The Pigeonhole Principle
&
Functions

Outline for Today

- **Special Quantifiers**
  - Quantifying over sets.
- **The Pigeonhole Principle**
  - Proving results are true just by counting.
- **Functions**
  - Modeling transformations between sets.
- **Cardinality (ITA)**
  - Revisiting our first lecture with our new techniques!

One Last Bit of Logic

Special Quantifiers

- Mathematicians are extremely lazy and don't like writing symbols they don't have to.
- Over time, several variations on the ∀ and ∃ quantifiers have been developed.
- We'd like to introduce you to one of them that we'll be using going forward.

Quantifying Over Sets

- The notation

$$\forall x \in S.\ \psi$$

  means "for ***every*** element x of set S, ψ is true."
- This is not technically a part of first-order logic; it is a shorthand for

$$\forall x.\ (x \in S \to \psi)$$

- The notation

$$\exists x \in S.\ \psi$$

  means "for ***some*** element x of set S, ψ is true."
- It's shorthand for

$$\exists x.\ (x \in S \wedge \psi)$$

Quantifying Over Sets

- The syntax

  $$\forall x \in S.\ \varphi$$

  $$\exists x \in S.\ \varphi$$

  is allowed for quantifying over sets.
- In CS103, you can use the set quantifiers, but please do not generalize it to more complex statements.
- For example, please don't do things like this:

  $$\forall x \text{ with } P(x).\ Q(x)$$

  $$\forall y \text{ such that } P(y) \wedge Q(y).\ R(y).$$

Changing Gears: The Pigeonhole Principle

The ***pigeonhole principle*** is the following:

If m objects are placed into n bins,
where $m > n$, then some bin contains
at least two objects.

(We sketched a proof in Lecture #02)

Why This Matters

- The pigeonhole principle can be used to show results must be true because they are “too big to fail.”
- Given a large enough number of objects with a bounded number of properties, eventually at least two of them will share a property.
- Can be used to prove some surprising results.

Using the Pigeonhole Principle

- To use the pigeonhole principle:
  - Find the m objects to distribute.
  - Find the $n < m$ buckets into which to distribute them.
  - Conclude by the pigeonhole principle that there must be two objects in some bucket.
- The details of how to proceed from there are specific to the particular proof you're doing.

A Surprising Application

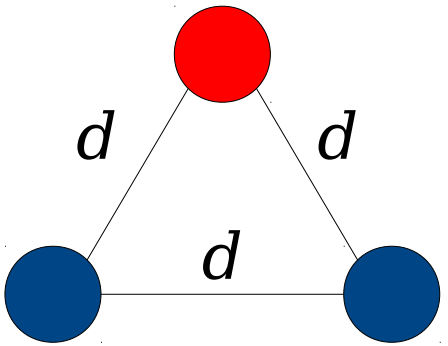
Theorem: Suppose that every point in the real plane is colored either red or blue. Then for any distance $d > 0$, there are two points exactly distance d from one another that are the same color.

Thought: There are two colors here, so if we start picking points, we'll be dropping them into one of two buckets (red or blue).

How many points do we need to pick to guarantee that we get two of the same color?

A Surprising Application

Theorem: Suppose that every point in the real plane is colored either red or blue. Then for any distance $d > 0$, there are two points exactly distance d from one another that are the same color.



A Surprising Application

Theorem: Suppose that every point in the real plane is colored either red or blue. Then for any distance $d > 0$, there are two points exactly distance d from one another that are the same color.

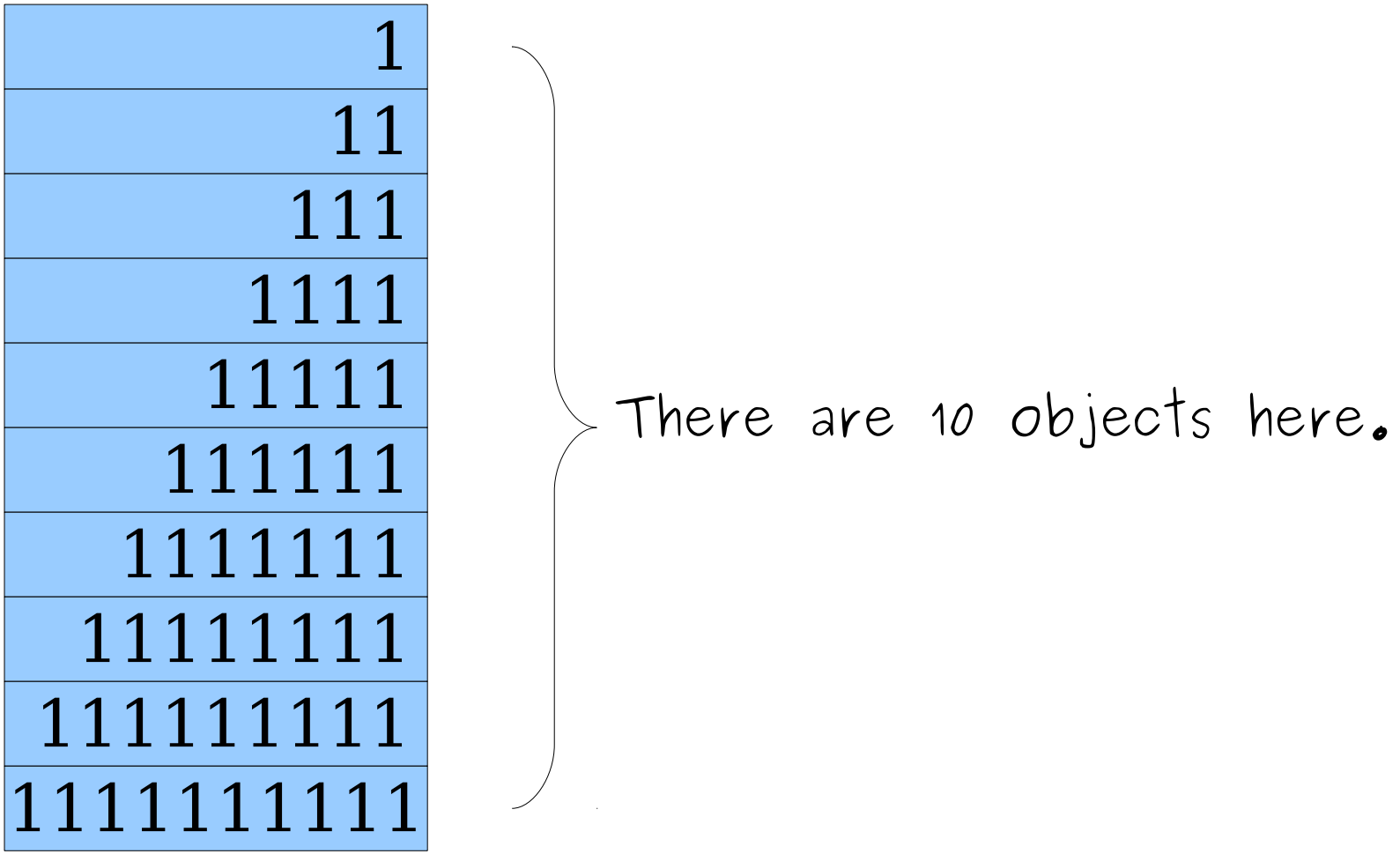
Proof: Consider any equilateral triangle whose side lengths are d. Put this triangle anywhere in the plane. Because the triangle has three vertices and each point in the plane is only one of two different colors, by the pigeonhole principle at least two of the vertices must have the same color. These vertices are at distance d from each other, as required. ■

