Binary Relations

Outline for Today

- **Binary Relations**
  - Reasoning about connections between objects.
- **Equivalence Relations**
  - Reasoning about clusters.
- **Strict Orders**
  - Reasoning about prerequisites.

Relationships

- In CS103, you've seen examples of relationships
  - between sets:
    - $A \subseteq B$
  - between numbers:
    - $x < y \quad x \equiv_k y \quad x \leq y$
  - between people:
    - p loves q
- Many other relations exist in more specialized contexts, and we'll see a bunch of them over the course of the quarter.

Binary Relations

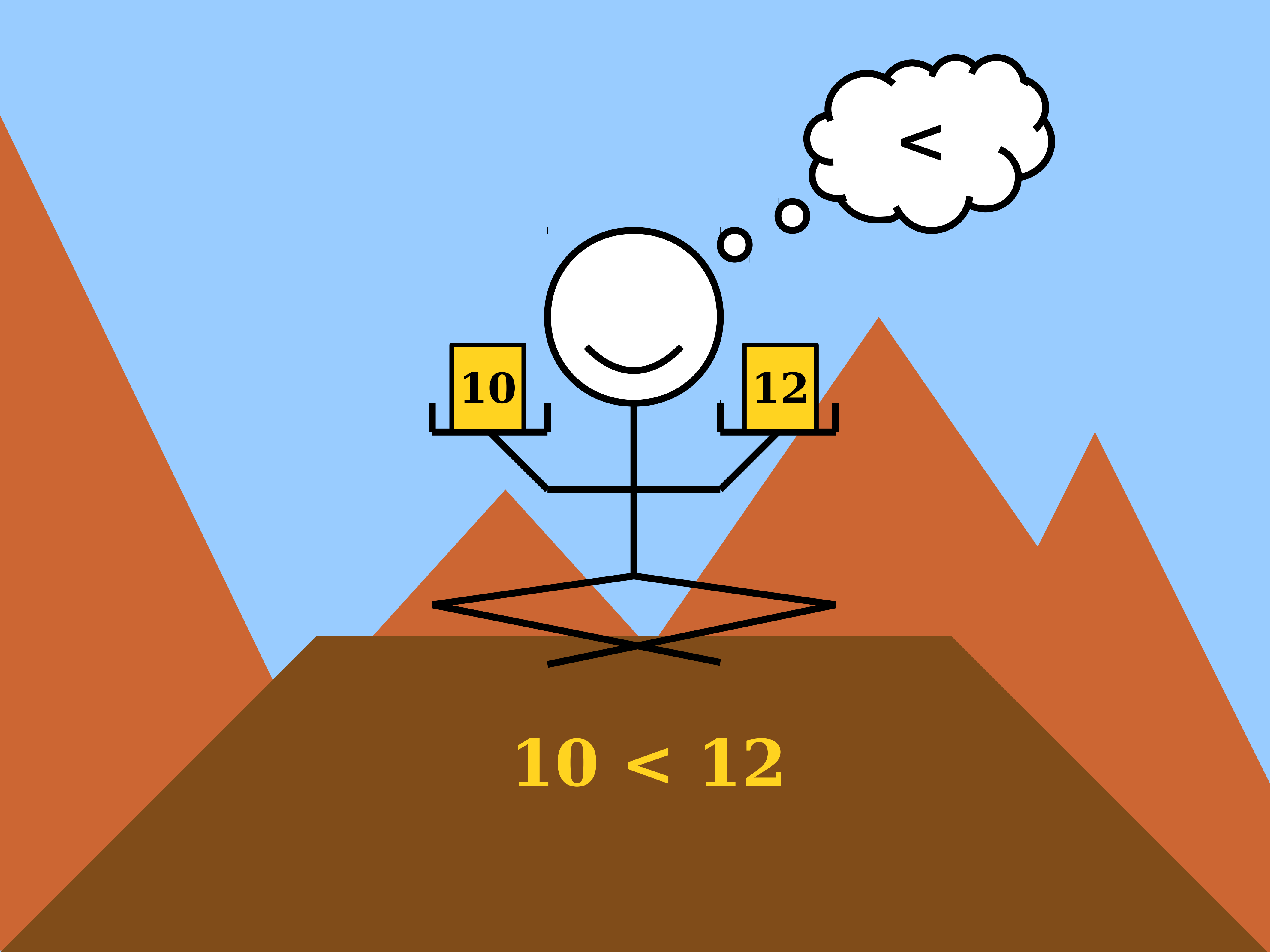
- Consider the set $\mathbb{N}$.
- We say $\leq$ is a ***binary relation over*** $\mathbb{N}$ because, given any $a, b \in \mathbb{N}$, the following questions are meaningful and have definitive yes or no answers:
  - Is $a \leq b$ true?
  - Is $b \leq a$ true?

Binary Relations

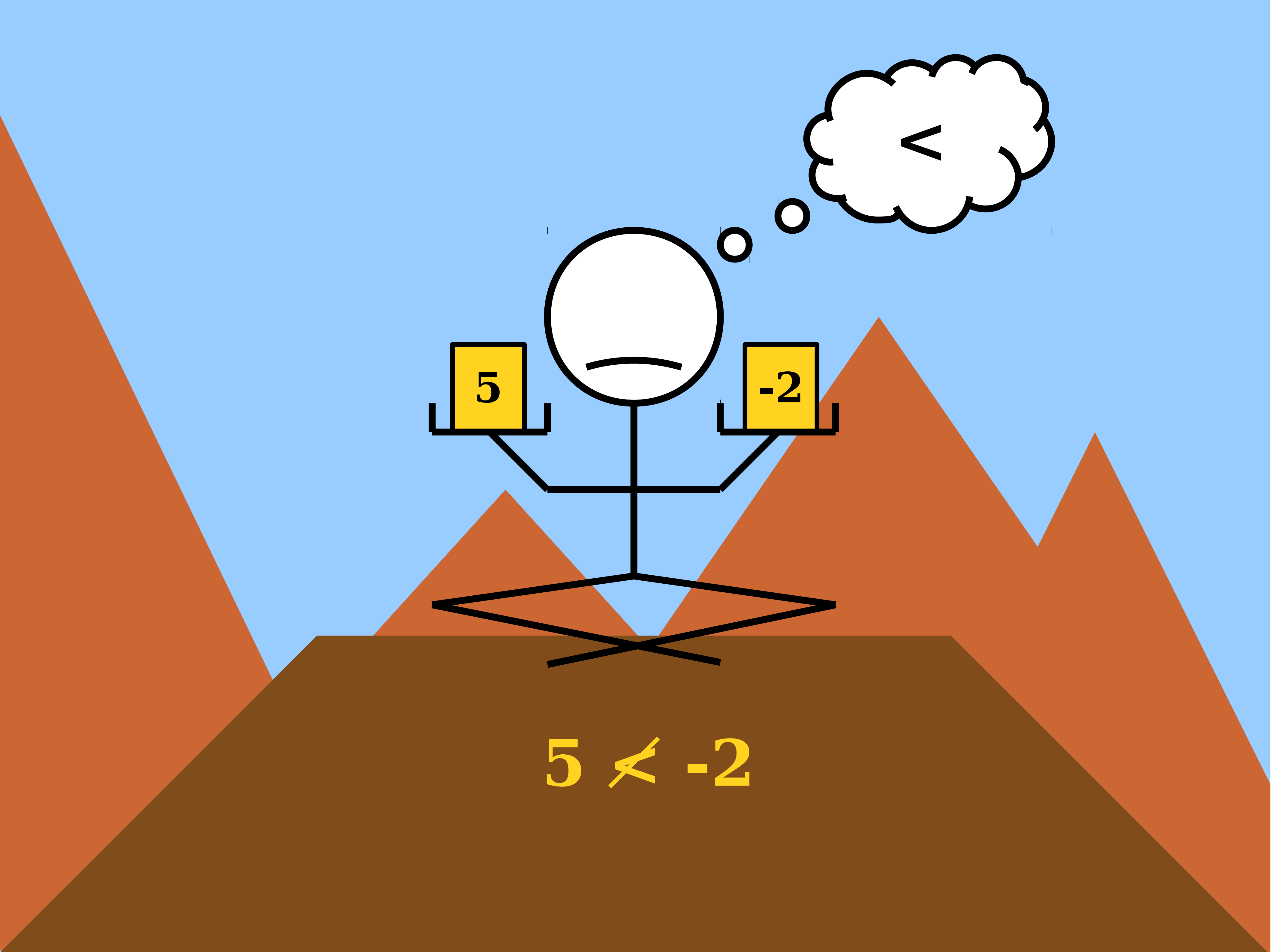
- Consider the set $\mathbb{Z}$.
- We say $\equiv_3$ is a ***binary relation over*** $\mathbb{Z}$ because, given any $a, b \in \mathbb{Z}$, the following questions are meaningful and have definitive yes or no answers:
  - Is $a \equiv_3 b$ true?
  - Is $b \equiv_3 a$ true?

