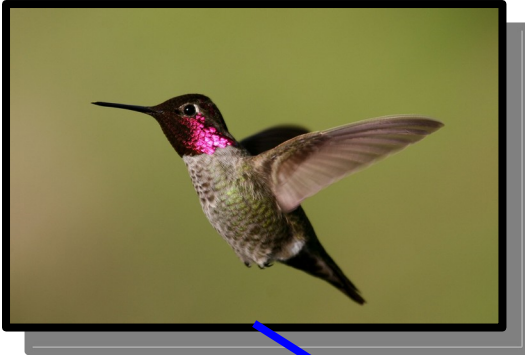
- If  $f$  is a function whose domain is  $A$  and whose codomain is  $B$ , we write  $f : A \rightarrow B$ .
- Think of this like a “function prototype” in C++.



# Defining Functions

# Defining Functions

- To define a function, you need to
  - specify the domain,
  - specify the codomain, and
  - give a **rule** used to evaluate the function.
- All three pieces are necessary.
  - We need to domain to know what the function can be applied to.
  - We need to codomain to know what the output space is.
  - We need the rule to be able to evaluate the function.
- There are many ways to do this. Let's go over a few examples.



*White-Tailed  
Kite*

*Anna's  
Hummingbird*

*Red-Shouldered  
Hawk*

Functions can be defined as a ***picture***.  
Draw the domain and codomain explicitly.  
Then, add arrows to show the outputs.

$$f : \mathbb{Z} \rightarrow \mathbb{Z}, \text{ where}$$
$$f(x) = x^2 + 3x - 15$$

---

Functions can be defined as a **rule**.  
Be sure to explicitly state what the  
domain and codomain are!

$f : \mathbb{Z} \rightarrow \mathbb{N}$ , where

$$f(n) = \begin{cases} n & \text{if } n \geq 0 \\ -n & \text{if } n \leq 0 \end{cases}$$

---

Some rules are given ***piecewise***. We select which rule to apply based on the conditions on the right. (Just make sure at least one condition applies and that all applicable conditions give the same result!)

$f : \mathbb{R} \rightarrow \mathbb{R}$  defined as

$$f(x) = \frac{x+2}{x+1}$$

This expression isn't defined when  $x = -1$ , so  $f$  isn't defined over its full domain. We therefore don't consider it to be a function.

Is this a function?

$f : \mathbb{N} \rightarrow \mathbb{R}$  defined as

$$f(x) = \frac{x+2}{x+1}$$

Yep, it's a function! Every natural number maps to some real number.

---

Is this a function?

# The Official Rules for Functions

- Formally speaking, we say that  $f : A \rightarrow B$  if the following two rules hold.
- First,  $f$  must obey its domain/codomain rules:

$$\forall a \in A. \exists b \in B. f(a) = b$$

*(“Every input in  $A$  maps to some output in  $B$ .”)*

- Second,  $f$  must be deterministic:

$$\forall a_1 \in A. \forall a_2 \in A. (a_1 = a_2 \rightarrow f(a_1) = f(a_2))$$

*(“Equal inputs produce equal outputs.”)*

- If you’re ever curious about whether something is a function, look back at these rules and check! For example:
  - Can a function have an empty domain?
  - Can a function have an empty codomain?

**Time-Out for Announcements!**

# Gradescope Tagging

- When you upload a PDF to Gradescope, please make sure to tag the pages that have your problem answers on them.
- The **altruistic** reason: if you don't do this, the TAs have to do it for you, and across 130+ submissions that adds up to hours of extra work.
- The **selfish** reason: if you don't tag the page containing a problem, Gradescope marks it as though you didn't submit it, and the TAs might give you no points because they thought you didn't submit anything.
- You can tag pages after you submit, so if you submit and then realize you forgot to tag things you can always go back and fix it.

# Gradescope with Partners

- ***Reminder:*** If you work with a partner on a problem set, then...
  - ... make sure that both your names are listed on the solution on Gradescope. Otherwise, one of you won't get credit!
  - ... make sure you make a single joint submission. Submitting two copies of the solution slows down grading.
- If you forgot to add your partner, no worries! You can add them retroactively on Gradescope. Please do that soon.

# Problem Set One Solutions

- We've just posted solutions to Problem Set One. They're linked from the main PS1 page.
- We recommend you read over our solution set before finishing PS2.
  - You'll get to see examples of polished written proofs.
  - Each problem has a "Why We Asked This Question" section, which gives some context.
  - We may have solved the problem differently than you, and this will give you more perspectives to use.
- We'll aim to have PS1 graded and returned tomorrow night.

Back to CS103!