The Hadwiger-Nelson Problem

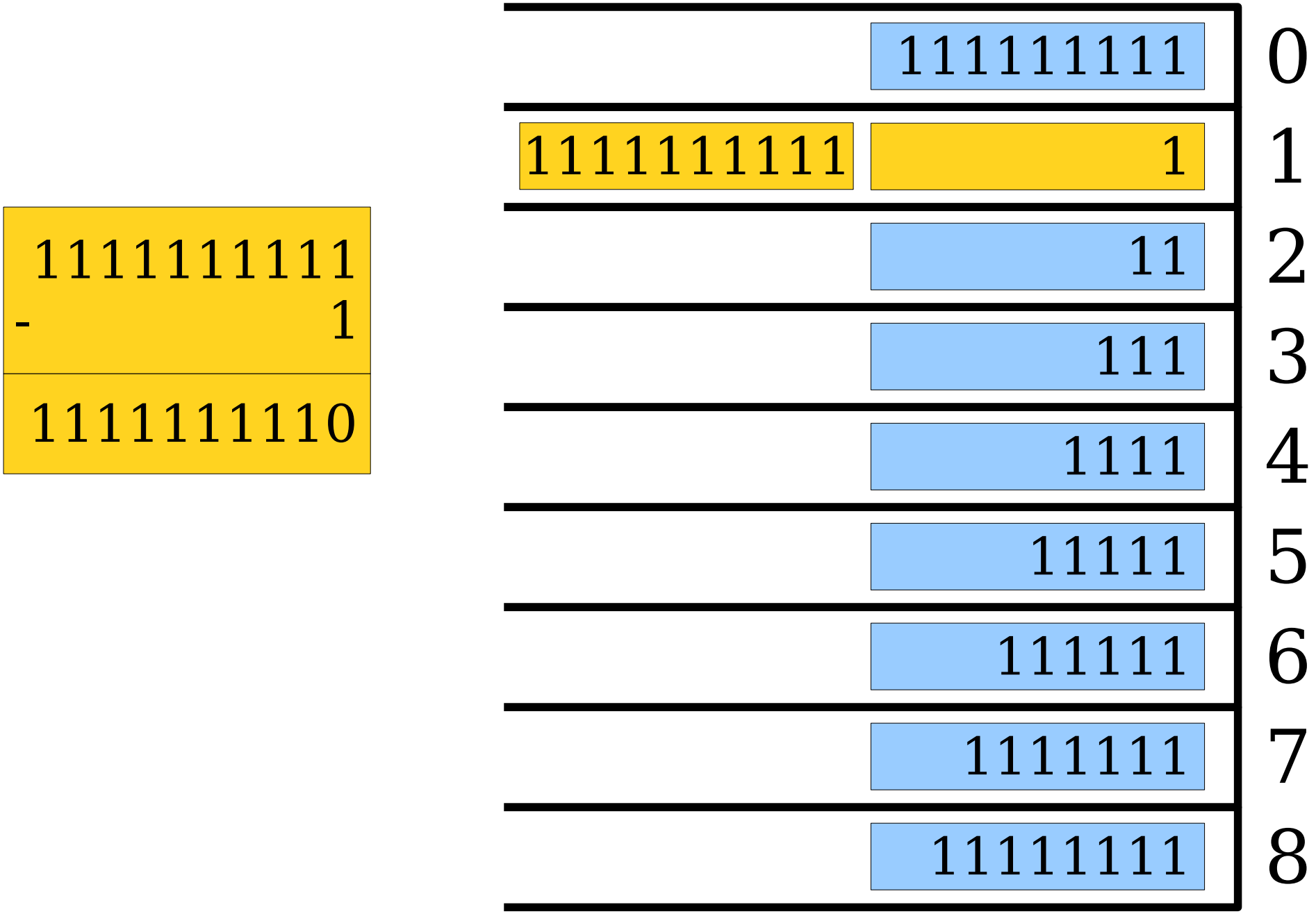
- No matter how you color the points of the plane, there will always be two points at distance 1 that are the same color.
- Relation to graph coloring:
  - Every point in the real plane is a node.
  - There's an edge between two points that are at distance exactly one.
- **Question:** What is the chromatic number of this graph? (That is, how many colors do you need to ensure no points at distance one are the same color?)
- This is the ***Hadwiger-Nelson problem***. It's known that the number is between 4 and 7, but no one knows for sure!

Theorem: For any nonzero natural number n, there is a nonzero multiple of n whose digits are all 0s and 1s.

Theorem: For any nonzero natural number n, there is a nonzero multiple of n whose digits are all 0s and 1s.



Theorem: For any nonzero natural number n, there is a nonzero multiple of n whose digits are all 0s and 1s.



Proof Idea

- Generate the numbers 1, 11, 111, … until $n+1$ numbers have been generated.
- There are n possible remainders modulo n, so two of these numbers have the same remainder.
- Their difference is a multiple of n.
- Their difference consists of 1s and 0s.

Theorem: Every positive natural number n has a nonzero multiple whose digits are all 0s and 1s.

Proof: Let n be an arbitrary positive natural number. Consider any $n+1$ distinct natural numbers whose digits are all 1's. Then, look at the remainders of those numbers modulo n. Since there are $n+1$ numbers and only n possible remainders modulo n, by the pigeonhole principle at least two of these numbers must leave the same remainder modulo n. Call these numbers p and q, and assume without loss of generality that $p > q$.

We claim that the number $p - q$ is the nonzero multiple of n that we're looking for. Since $p > q$, we know that $p - q$ is nonzero. Additionally, given the way that p and q are structured, the 1s in the digits of q will cancel out some tail of the 1s in the digits of p, so $p - q$ consists of a string of 1s followed by a string of 0s. If we can show that $p - q$ is a multiple of n, then we're done.

We know that p and q have the same remainder modulo n; let's call that remainder r. This means that there are integers s and t such that $p = sn + r$ and $q = tn + r$. Then

$$p - q = (sn + r) - (tn + r) = (s - t)n.$$

So $p - q$ is a multiple of n. Therefore, $p - q$ is a nonzero multiple of n with only 1s and 0s as its digits, as required. ■

Notes on Pigeonholes

- Typically, the pigeonhole principle is used as a key step in a larger proof.
- Proofs using the pigeonhole principle are often insight-based; you need to have the right observation to see when to use the pigeonhole principle.
- *These previous two proofs are by no means obvious*. Don't panic if you didn't come up with them on your own!