Binary Relations

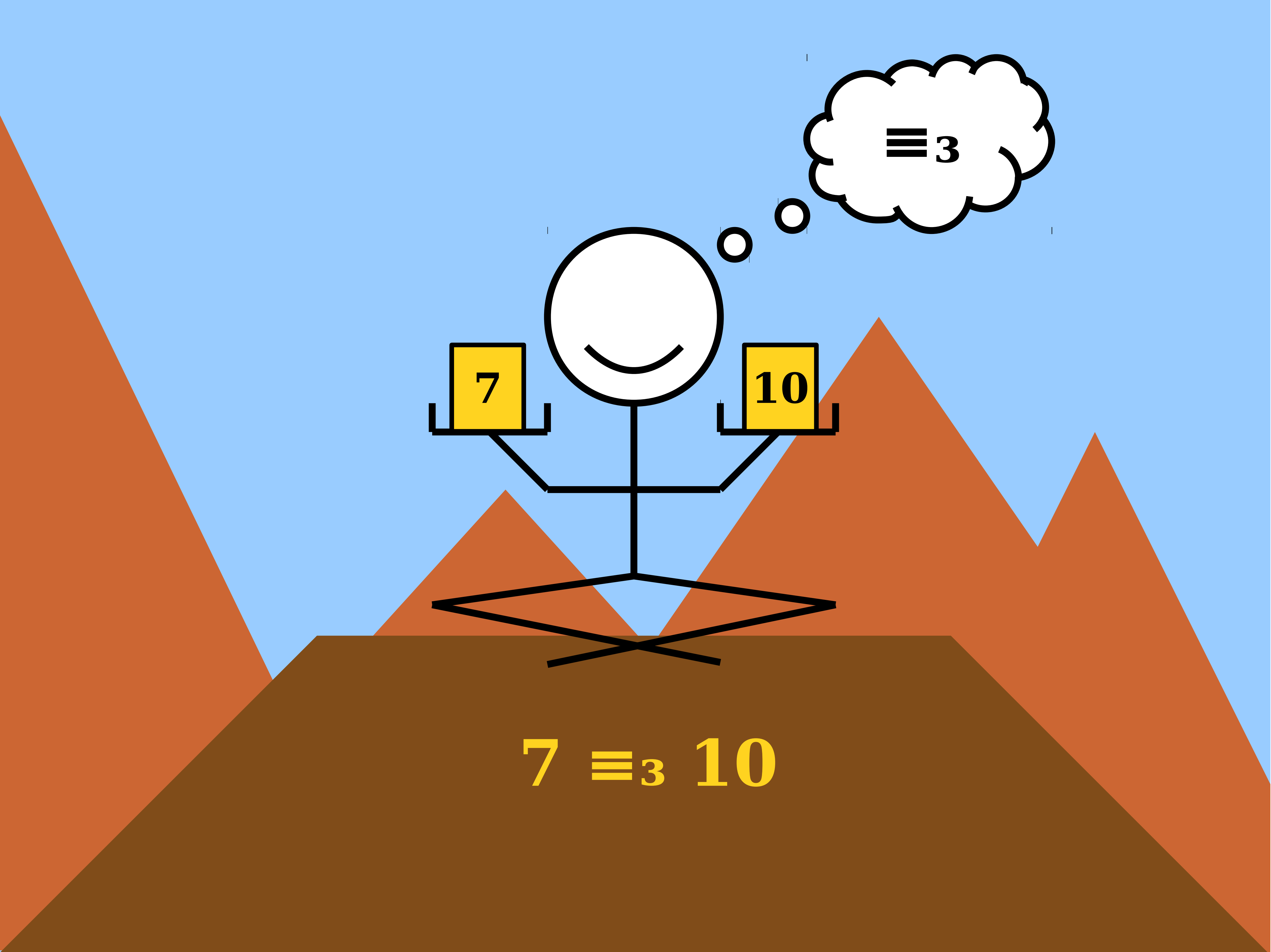
- Consider the set $A = \{a, b, c\}$.
- We say $=$ is a ***binary relation over A*** because, given any $a, b \in A$, the following questions are meaningful and have definitive yes or no answers:
  - Is $a = b$ true?
  - Is $b = a$ true?