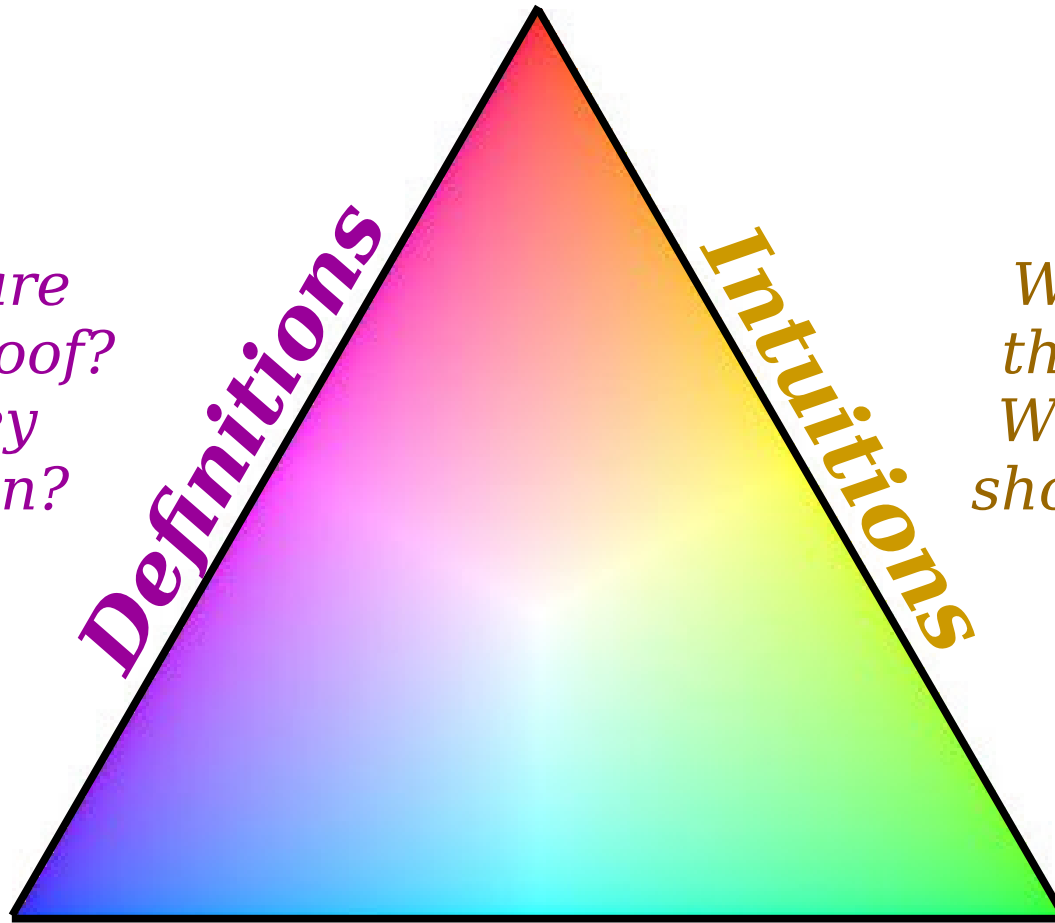
# Special Types of Functions

*What terms are  
used in this proof?  
What do they  
formally mean?*

***Definitions***

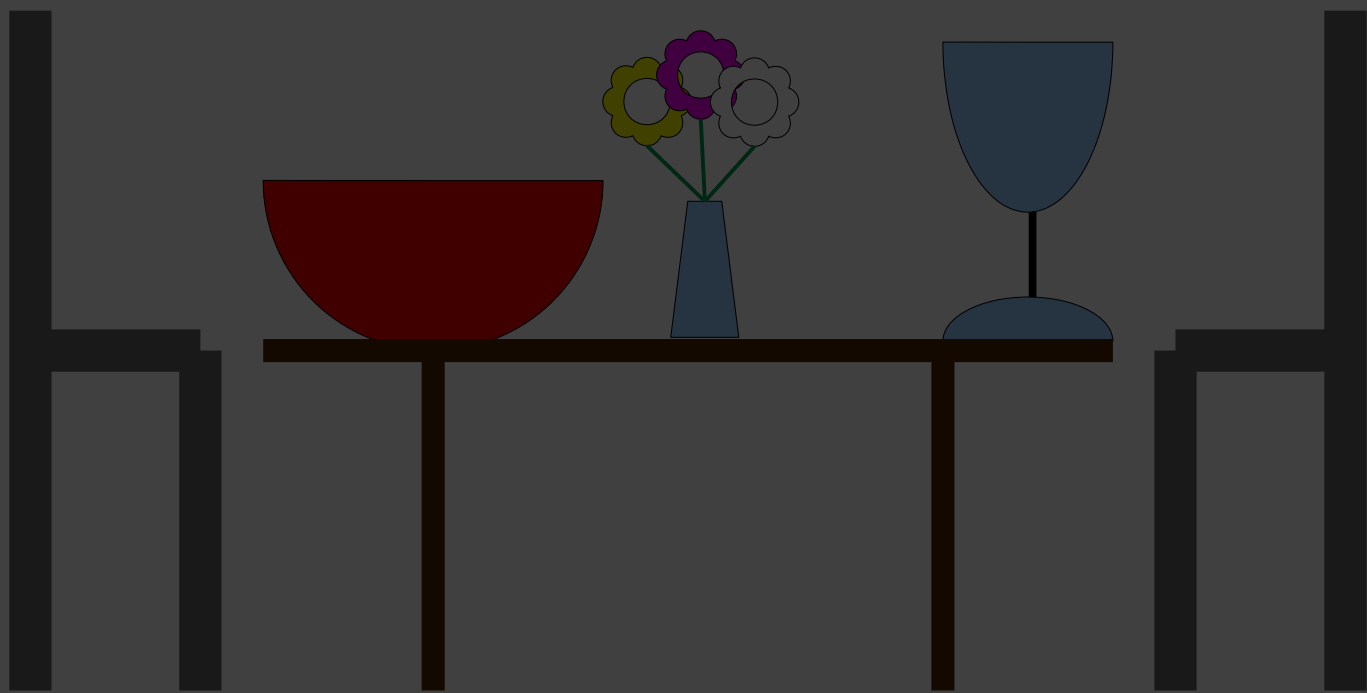
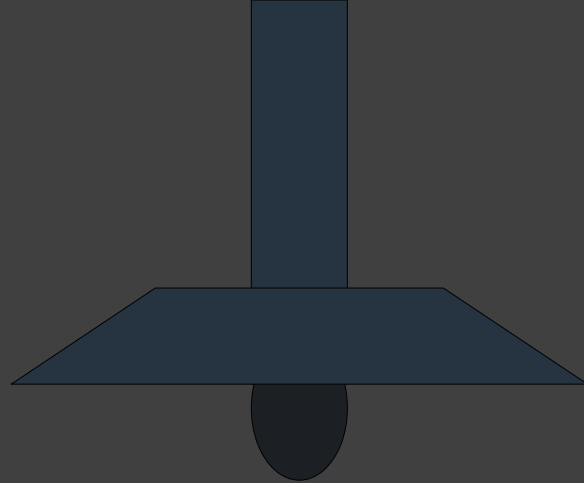
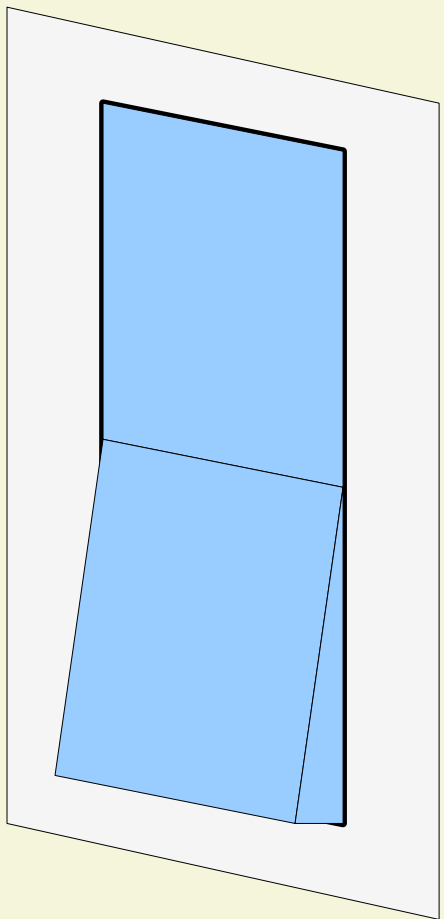
***Intuitions***