Time-Out for Announcements!

Solution Sets

- Solutions to the checkpoint problem and the discussion problems from this week are available outside.
- There are a lot of very lonely solution sets sitting in a dark filing cabinet, wondering if some day, someone will be their friends. Please pick them up!

Problem Set Grading

- Problem Set Two should be graded and returned by the end of class (allow about an hour for us to finalize and release everything).
- Problem Set Three checkpoints also graded, will be returned at around the same time.

Midterm Logistics

- The first midterm is on ***Thursday, October 23*** from ***6PM – 9PM***.
  - We'll announce testing locations on Friday.
  - Alternate exams: if you've requested an alternate exam time, Maesen should have replied to you. If you haven't heard from her and need to take the exam at an alternate time, let us know ASAP.
- Covers material up through and including first-order logic (content from PS1 – PS3).
- The exam is closed-book and closed-computer.
- You can bring one double-sided, 8.5” × 11” page of notes with you when you take the exam.

Review Materials

- We've posted some practice problems up on the course website that you can use to review for the midterm.
- We'll release solutions and a new set of practice problems on Friday.
- We'll also poll you for input on what you'd like more practice with so that we can put together even more practice problems.

Practice Exam

- We have reserved Annenberg Auditorium on next **Monday, October 20** from **7PM – 10PM** for a practice midterm exam.
- Want to try working through a practice midterm under realistic exam conditions? Stop on by! The TAs will be available to answer questions afterwards.
- Purely optional, but highly recommended if you want to get a sense for what the test will be like.
- SCPD students – we'll send out the practice exam through the normal distribution channels so that you can take it on your own time.

CS + History
Information Session

October 22, 2014 at 4:15 pm
Lane History Corner, room 307

Stanford recently created a series of joint major programs uniting Computer Science with the Humanities, including History. For info about this exciting new opportunity come to our CS + History Information session.

Advice about Working in Teams

Your Questions

“How important have you found GPA to be in success after college? I feel like my worry about keeping my GPA up has kept me from enjoying some classes that I'm in or even from taking some hard, but interesting, classes in the first place.”

“I'm having a lot of trouble on PSet 3 and now I'm really frightened about the midterm. If I can't do PSet 3 by myself, how can I expect to do well on the midterm, when I don't have anyone?”

“For the harder questions on problem sets, I read the question, think about it, but usually don't come up with a solution until I'm lying in bed, or on a long bike ride, or in the shower, etc. Am I going to be hosed on the exam?”

“We've talked a lot about 'translating' English statements into first order logic. Could we practice translating from logic back into English, so that once we manipulate a logical statement we can find meaning in it again?”

“So I'm about to declare CS. Any tips on how I should choose a CS advisor? Who would you recommend?”

Back to CS103!

Functions

A ***function*** is a means of associating each object in one set with an object in some other set.

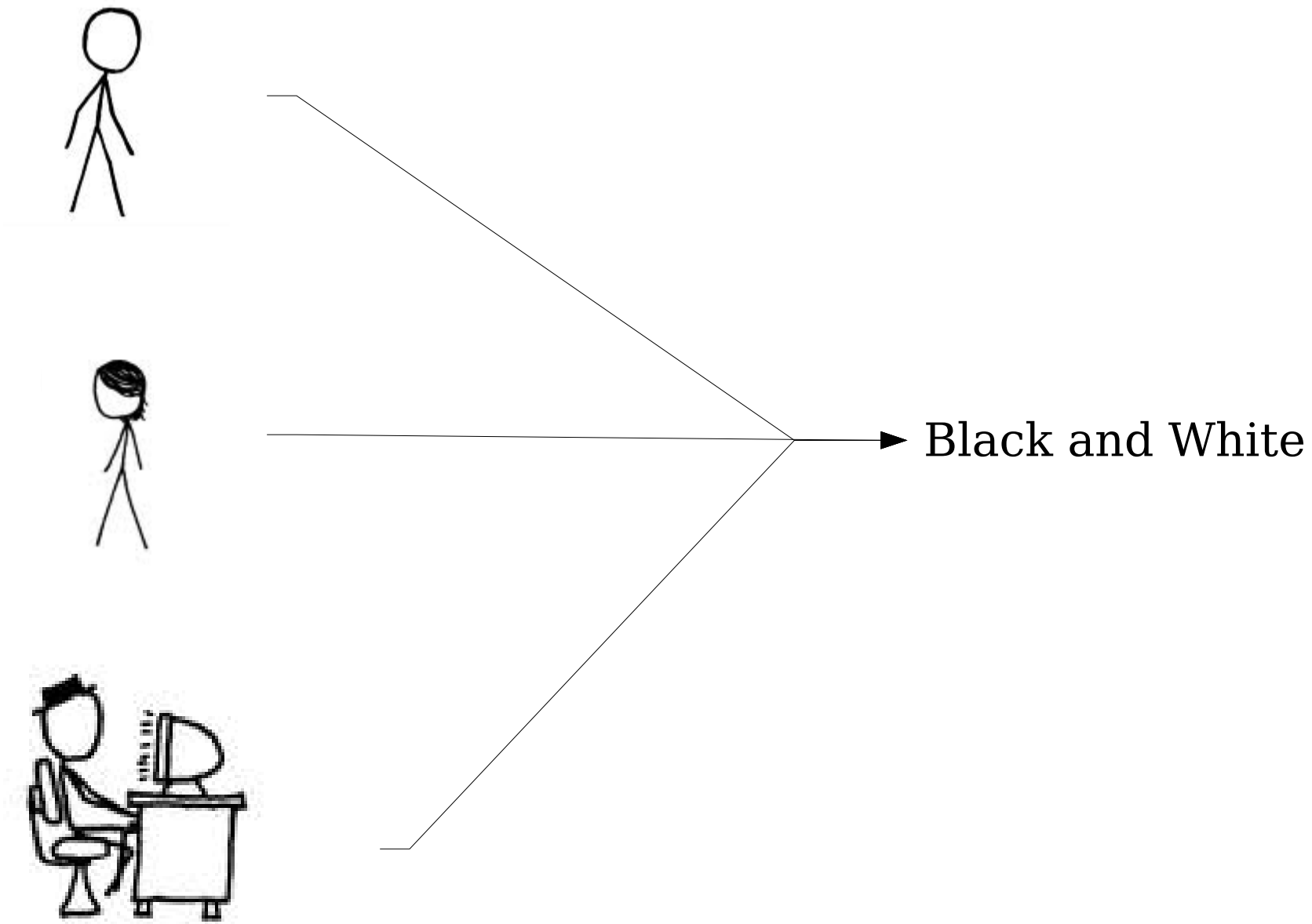




Dikdik

Nubian Ibex

Sloth


Black and White

Terminology

- A ***function*** f is a mapping from one set A to another set B such that every element of A is associated with a single element of B.
  - For each $a \in A$, there is some $b \in B$ with $f(a) = b$.
  - If $f(a) = b_0$ and $f(a) = b_1$, then $b_0 = b_1$.
- If f is a function from A to B, we say that f is a ***mapping*** from A to B.
  - We call A the ***domain*** of f.
  - We call B the ***codomain*** of f.
- We denote that f is a function from A to B by writing $\boldsymbol{f : A \to B}$.