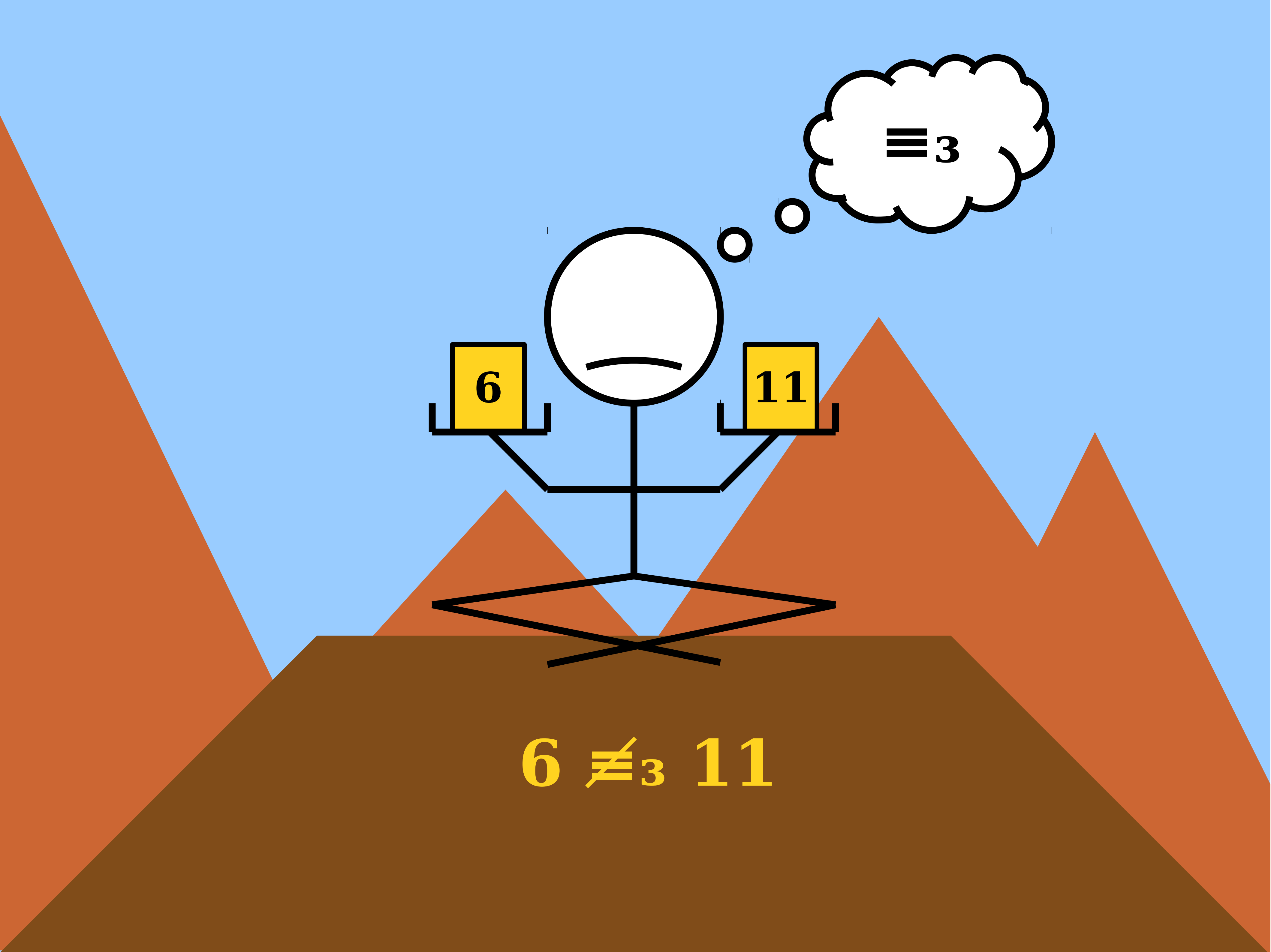
<
10
12
10 < 12

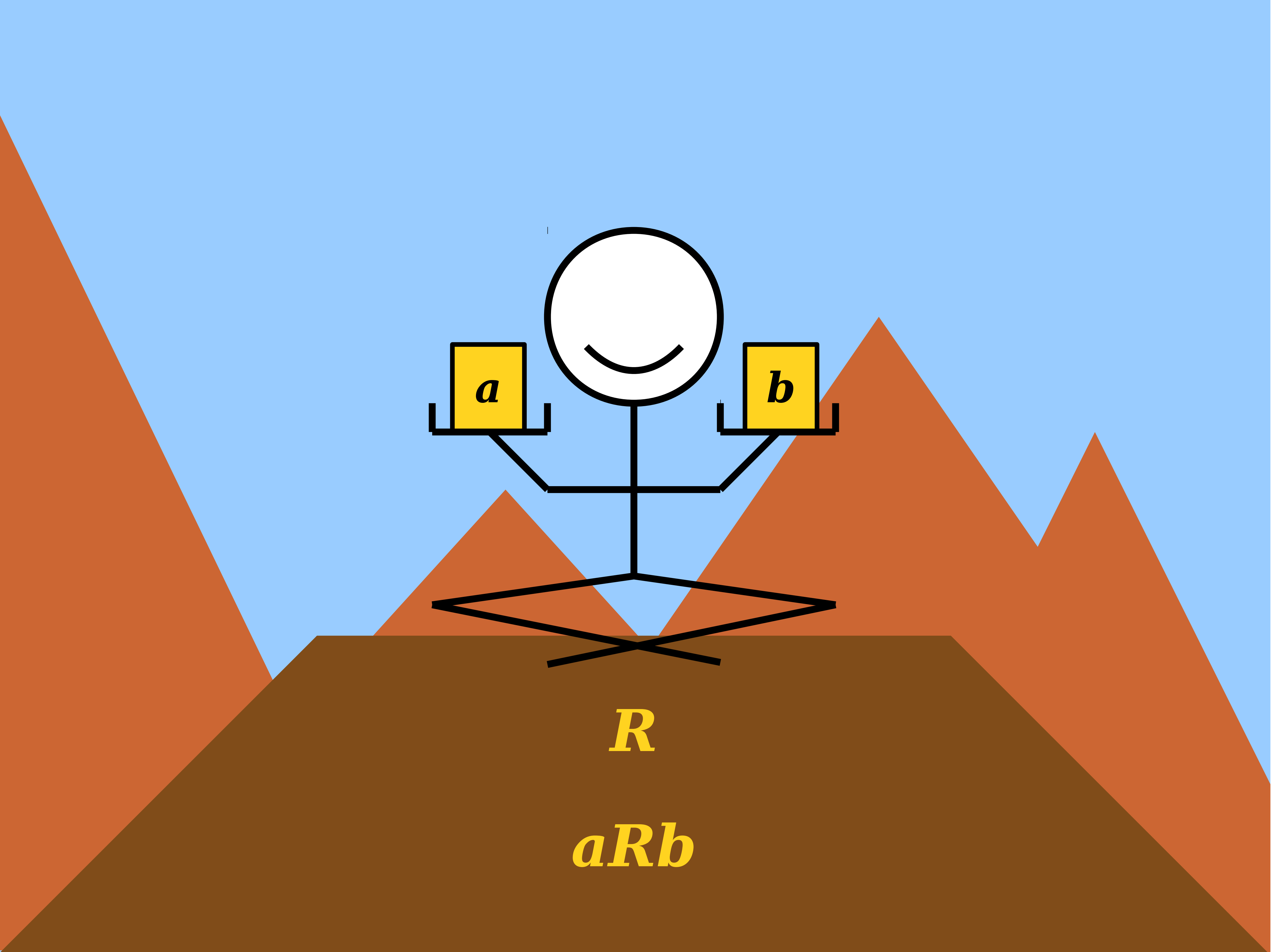


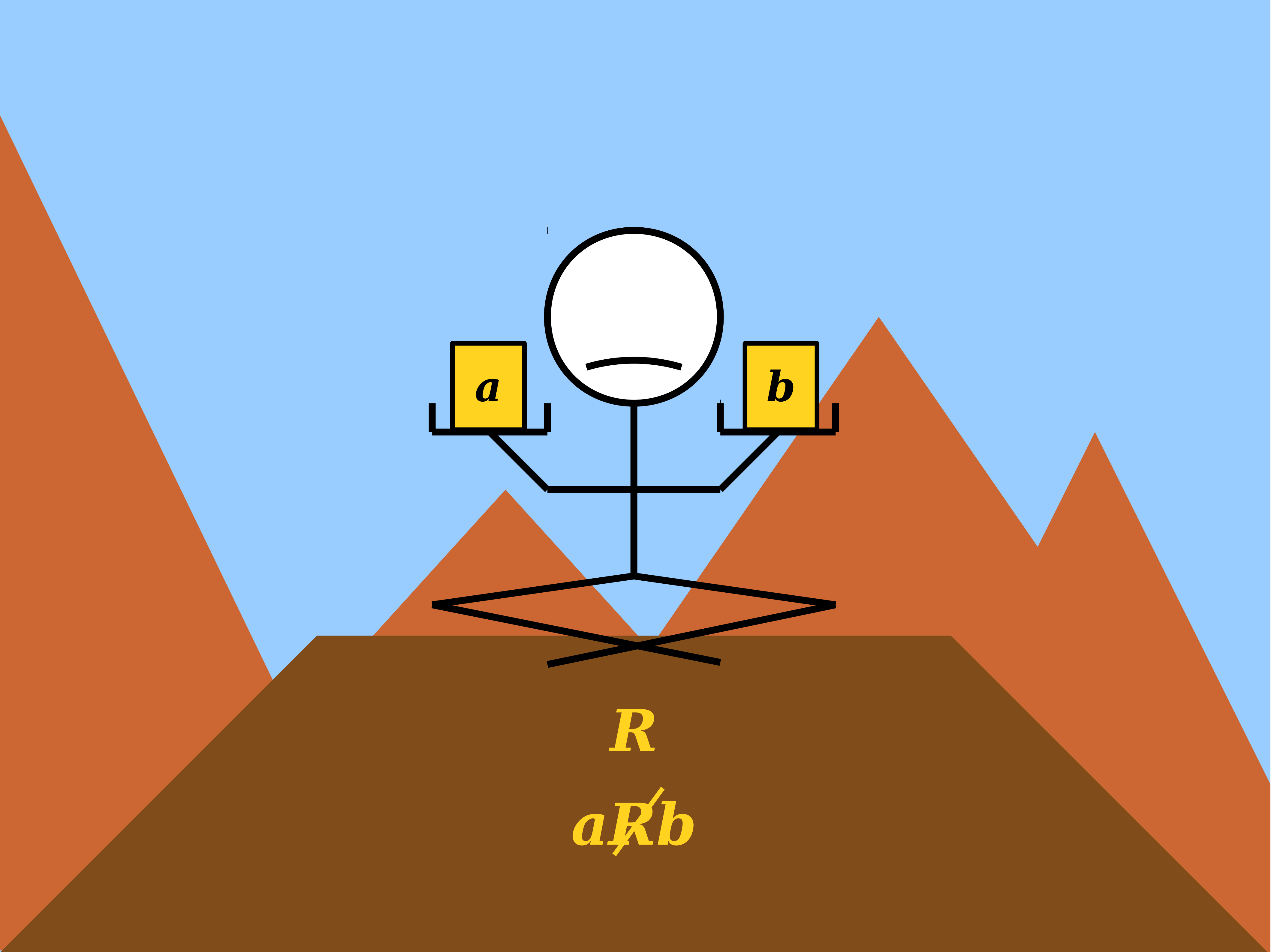
<
5
-2
5 ≮ -2



≡₃
7
10
7 ≡₃ 10


≡₃
6
11
6 ≢₃ 11


a
b
R
aRb

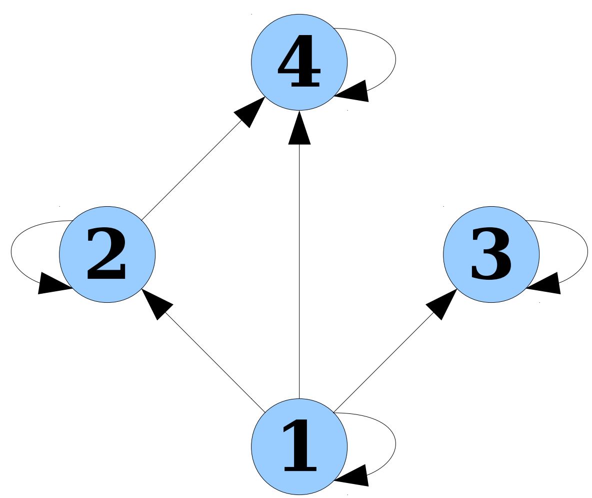

a
b
R
a$\not R$b

Binary Relations

- A ***binary relation over a set A*** is a structure that indicates properties about pairs of objects drawn from a set A.
- The particular property indicated depends on the choice of binary relation.
  - Some examples: $<$, $\leq$, $\subseteq$, $\equiv_k$, etc.
- If R is a binary relation over a set A and $a, b \in A$, we write aRb to indicate that the relation given by R holds from a to b.
  - For example, $42 < 137$, $\emptyset \subseteq \mathbb{N}$, etc.
- *Order matters in binary relations*. If we write aRb, we mean that a relation holds between a and b, in that order.
  - For example, $42 < 137$, but $137 \not< 42$.
  - It's possible that order doesn't matter ($5 \equiv_3 2$ and $2 \equiv_3 5$, for example), but it depends on the particular relation.
  - The relation isn't required to hold in either direction. For example, both of the statements $\{a, b\} \subseteq \mathbb{N}$ and $\mathbb{N} \subseteq \{a, b\}$ are false.