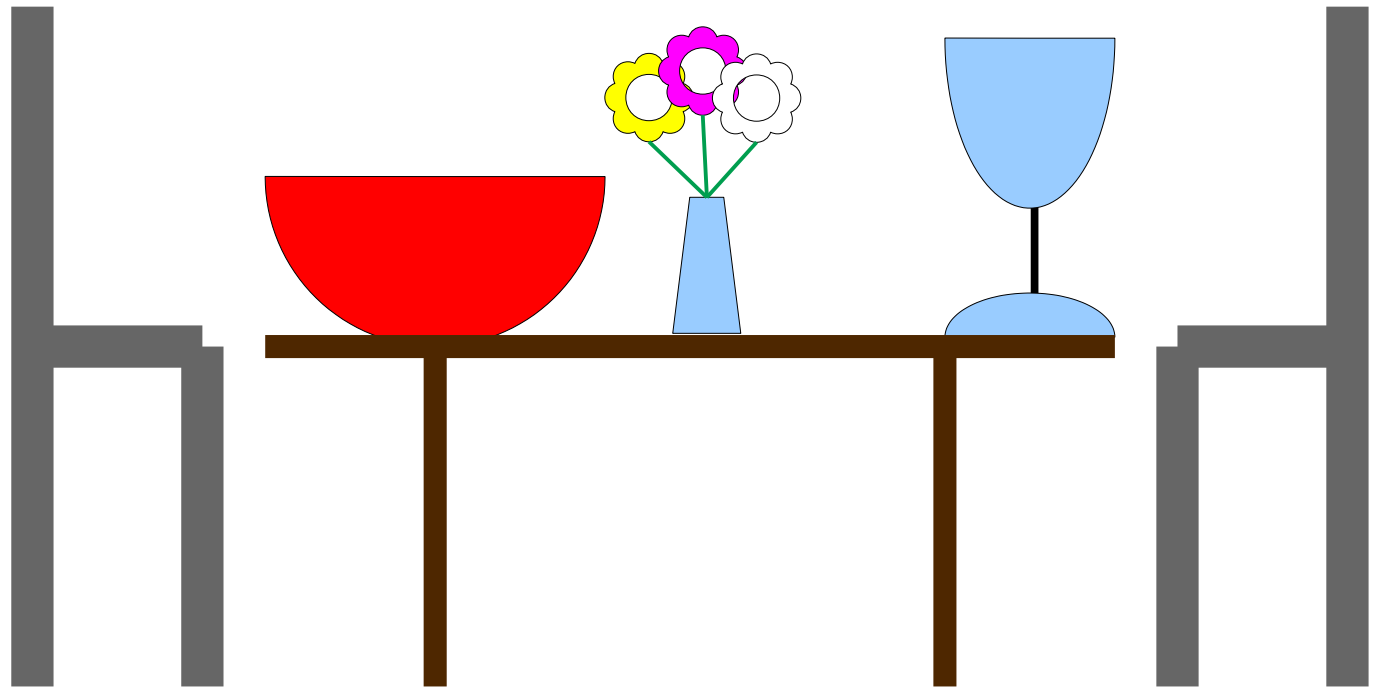
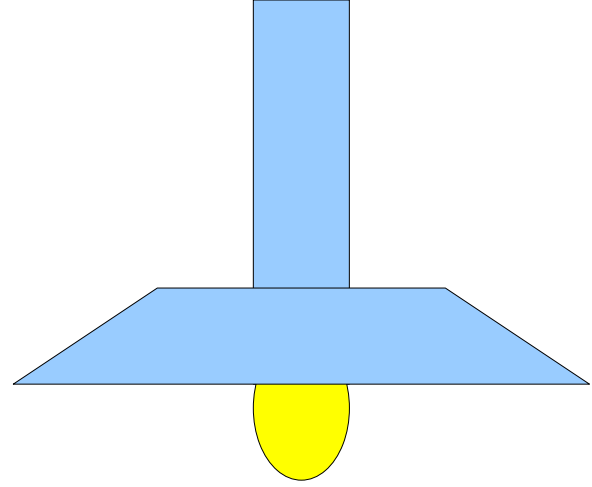
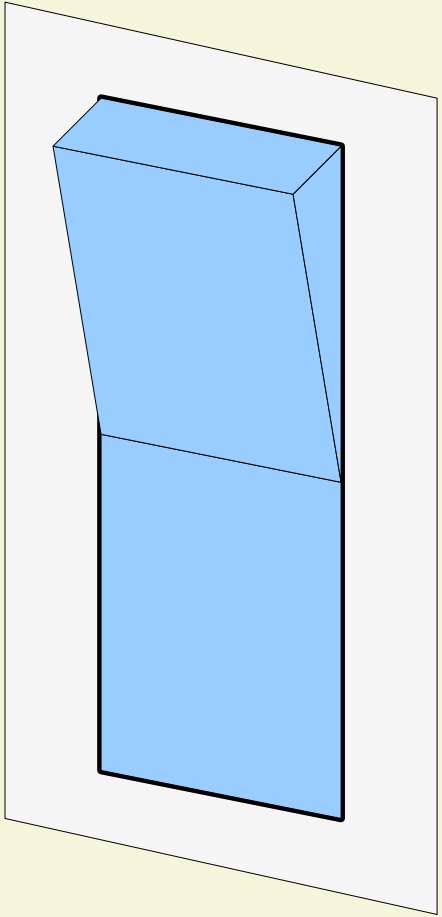
*What does this  
theorem mean?  
Why, intuitively,  
should it be true?*