Defining Functions

- Typically, we specify a function by describing a rule that maps every element of the domain to some element of the codomain.
- Examples:
  - $f(n) = n + 1$, where $f : \mathbb{Z} \to \mathbb{Z}$
  - $f(x) = \sin x$, where $f : \mathbb{R} \to \mathbb{R}$
  - $f(x) = \lceil x \rceil$, where $f : \mathbb{R} \to \mathbb{Z}$
- Notice that we're giving both a rule and the domain/codomain.

Defining Functions

Typically, we specify a function by describing a rule that maps every element of the domain to some element of the codomain.

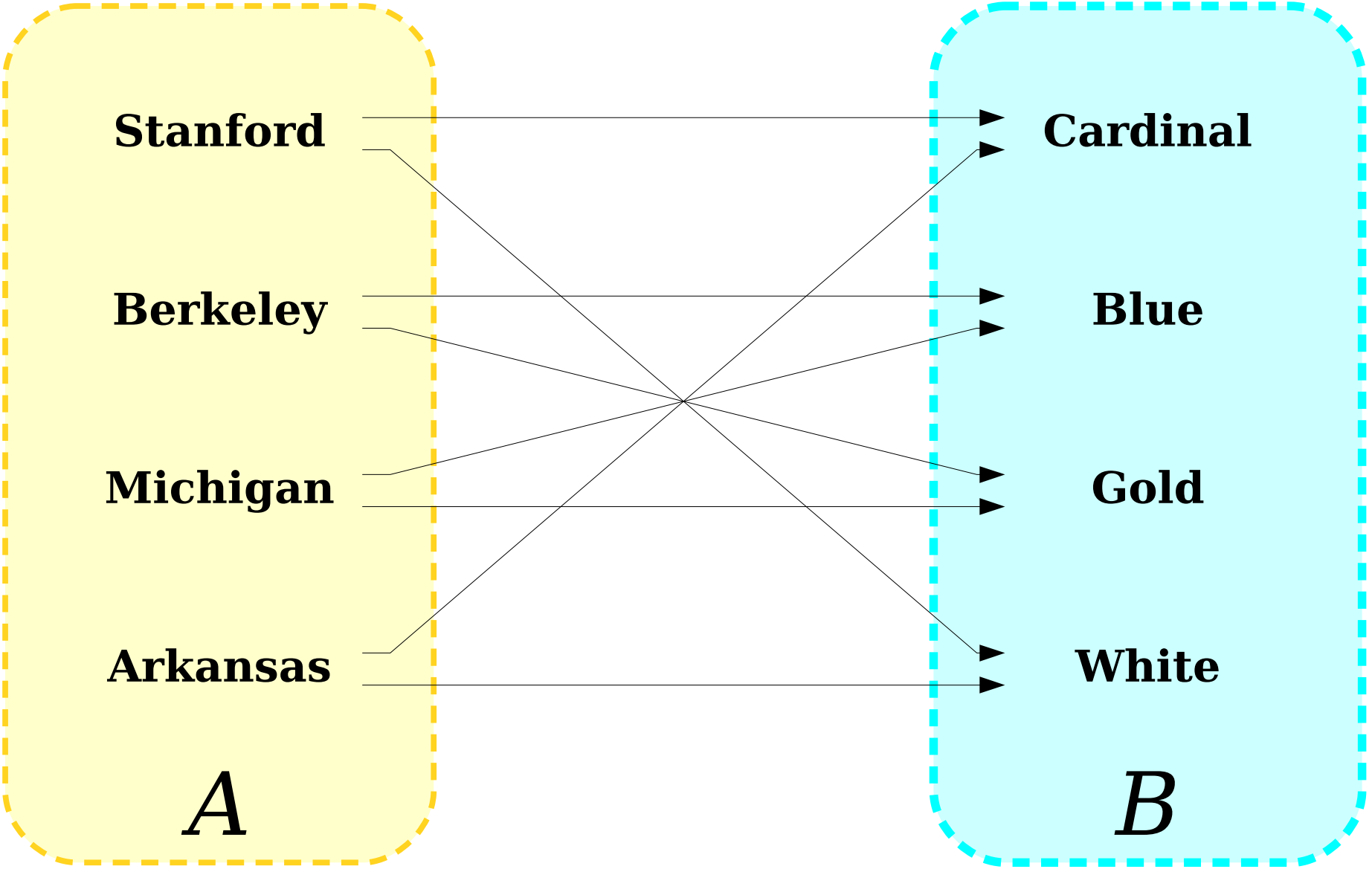
Examples:

- $f(n) = n + 1$, where $f : \mathbb{Z} \to \mathbb{Z}$
- $f(x) = \sin x$, where $f : \mathbb{R} \to \mathbb{R}$
- $f(x) = \lceil x \rceil$, where $f : \mathbb{R} \to \mathbb{Z}$