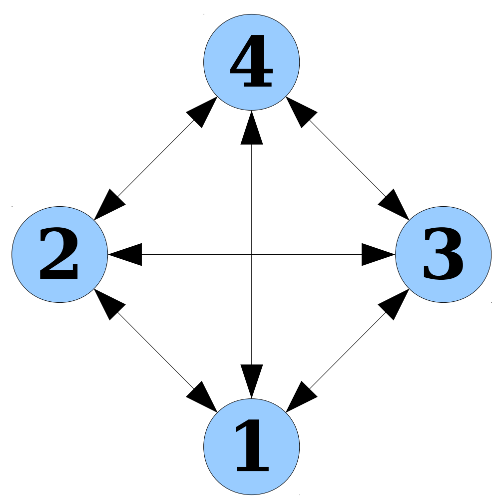
Visualizing Relations

- We can visualize a binary relation R over a set A by drawing the elements of A and drawing a line between an element a and an element b if aRb is true.
- Example: the relation $a \mid b$ (meaning "a divides b") over the set {1, 2, 3, 4} looks like this:



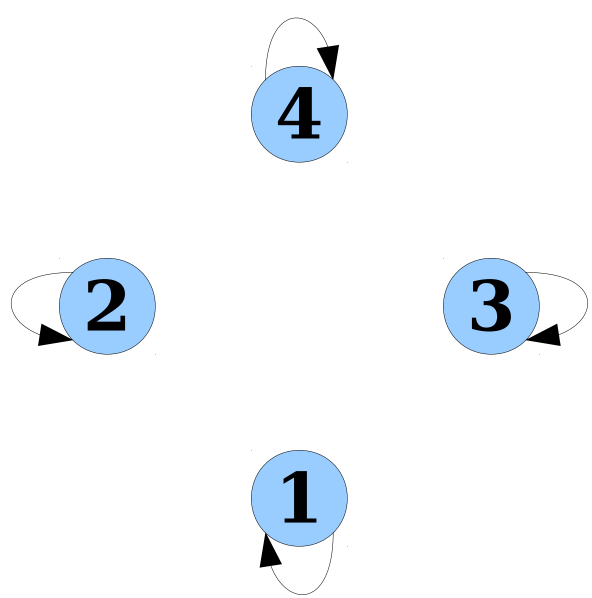
Visualizing Relations

- We can visualize a binary relation R over a set A by drawing the elements of A and drawing a line between an element a and an element b if aRb is true.
- Example: the relation $a \neq b$ over the set {1, 2, 3, 4} looks like this:



Visualizing Relations

- We can visualize a binary relation R over a set A by drawing the elements of A and drawing a line between an element a and an element b if aRb is true.
- Example: the relation $a = b$ over the set {1, 2, 3, 4} looks like this:

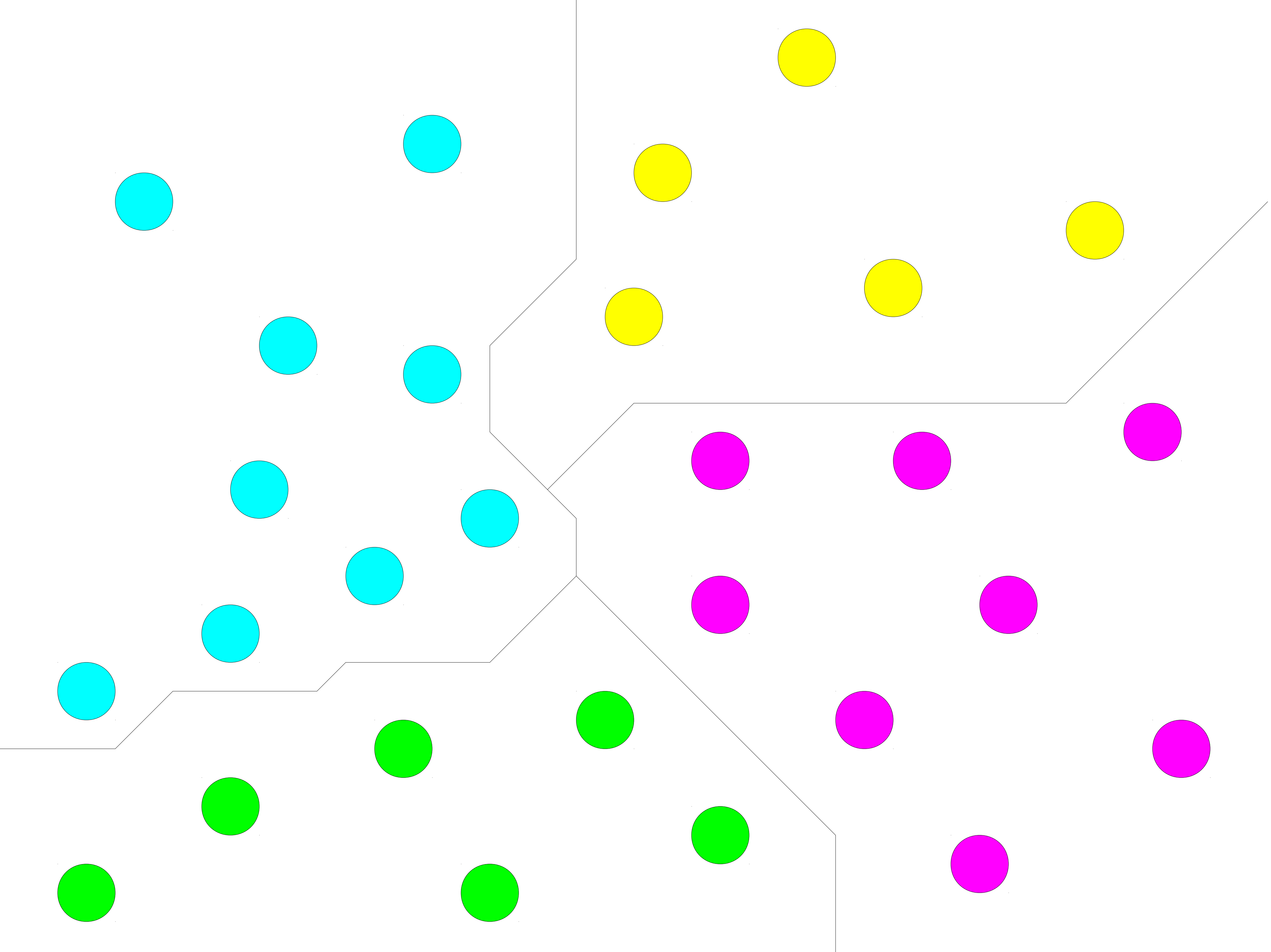


Capturing Structure

Capturing Structure

- Binary relations are an excellent way for capturing certain structures that appear in computer science.
- Today, we'll look at two of them (and possibly one more if we have time.)
  - **Partitions**
  - **Prerequisites**