***Conventions***

*What is the standard  
format for writing a proof?  
What are the techniques  
for doing so?*





# Undoing by Doing Again

- Some operations invert themselves. For example:
  - Flipping a switch twice is the same as not flipping it at all.
  - In first-order logic,  $\neg\neg A$  is equivalent to  $A$ .
  - In algebra,  $-(-x) = x$ .
  - In set theory,  $(A \Delta B) \Delta B = A$ . (*Yes, really!*)
- Operations with these properties are surprisingly useful in CS theory and come up in a bunch of contexts.
  - Storing compressed approximations of sets (XOR filters).
  - Building encryption systems (symmetric block ciphers).
  - Transmitting a large file to multiple receivers (fountain codes).

# Involutions

- A function  $f : A \rightarrow A$  from a set back to itself is called an ***involution*** if the following first-order logic statement is true about  $f$ :

$$\forall x \in A. f(f(x)) = x.$$

*(“Applying  $f$  twice is equivalent to not applying  $f$  at all.”)*

- Involutions have lots of interesting properties. Let’s explore them and see what we can find.

# Involutions

- Which of the following are involutions?
  - $f : \mathbb{Z} \rightarrow \mathbb{Z}$  defined as  $f(x) = x$ . *Yep!*
  - $g : \mathbb{Z} \rightarrow \mathbb{Z}$  defined as  $g(x) = -x$ . *Yep!*
  - $h : \mathbb{R} \rightarrow \mathbb{R}$  defined as  $h(x) = 1/x$ . *Not a function!*
  - $p : \mathbb{N} \rightarrow \mathbb{N}$  defined as follows: *Yep!*

$$p(n) = \begin{cases} n+1 & \text{if } n \text{ is even} \\ n-1 & \text{if } n \text{ is odd} \end{cases}$$

---

A function  $f : A \rightarrow A$  is called an ***involution*** if the following first-order logic statement is true about  $f$ :

$$\forall x \in A. f(f(x)) = x.$$

# Proofs on Involutions

**Theorem:** The function  $f : \mathbb{Z} \rightarrow \mathbb{Z}$  defined as

$$f(n) = \begin{cases} n+1 & \text{if } n \text{ is even} \\ n-1 & \text{if } n \text{ is odd} \end{cases}$$

is an involution.

**Proof:**

What does it mean for  $f$  to be an involution?

$$\forall n \in \mathbb{Z}. f(f(n)) = n.$$

Therefore, we'll have the reader pick some  $n \in \mathbb{Z}$ , then argue that  $f(f(n)) = n$ .

**Theorem:** The function  $f : \mathbb{Z} \rightarrow \mathbb{Z}$  defined as

$$f(n) = \begin{cases} n+1 & \text{if } n \text{ is even} \\ n-1 & \text{if } n \text{ is odd} \end{cases}$$

is an involution.

**Proof:** Pick some  $n \in \mathbb{Z}$ . We need to show that  $f(f(n)) = n$ . To do so, we consider two cases.

*Case 1:*  $n$  is even. Then  $f(n) = n+1$ , which is odd. This means that  $f(f(n)) = f(n+1) = (n+1) - 1 = n$ .

*Case 2:*  $n$  is odd. Then  $f(n) = n - 1$ , which is even. Then we see that  $f(f(n)) = f(n - 1) = (n - 1) + 1 = n$ .

In either case, we see that  $f(f(n)) = n$ , which is what we need to show. ■

This proof contains no first-order logic syntax (quantifiers, connectives, etc.). It's written in plain English, just as usual.