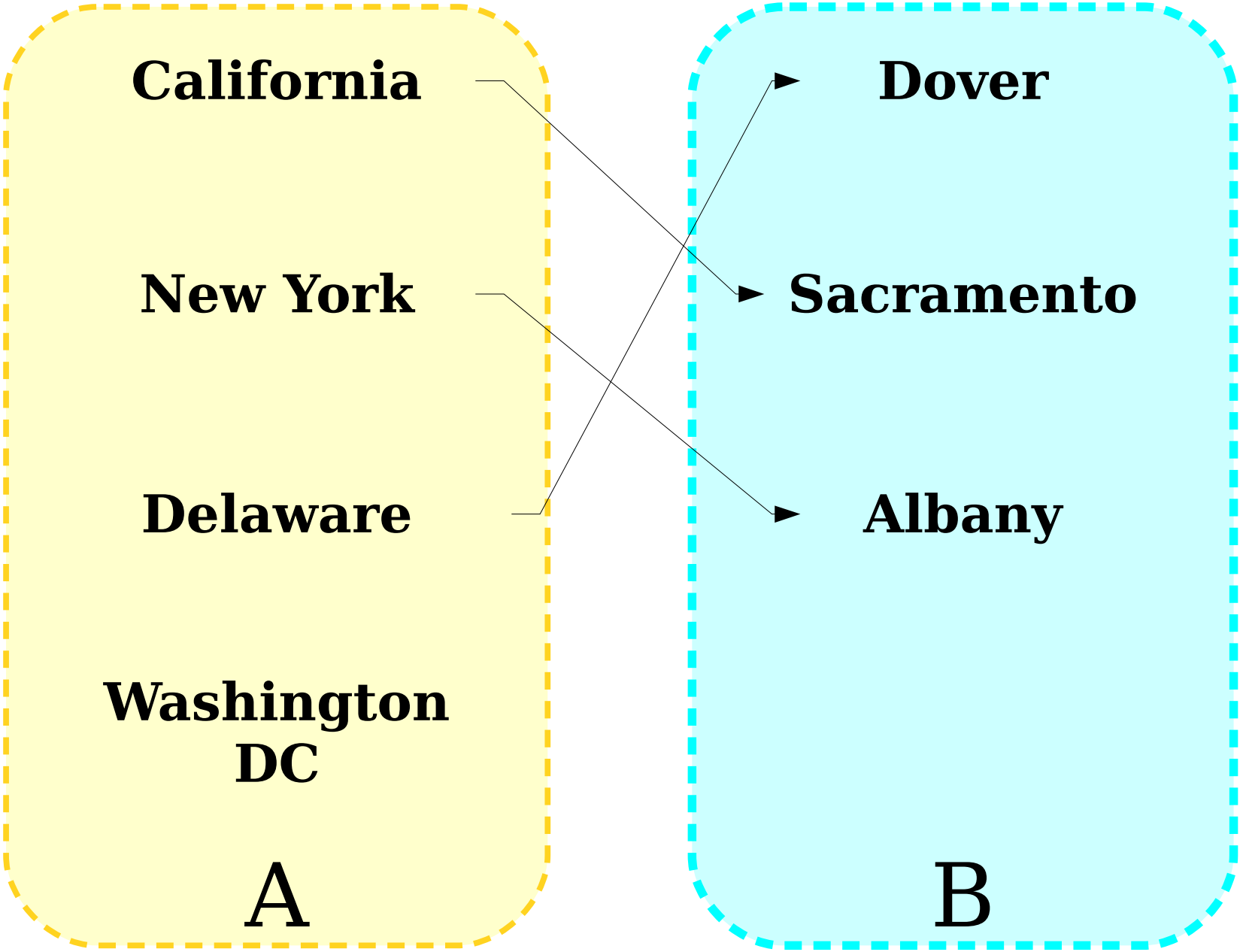
This is the ceiling function – the smallest integer greater than or equal to x. For example, ⌈1⌉ = 1, ⌈1.37⌉ = 2, and ⌈π⌉ = 4.

Notice that we're giving both a rule and the domain/codomain.

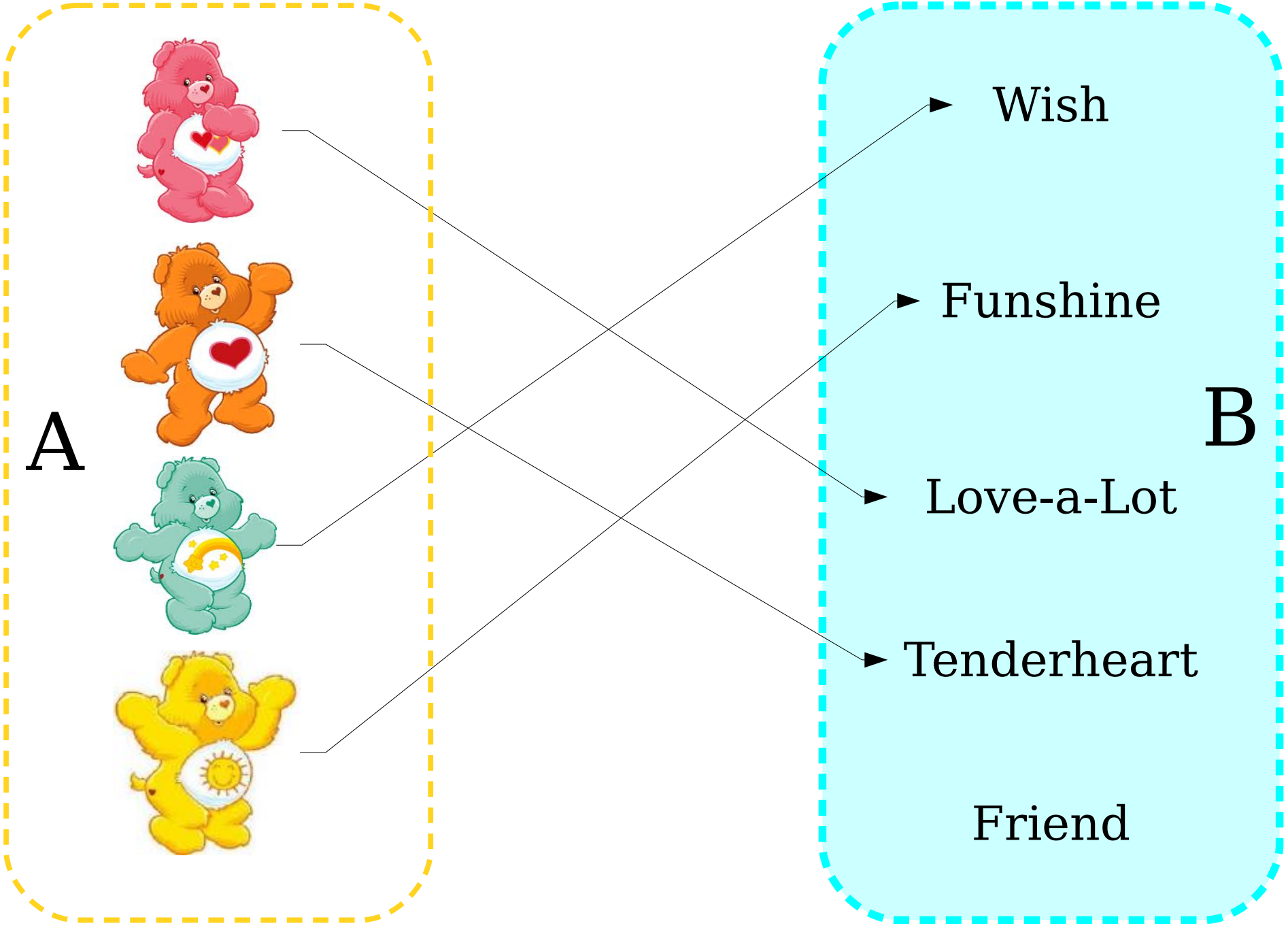
Is this a function from A to B?



Is this a function from A to B?



Is this a function from A to B?