Partitions



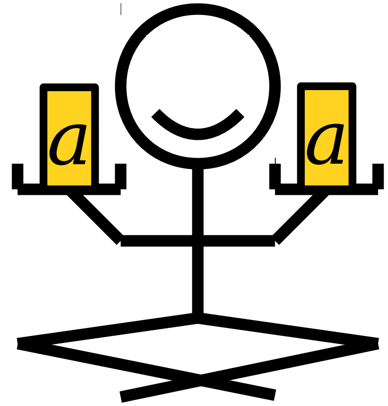
Partitions

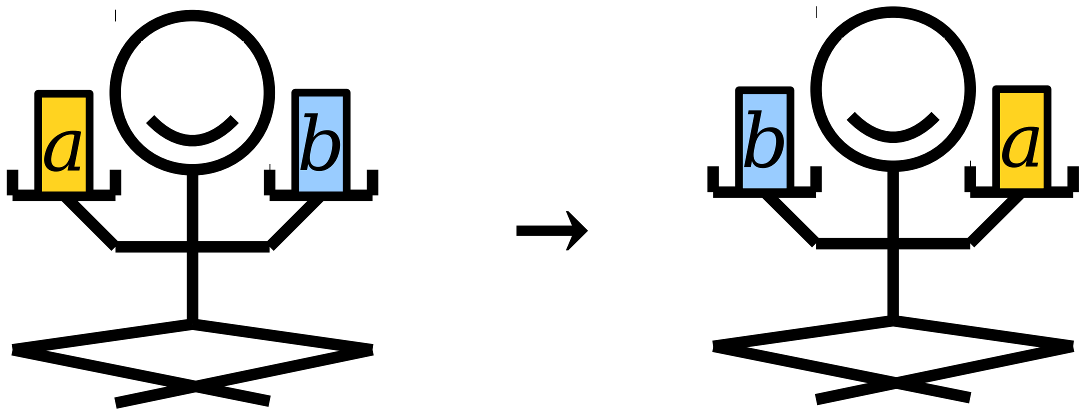
- A ***partition of a set*** is a way of splitting the set into disjoint, nonempty subsets so that every element belongs to exactly one subset.
- Intuitively, a partition of a set breaks the set apart into smaller pieces.
- There doesn't have to be any rhyme or reason to what those pieces are, though often there is one.

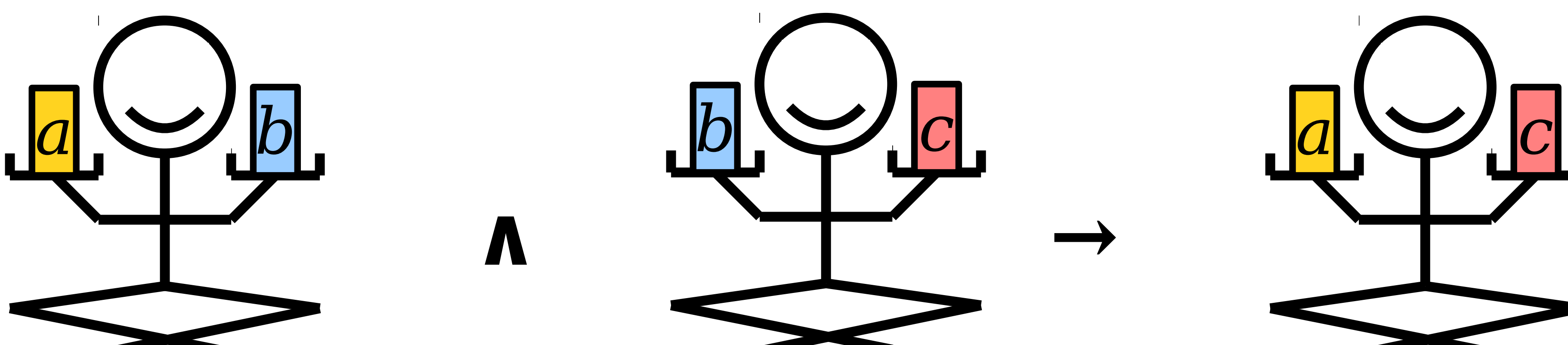
Partitions and Clustering

- Partitions are useful for data mining.
- If you have a set of data, you can often learn something from the data by finding a "good" clustering of that data and inspecting the clusters.
- Interested to learn more? Take CS161 or CS246!

What's the connection between partitions and binary relations?


a
a


a
b
b
a


a
b
b
c
a
c

$$\forall a \in A.\ aRa$$

$$\forall a \in A.\ \forall b \in A.\ (aRb \to bRa)$$

$$\forall a \in A. \forall b \in A. \forall c \in A. (aRb \land bRc \to aRc)$$

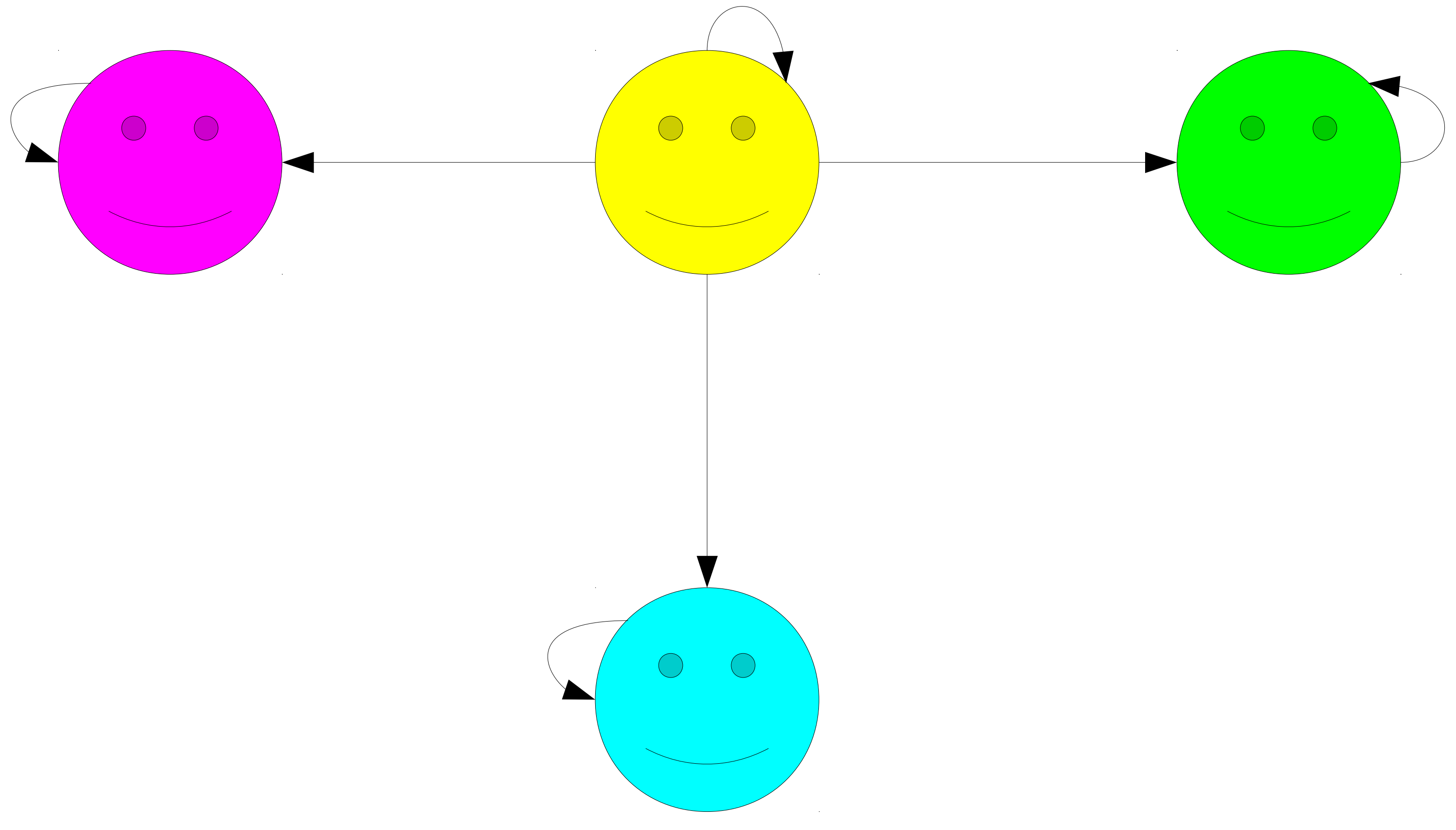
Reflexivity

- Some relations always hold from any element to itself.
- Examples:
  - $x = x$ for any x.
  - $A \subseteq A$ for any set A.
  - $x \equiv_k x$ for any x.
- Relations of this sort are called ***reflexive***.
- Formally speaking, a binary relation R over a set A is reflexive if the following is true:

$$\forall a \in A.\ aRa$$

("Every element is related to itself.")