**Theorem:** The function  $f : \mathbb{N} \rightarrow \mathbb{N}$  defined as  $f(n) = n^2$  is not an involution.

What does it mean for  $f$  to be an involution?

$$\forall n \in \mathbb{N}. f(f(n)) = n.$$

What is the negation of this statement?

$$\begin{aligned} &\neg \forall n \in \mathbb{N}. f(f(n)) = n \\ &\exists n \in \mathbb{N}. \neg (f(f(n)) = n) \\ &\exists n \in \mathbb{N}. f(f(n)) \neq n \end{aligned}$$

Therefore, we need to pick some concrete choice of  $n$  such that  $f(f(n)) \neq n$ .

**Theorem:** The function  $f : \mathbb{N} \rightarrow \mathbb{N}$  defined as  $f(n) = n^2$  is not an involution.

**Proof:** We need to show that there is some  $n \in \mathbb{N}$  where  $f(f(n)) \neq n$ .

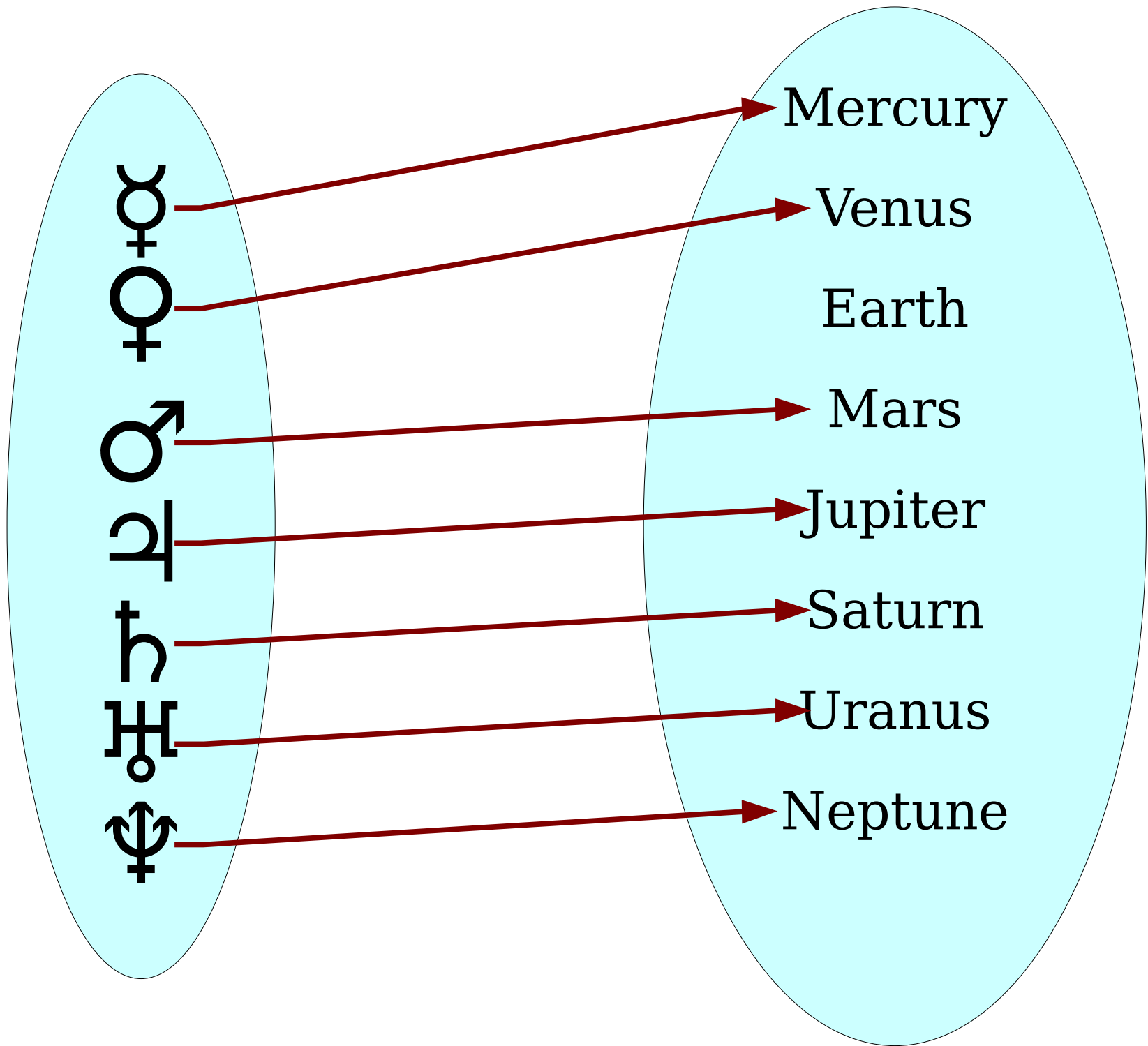
Pick  $n = 2$ . Then

$$\begin{aligned} f(f(n)) &= f(f(2)) \\ &= f(4) \\ &= 16, \end{aligned}$$

which means that  $f(f(n)) \neq n$ , as required. ■

This proof contains no first-order logic syntax (quantifiers, connectives, etc.). It's written in plain English, just as usual.

# Another Class of Functions



# Injective Functions

- A function  $f : A \rightarrow B$  is called **injective** (or **one-to-one**) if the following statement is true about  $f$ :

$$\forall a_1 \in A. \forall a_2 \in A. (a_1 \neq a_2 \rightarrow f(a_1) \neq f(a_2))$$

*(“If the inputs are different, the outputs are different.”)*

- The following first-order definition is equivalent (*why?*) and is often useful in proofs.

$$\forall a_1 \in A. \forall a_2 \in A. (f(a_1) = f(a_2) \rightarrow a_1 = a_2)$$

*(“If the outputs are the same, the inputs are the same.”)*

- A function with this property is called an **injection**.
- How does this compare to our second rule for functions?