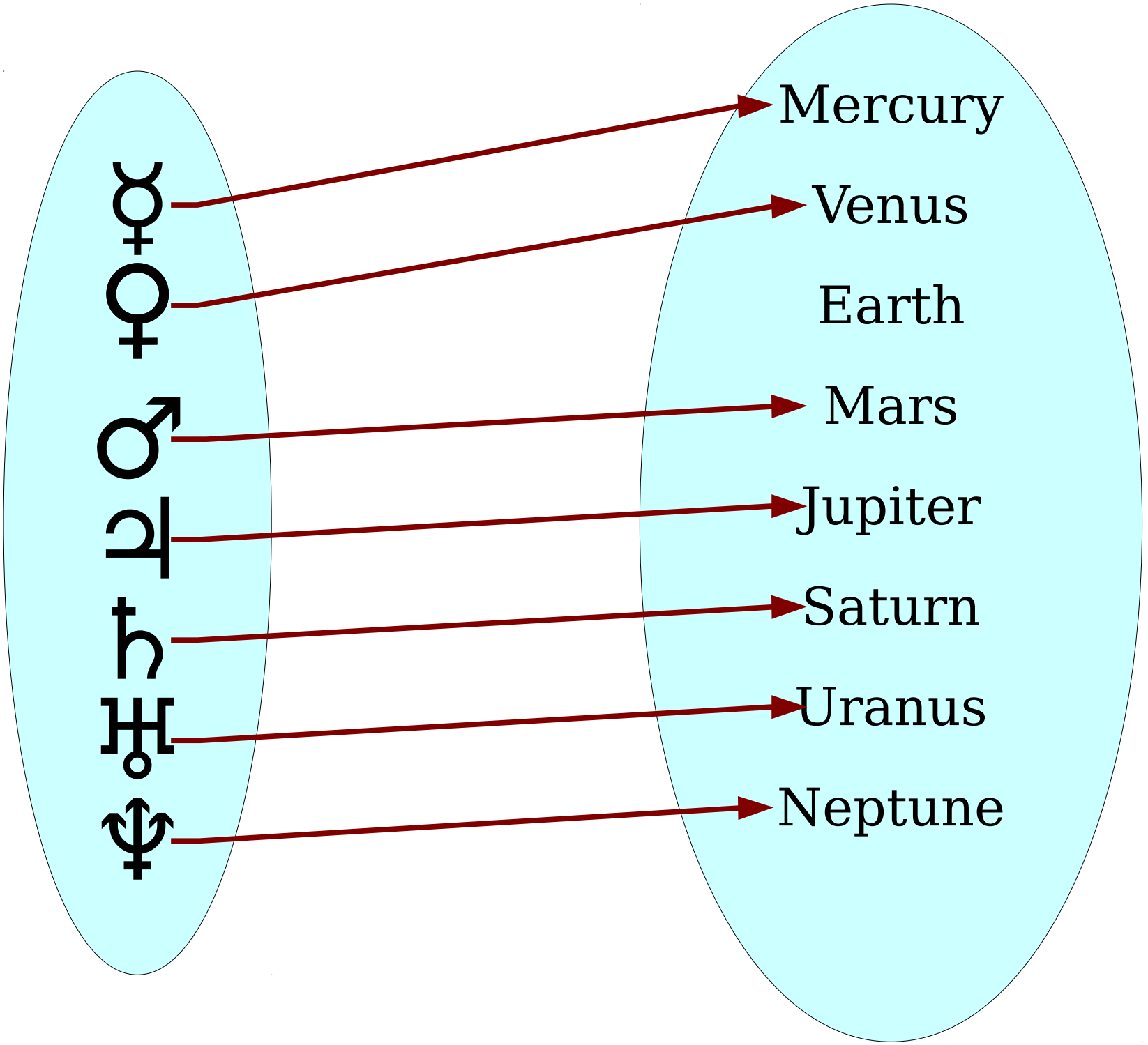


Piecewise Functions

- Functions may be specified ***piecewise***, with different rules applying to different elements.
- As an example, consider this function $f : \mathbb{N} \to \mathbb{Z}$ defined as follows:

$$f(n) = \begin{cases} -n/2 & \text{if } n \text{ is even} \\ (n+1)/2 & \text{otherwise} \end{cases}$$

- When defining a function piecewise, it's up to you to confirm that it defines a legal function!


☿
♀
♁
♂
♃
♄
♅
♆
Mercury
Venus
Earth
Mars
Jupiter
Saturn
Uranus
Neptune

Injective Functions

- A function $f : A \to B$ is called ***injective*** (or ***one-to-one***) if each element of the codomain has at most one element of the domain that maps to it.
  - A function with this property is called an ***injection***.
- Formally, $f : A \to B$ is an injection if this statement is true:

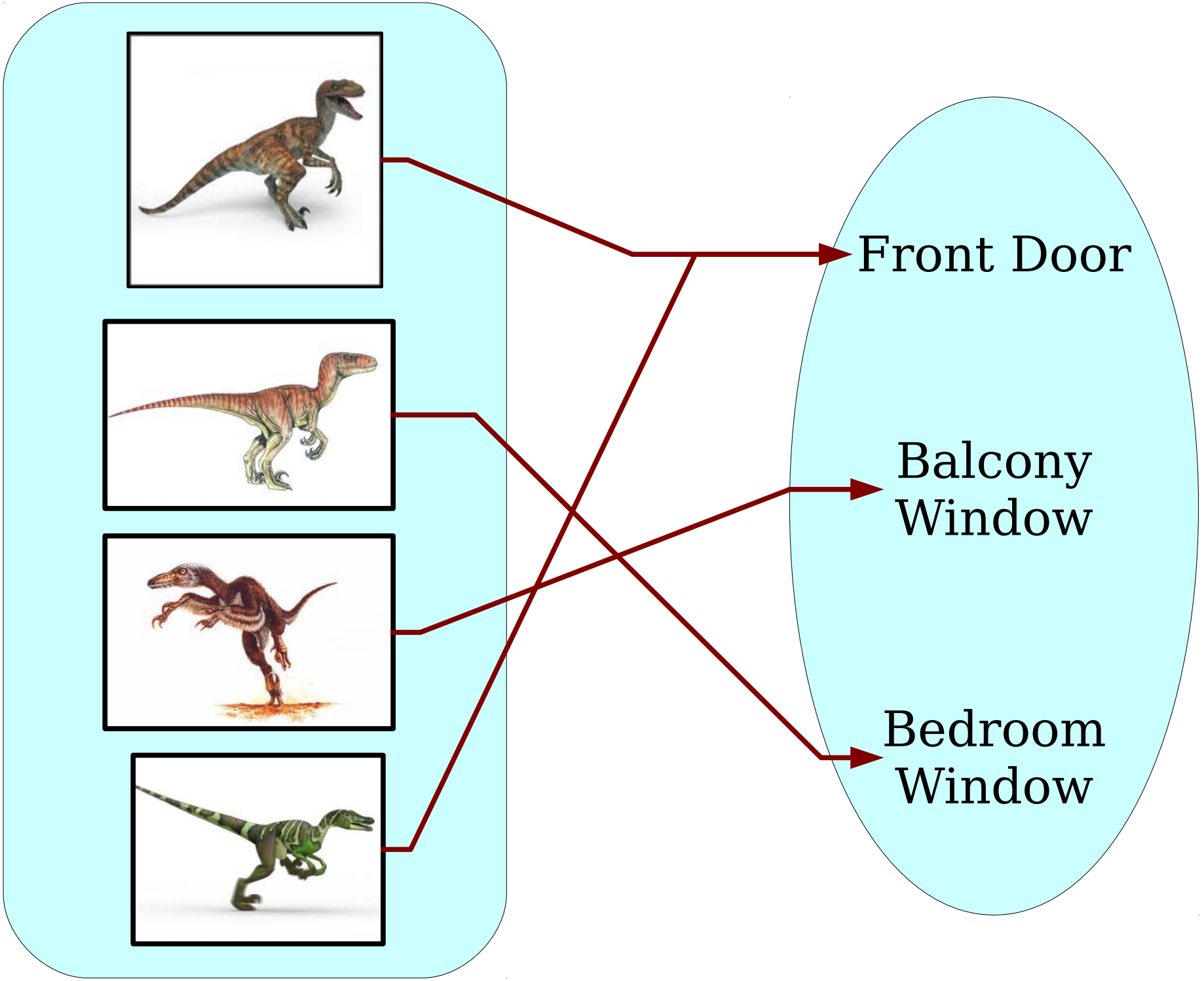
$$\forall a_1 \in A.\ \forall a_2 \in A.\ (f(a_1) = f(a_2) \to a_1 = a_2)$$