Reflexivity Visualized



$$\forall a \in A.\ aRa$$

(*"Every element is related to itself."*)

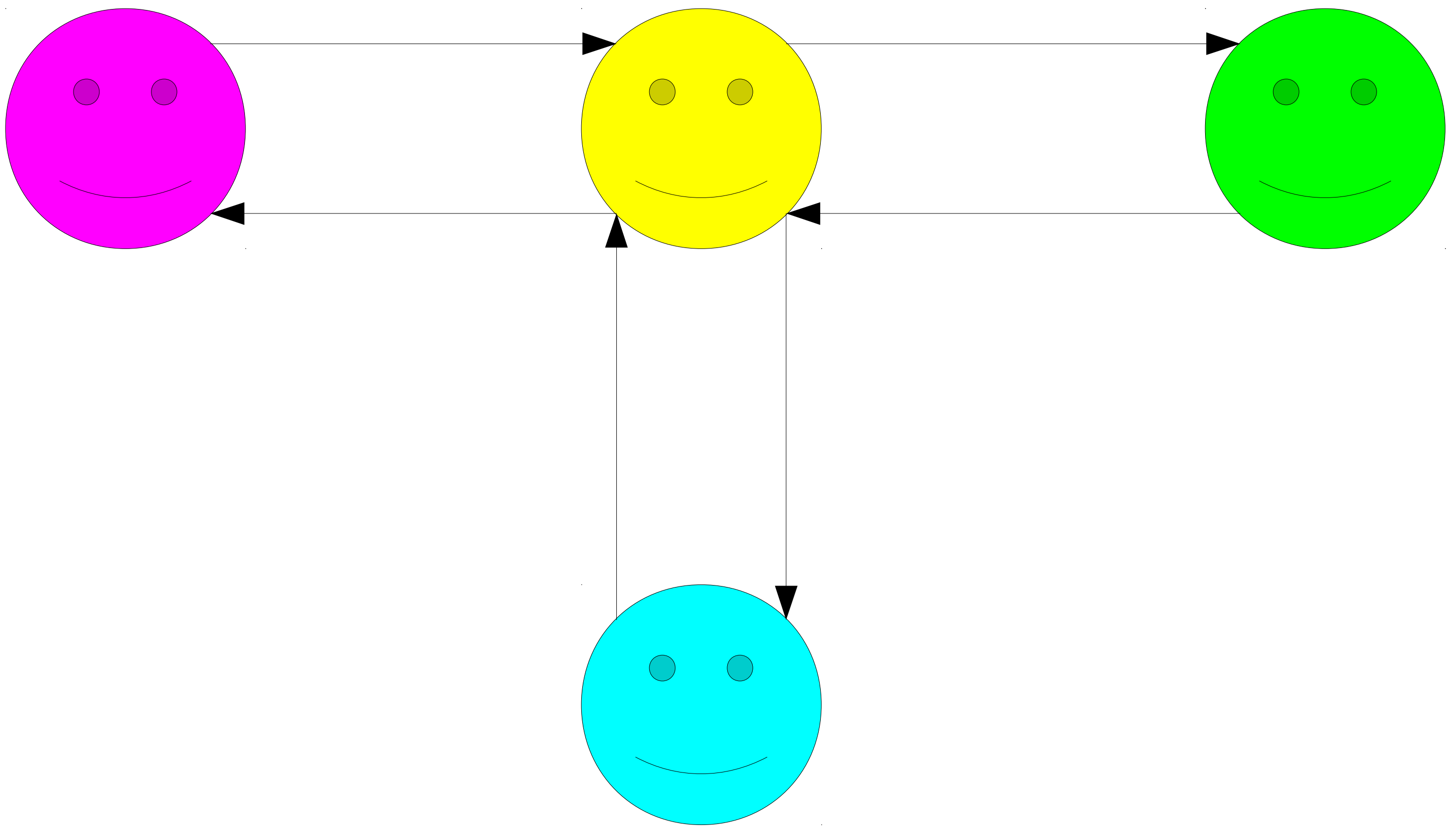
Symmetry

- In some relations, the relative order of the objects doesn't matter.
- Examples:
  - If $x = y$, then $y = x$.
  - If $x \equiv_k y$, then $y \equiv_k x$.
- These relations are called ***symmetric***.
- Formally: a binary relation R over a set A is called *symmetric* if

$$\forall a \in A.\ \forall b \in A.\ (aRb \to bRa)$$

("If a is related to b, then b is related to a.")

Symmetry Visualized



$$\forall a \in A. \forall b \in A. (aRb \rightarrow bRa)$$

(*"If a is related to b, then b is related to a."*)