# Injective

- Let  $S$  be the set of all CS103 students. Which of the following are injective?
  - $f: S \rightarrow \mathbb{N}$  where  $f(x)$  is  $x$ 's Stanford ID number.
  - $g: S \rightarrow C$ , where  $C$  is the set of all continents and  $g(x)$  is  $x$ 's continent of birth.
  - $h: S \rightarrow N$ , where  $N$  is the set of all given (first) names, where  $h(x)$  is  $x$ 's given (first) name.

---

$f: A \rightarrow B$  is **injective** when either equivalent statement is true:

$$\forall x_1 \in A. \forall x_2 \in A. (x_1 \neq x_2 \rightarrow f(x_1) \neq f(x_2))$$

$$\forall x_1 \in A. \forall x_2 \in A. (f(x_1) = f(x_2) \rightarrow x_1 = x_2)$$

# Proofs on Injections

# Injective Functions

**Theorem:** Let  $f : \mathbb{N} \rightarrow \mathbb{N}$  be defined as  $f(n) = 2n + 7$ .  
Then  $f$  is injective.

**Proof:**

What does it mean for the function  $f$  to be injective?

$$\forall n_1 \in \mathbb{N}. \forall n_2 \in \mathbb{N}. ( f(n_1) = f(n_2) \rightarrow n_1 = n_2 )$$

$$\forall n_1 \in \mathbb{N}. \forall n_2 \in \mathbb{N}. ( n_1 \neq n_2 \rightarrow f(n_1) \neq f(n_2) )$$

Therefore, we'll pick arbitrary  $n_1, n_2 \in \mathbb{N}$ ,  
assume  $f(n_1) = f(n_2)$ , then prove that  $n_1 = n_2$ .

# Injective Functions

**Theorem:** Let  $f : \mathbb{N} \rightarrow \mathbb{N}$  be defined as  $f(n) = 2n + 7$ .  
Then  $f$  is injective.

**Proof:** Consider any  $n_1, n_2 \in \mathbb{N}$  where  $f(n_1) = f(n_2)$ . We will prove that  $n_1 = n_2$ .

Since  $f(n_1) = f(n_2)$ , we see that

$$2n_1 + 7 = 2n_2 + 7.$$

This in turn means that

$$2n_1 = 2n_2,$$

so  $n_1 = n_2$ , as required. ■

Good exercise: Repeat this proof using the other definition of injectivity!

# Injective Functions

**Theorem:** Let  $f : \mathbb{N} \rightarrow \mathbb{N}$  be defined as  $f(n) = 2n + 7$ .  
Then  $f$  is injective.

**Proof:** Consider any  $n_1, n_2 \in \mathbb{N}$  where  $f(n_1) = f(n_2)$ . We will prove that  $n_1 = n_2$ .

Since  $f(n_1) = f(n_2)$ , we see that

$$2n_1 + 7 = 2n_2 + 7.$$

This in turn means that

$$2n_1 = 2n_2,$$

so  $n_1 = n_2$ , as required. ■

This proof contains no first-order logic syntax (quantifiers, connectives, etc.). It's written in plain English, just as usual.

# Injective Functions

**Theorem:** Let  $f : \mathbb{Z} \rightarrow \mathbb{N}$  be defined as  $f(x) = x^4$ . Then  $f$  is not injective.

**Proof:**

What does it mean for  $f$  to be injective?

$$\forall x_1 \in \mathbb{Z}. \forall x_2 \in \mathbb{Z}. (x_1 \neq x_2 \rightarrow f(x_1) \neq f(x_2))$$

What is the negation of this statement?

$$\neg \forall x_1 \in \mathbb{Z}. \forall x_2 \in \mathbb{Z}. (x_1 \neq x_2 \rightarrow f(x_1) \neq f(x_2))$$

$$\exists x_1 \in \mathbb{Z}. \neg \forall x_2 \in \mathbb{Z}. (x_1 \neq x_2 \rightarrow f(x_1) \neq f(x_2))$$

$$\exists x_1 \in \mathbb{Z}. \exists x_2 \in \mathbb{Z}. \neg (x_1 \neq x_2 \rightarrow f(x_1) \neq f(x_2))$$

$$\exists x_1 \in \mathbb{Z}. \exists x_2 \in \mathbb{Z}. (x_1 \neq x_2 \wedge \neg (f(x_1) \neq f(x_2)))$$

$$\exists x_1 \in \mathbb{Z}. \exists x_2 \in \mathbb{Z}. (x_1 \neq x_2 \wedge f(x_1) = f(x_2))$$

Therefore, we need to find  $x_1, x_2 \in \mathbb{Z}$  such that  $x_1 \neq x_2$ , but  $f(x_1) = f(x_2)$ .

Can we do that?

# Injective Functions

**Theorem:** Let  $f : \mathbb{Z} \rightarrow \mathbb{N}$  be defined as  $f(x) = x^4$ . Then  $f$  is not injective.

**Proof:** We will prove that there exist integers  $x_1$  and  $x_2$  such that  $x_1 \neq x_2$ , but  $f(x_1) = f(x_2)$ .