(*"If the outputs are the same, the inputs are the same"*)

- Equivalently:

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

(*"If the inputs are different, the outputs are different"*)


Front Door
Balcony Window
Bedroom Window

Surjective Functions

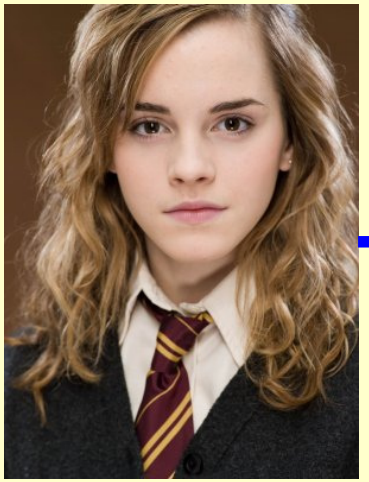
- A function $f : A \to B$ is called ***surjective*** (or ***onto***) if each element of the codomain has at least one element of the domain that maps to it.
  - A function with this property is called a ***surjection***.
- Formally, $f : A \to B$ is a surjection if this statement is true:

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

(*"For every possible output, there's at least one possible input that produces it"*)

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?



Katniss Everdeen

Merida

Hermione Granger

Bijections

- A function that associates each element of the codomain with a unique element of the domain is called ***bijective***.
  - Such a function is a ***bijection***.
- Formally, a bijection is a function that is both ***injective*** and ***surjective***.
- Bijections are sometimes called ***one-to-one correspondences***.
  - Not to be confused with “one-to-one functions.”

Cardinality Revisited

Cardinality

- Recall *(from our first lecture!)* that the ***cardinality*** of a set is the number of elements it contains.
- If S is a set, we denote its cardinality by $|S|$.
- For finite sets, cardinalities are natural numbers:
  - $|\{1, 2, 3\}| = 3$
  - $|\{100, 200\}| = 2$
- For infinite sets, we introduced ***infinite cardinals*** to denote the size of sets:

$$|\mathbb{N}| = \aleph_0$$

Defining Cardinality

- It is difficult to give a rigorous definition of what cardinalities actually are.
  - What is 4? What is $\aleph_0$?
- ***Idea:*** Define cardinality as a *relation* between two sets rather than as an absolute quantity.
- We'll define what these relations between sets mean without actually defining what "a cardinality" actually is:

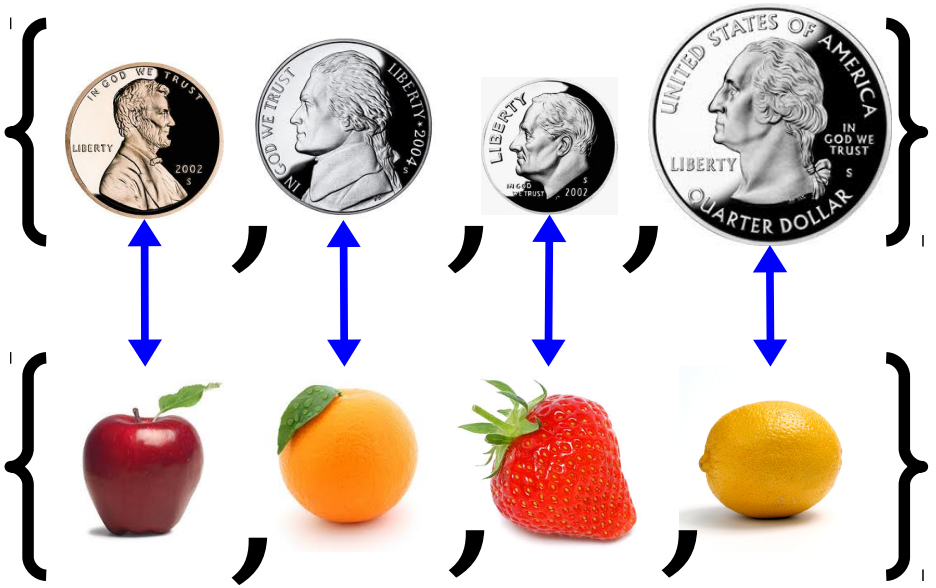
$$|S| = |T| \quad |S| \neq |T| \quad |S| \leq |T| \quad |S| < |T|$$

- Cardinality exists *between* sets!

Comparing Cardinalities

- The relationships between set cardinalities are defined in terms of functions between those sets.
- $|S| = |T|$ is defined using bijections.

  $|S| = |T|$ if there exists a *bijection* $f : S \to T$

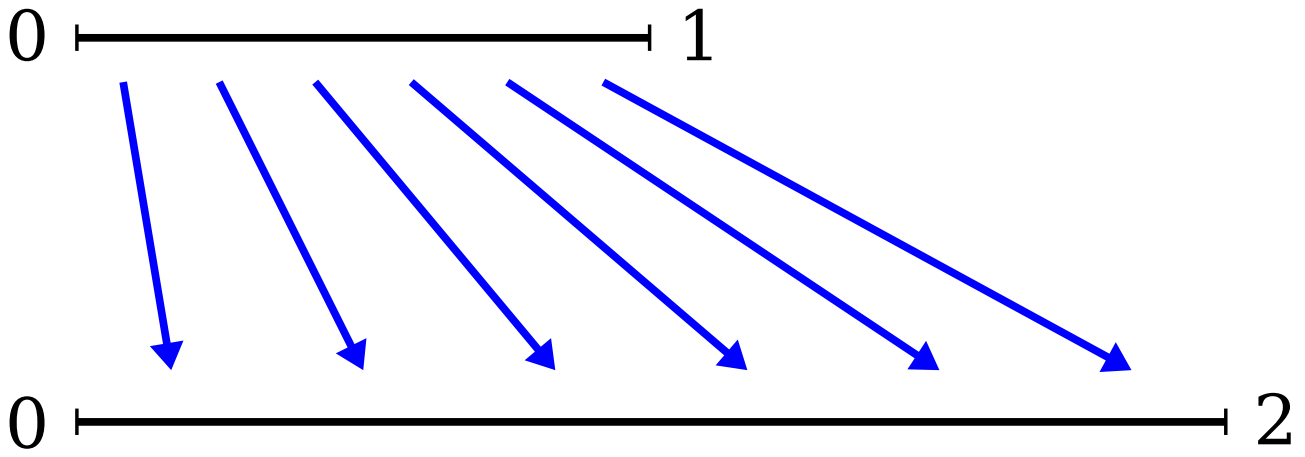


Properties of Cardinality

- Equality of cardinality is an equivalence relation.
- For any sets R, S, and T, the following are true:
  - $|S| = |S|$.
  - If $|S| = |T|$, then $|T| = |S|$.
  - If $|R| = |S|$ and $|S| = |T|$, then $|R| = |T|$.
- ***Read the course notes for proofs of these results!***

Infinity is Weird...

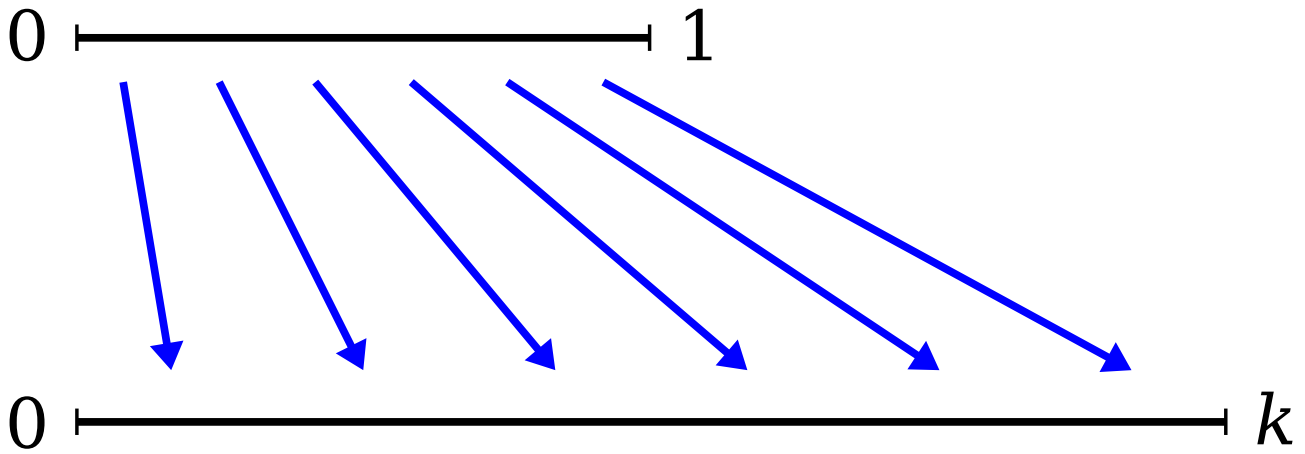
Home on the Range



$$f : [0, 1] \to [0, 2]$$
$$f(x) = 2x$$

$$|[0, 1]| = |[0, 2]|$$

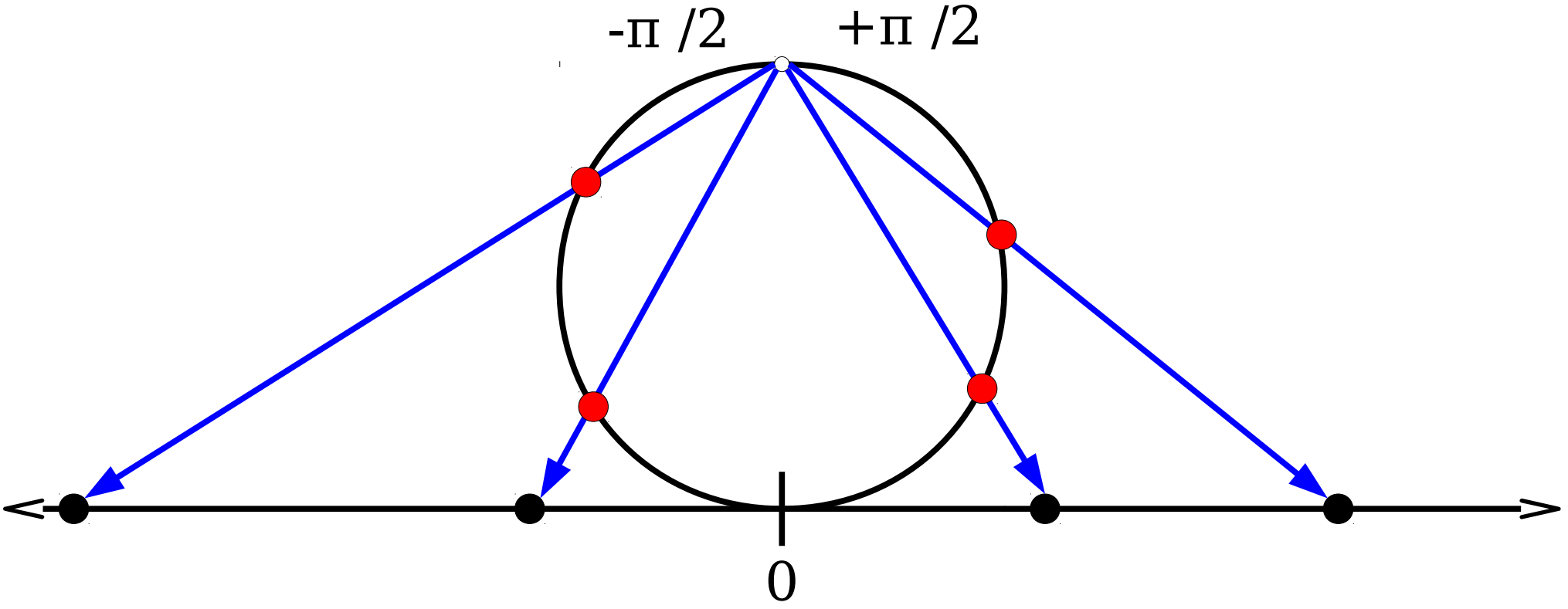
Home on the Range



$$f : [0, 1] \to [0, k]$$
$$f(x) = kx$$

$$|[0, 1]| = |[0, k]|$$

Put a Ring On It



$$f : (-\pi/2, \pi/2) \to \mathbb{R}$$
$$f(x) = \tan x$$

$$|(-\pi/2, \pi/2)| = |\mathbb{R}|$$

Next Time

- **Equal Cardinalities**
  - Showing that two sets *do* have the same size.
- **Unequal Cardinalities**
  - Showing that two sets *don't* have the same size.
- **Diagonalization**
  - Formalizing the diagonal argument from our first lecture.