Let  $x_1 = -1$  and  $x_2 = +1$ . Notice that

$$f(x_1) = f(-1) = (-1)^4 = 1$$

and

$$f(x_2) = f(1) = 1^4 = 1,$$

so  $f(x_1) = f(x_2)$  even though  $x_1 \neq x_2$ , as required. ■

	To <i>prove</i> that this is true...	
$\forall x. A$	Have the reader pick an arbitrary $x$ . We then prove $A$ is true for that choice of $x$ .	
$\exists x. A$	Find an $x$ where $A$ is true. Then prove that $A$ is true for that specific choice of $x$ .	
$A \rightarrow B$	Assume $A$ is true, then prove $B$ is true.	
$A \wedge B$	Prove $A$ . Then prove $B$ .	
$A \vee B$	Either prove $\neg A \rightarrow B$ or prove $\neg B \rightarrow A$ . <i>(Why does this work?)</i>	
$A \leftrightarrow B$	Prove $A \rightarrow B$ and $B \rightarrow A$ .	
$\neg A$	Simplify the negation, then consult this table on the result.	

# Two More Classes of Functions

Lassen Peak

Mt. Shasta

Crater Lake

Mt. McLoughlin

Mt. Hood

Mt. St. Helens

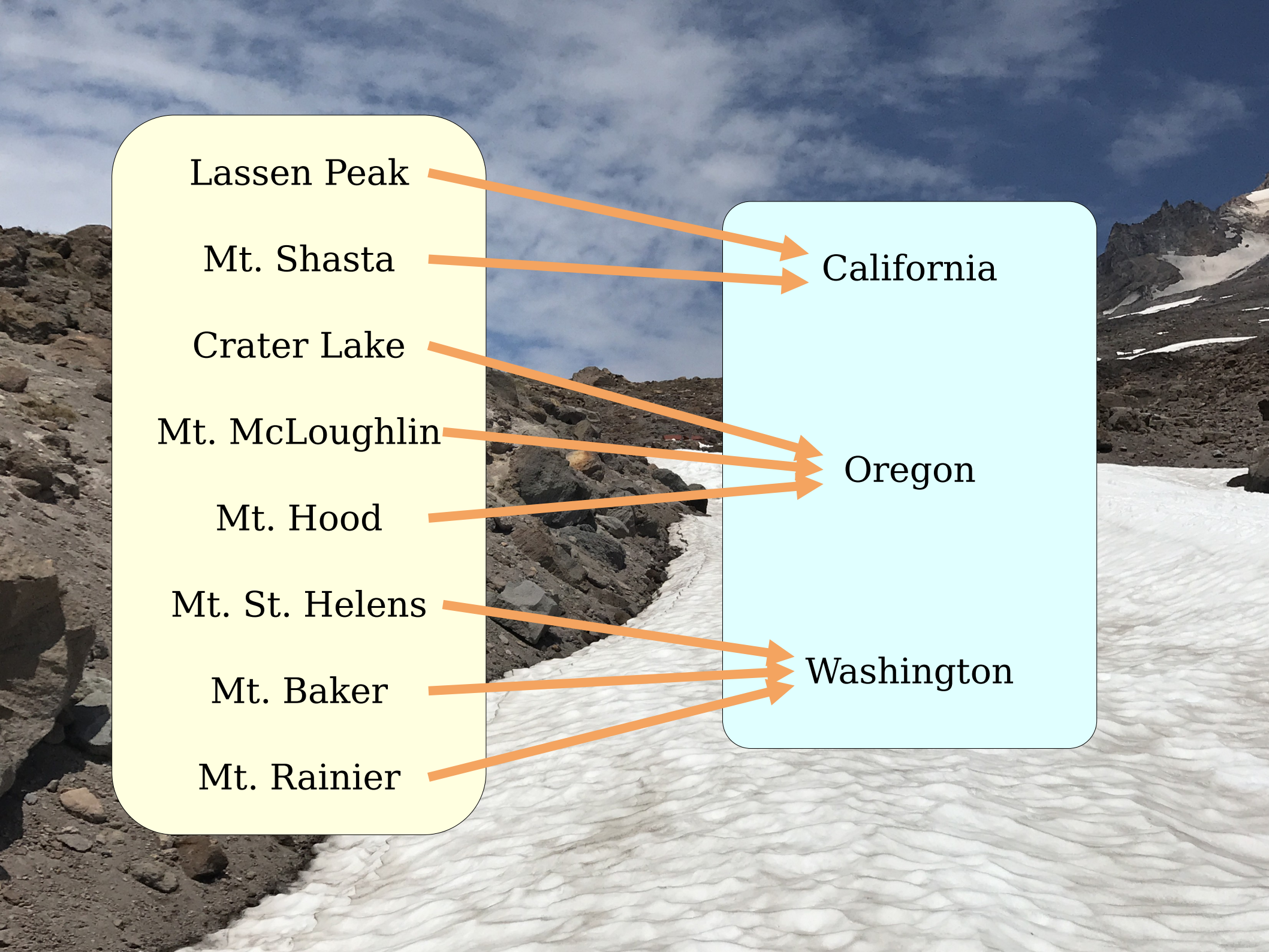
Mt. Baker

Mt. Rainier

California

Oregon

Washington



# Surjective Functions

- A function  $f : A \rightarrow B$  is called **surjective** (or **onto**) if this first-order logic statement is true about  $f$ :

$$\forall b \in B. \exists a \in A. f(a) = b$$

*(“For every output, there's an input that produces it.”)*

- A function with this property is called a **surjection**.
- How does this compare to our first rule of functions?

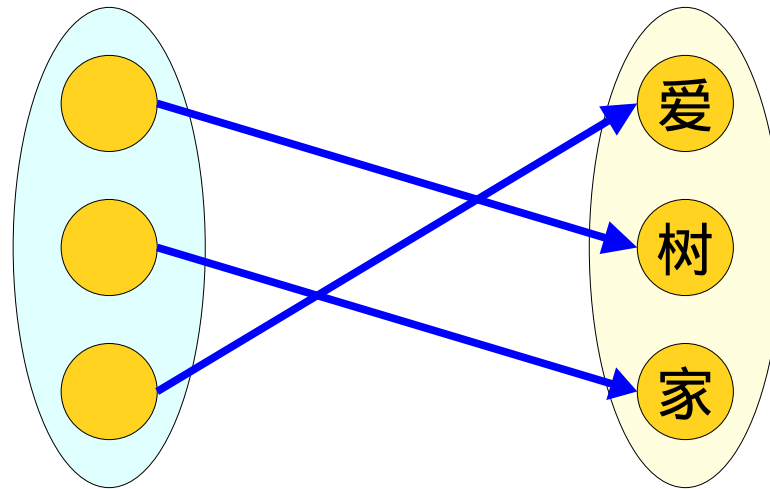
Check the appendix for  
sample proofs involving  
injections.

# Injections and Surjections

- An injective function associates *at most* one element of the domain with each element of the codomain.
- A surjective function associates *at least* one element of the domain with each element of the codomain.
- What about functions that associate *exactly one* element of the domain with each element of the codomain?

# Bijections

- A ***bijection*** is a function that is both injective and surjective.
- Intuitively, if  $f : A \rightarrow B$  is a bijection, then  $f$  represents a way of pairing off elements of  $A$  and elements of  $B$ .



# Bijections

- Which of the following are bijections?
  - $f : \mathbb{R} \rightarrow \mathbb{R}$  defined as  $f(x) = x$ . *Yep!*
  - $g : \mathbb{Z} \rightarrow \mathbb{R}$  defined as  $g(x) = x$ . *Nope!*
  - $h : \mathbb{R} \rightarrow \mathbb{R}$  defined as  $h(x) = 2x + 1$ . *Yep!*
  - $p : \mathbb{Z} \rightarrow \mathbb{Z}$  defined as  $p(x) = 2x + 1$ . *Nope!*

---

A ***bijection*** is a function that is both injective and surjective.

# Next Time

- ***First-Order Assumptions***
  - The difference between assuming something is true and proving something is true.
- ***Connecting Function Types***
  - Involutions, injections, and surjections are related to one another. How?
- ***Function Composition***
  - Sequencing functions together.

## ***Appendix:*** More Proofs on Functions

***Proof 1:*** Proving a function is surjective.

# Surjective Functions

**Theorem:** Let  $f : \mathbb{R} \rightarrow \mathbb{R}$  be defined as  $f(x) = 2x$ . Then  $f(x)$  is surjective.

**Proof:**

What does it mean for  $f$  to be surjective?

$$\forall y \in \mathbb{R}. \exists x \in \mathbb{R}. f(x) = y$$

Therefore, we'll choose an arbitrary  $y \in \mathbb{R}$ , then prove that there is some  $x \in \mathbb{R}$  where  $f(x) = y$ .

# Surjective Functions

**Theorem:** Let  $f : \mathbb{R} \rightarrow \mathbb{R}$  be defined as  $f(x) = 2x$ . Then  $f(x)$  is surjective.

**Proof:** Consider any  $y \in \mathbb{R}$ . We will prove that there is a choice of  $x \in \mathbb{R}$  such that  $f(x) = y$ .

Let  $x = y / 2$ . Then we see that

$$f(x) = f(y / 2) = 2y / 2 = y.$$

So we see that  $f(x) = y$ , as required. ■

This proof contains no first-order logic syntax (quantifiers, connectives, etc.). It's written in plain English, just as usual.

***Proof 2:*** Proving a function is not surjective.

# Surjective Functions

**Theorem:** Let  $g : \mathbb{N} \rightarrow \mathbb{N}$  be defined as  $g(n) = 2n$ . Then  $g(x)$  is not surjective.

What does it mean for  $g$  to be surjective?

$$\forall n \in \mathbb{N}. \exists m \in \mathbb{N}. g(m) = n$$

What is the negation of the above statement?

$$\neg \forall n \in \mathbb{N}. \exists m \in \mathbb{N}. g(m) = n$$

$$\exists n \in \mathbb{N}. \neg \exists m \in \mathbb{N}. g(m) = n$$

$$\exists n \in \mathbb{N}. \forall m \in \mathbb{N}. g(m) \neq n$$

Therefore, we need to find a natural number  $n$  where, regardless of which  $m$  we pick, we have  $g(m) \neq n$ .

# Surjective Functions

**Theorem:** Let  $g : \mathbb{N} \rightarrow \mathbb{N}$  be defined as  $g(n) = 2n$ . Then  $g(x)$  is not surjective.

**Proof:** Let  $n = 137$ .

Our overall goal is to prove

$$\exists n \in \mathbb{N}. \forall m \in \mathbb{N}. g(m) \neq n.$$

We just made our choice of  $n$ .

Therefore, we need to prove

$$\forall m \in \mathbb{N}. g(m) \neq n.$$

We'll therefore pick an arbitrary  $m \in \mathbb{N}$ , then prove that  $g(m) \neq n$ .

# Surjective Functions

**Theorem:** Let  $g : \mathbb{N} \rightarrow \mathbb{N}$  be defined as  $g(n) = 2n$ . Then  $g(x)$  is not surjective.

**Proof:** Let  $n = 137$ . Now, pick an arbitrary  $m \in \mathbb{N}$ . We need to show that  $g(m) \neq n$ .

Notice that  $g(m) = 2m$  is even, while 137 is odd. Therefore, we have  $g(m) \neq 137$ , as required. ■

This proof contains no first-order logic syntax (quantifiers, connectives, etc.). It's written in plain English, just as usual.