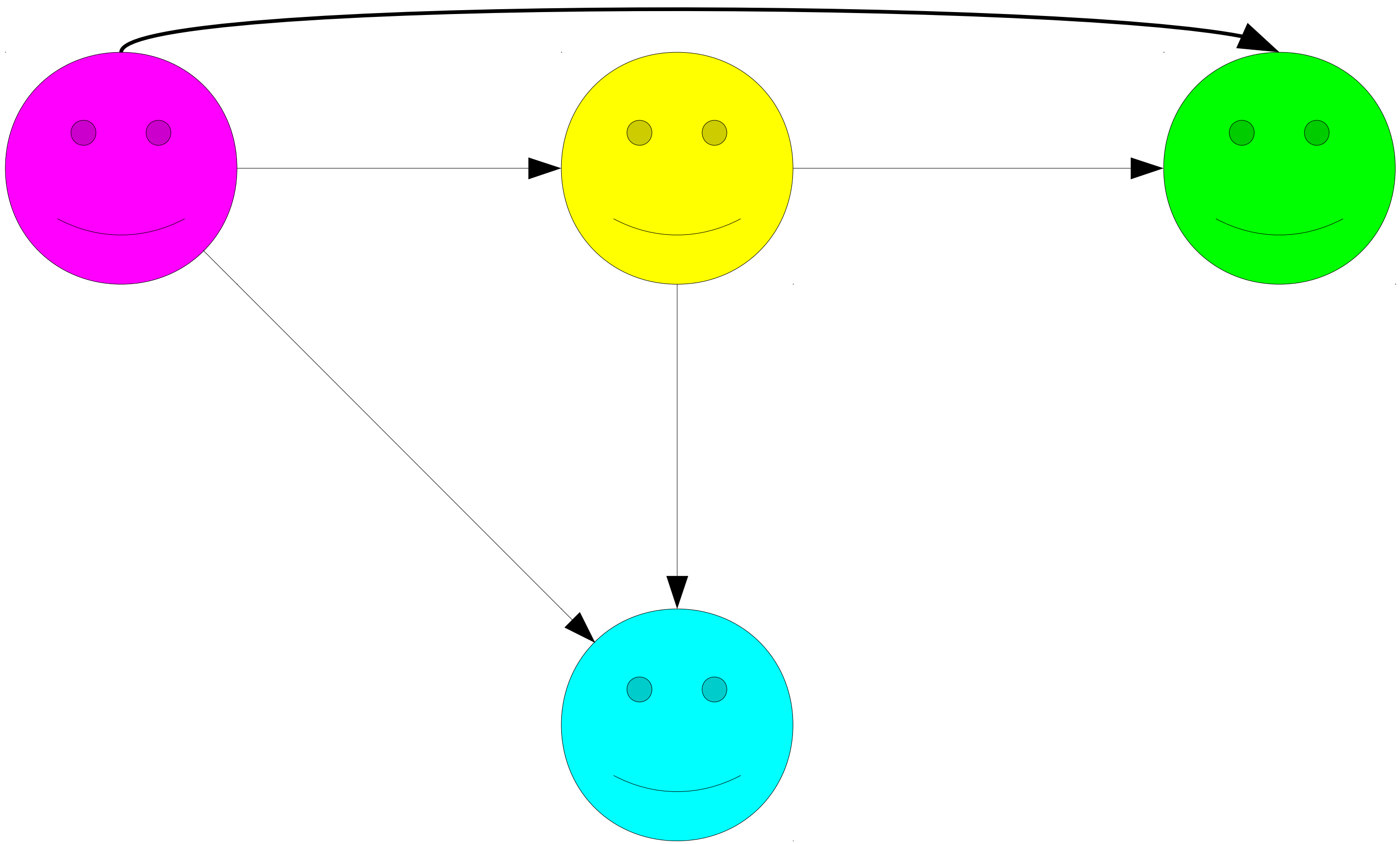
Transitivity

- Many relations can be chained together.
- Examples:
  - If $x = y$ and $y = z$, then $x = z$.
  - If $R \subseteq S$ and $S \subseteq T$, then $R \subseteq T$.
  - If $x \equiv_k y$ and $y \equiv_k z$, then $x \equiv_k z$.
- These relations are called ***transitive***.
- A binary relation R over a set A is called *transitive* if

$$\forall a \in A.\ \forall b \in A.\ \forall c \in A.\ (aRb \land bRc \to aRc)$$

(*"Whenever a is related to b and b is related to c, we know a is related to c.*)

Transitivity Visualized



$\forall a \in A. \forall b \in A. \forall c \in A. (aRb \land bRc \rightarrow aRc)$

(*"Whenever a is related to b and b is related to c, we know a is related to c.*)

Equivalence Relations

- An ***equivalence relation*** is a relation that is reflexive, symmetric and transitive.
- Some examples:
  - $x = y$
  - $x \equiv_k y$
  - x has the same color as y
  - x has the same shape as y.

Time-Out for Announcements!

Problem Set Two

- Problem Set Two checkpoint was due at the start of class; we'll try to get it graded and returned by Wednesday.
- Problem Set One solutions are available now; if you don't get them today in class, you can pick them up in the return drawer in Gates.
- Rest of Problem Set Two is due on Friday. Stop by office hours with questions, or ask on Piazza!

{w} Stanford Women in Computer Science

WICS PRESENTS

My Stanford CS Journey: A Student Panel and Mixer

Wednesday, Oct 7th

5:30-7:00pm @ Gates 219

Dinner is provided

Are you excited to explore the multitude of CS opportunities that Stanford has to offer? Want to hear from your peers about their journey through CS? Or just want to hang out with other super passionate CS folks with free food? Come for the student panel and undergrad mixer!

WICS.STANFORD.EDU **FACEBOOK.COM/STANFORDWICS** **WICS-BOARD@LISTS.STANFORD.EDU**

Your Questions

There were a total of 13 questions – wow! If I don't answer your question, feel free to ask it again next time.

“What are some of the popular real world applications for the topics we discuss in CS103?”

The material from the back half of this course is critical in the design of compilers and programming languages. If you take CS143, you'll see why this is.

The discrete math topics we'll see are the mathematical basis for algorithm analysis and design – take CS161 for details!

Many of the concepts we'll see here generalize to concepts you'll see in AI and natural language processing. Take CS124 for details!

“Why do you use OpenOffice as opposed to the Microsoft suite?”

I use Linux because it's really nice for development, so that kinda rules out Microsoft Office. Nothing personal again Microsoft, I promise!

“What do you think about work-life balance?”

Everyone is different, but everyone should try to find a way to strike a balance between the two. Your future self will thank you. ☺

It's tough to do this while you're living on campus because there's such a tight coupling between your work life and your social life. When hunting for jobs, though, ask people about how long they work and what they do for fun. The answers can be very revealing.

“If you hadn't majored in CS, what would you have studied?”

Probably either aero/astro (you get to put things in space!), math (I really like the sorts of results you discover there), or law (uses a lot of the same skills, honestly.)

"What do you think is the best way for CS students interested in social impact and social justice to get involved (and why do you think a lot of students don't seem to care about much else than making $$)?"

I don't think there are as many profit-minded students as you'd expect, and many of those who are have really goood reasons for it.

I think it's just a matter of what people's experiences are. If you don't take a lot of humanities/social science classes, travel the world, or live in a big city, it's easy to lose sight of what a lot of the important problems are.

If you're interested in getting involved, get on the CS+SG mailing list, contact the Haas center, look into SIG fellowships, or come talk to me!

Back to CS103!

Equivalence Relation Proofs

- Let's suppose you've found a binary relation R over a set A and want to prove that it's an equivalence relation.
- How exactly would you go about doing this?

An Example Relation

- Consider the following binary relation ~ defined over the set ℤ:

$$a \sim b \quad \textbf{if} \quad a+b \textbf{ is even}$$

- (Notice that we use "if" rather than "iff" in this definition. When defining a relation, the convention is to use "if." See the Mathematical Vocabulary handout for more details.)
- Some examples:

$$0 \sim 4 \quad 1 \sim 9 \quad 2 \sim 6 \quad 5 \sim 5$$

- Turns out, this is an equivalence relation! Let's see how to prove it.

What properties must ~ have to be an equivalence relation?

Reflexivity
Symmetry
Transitivity

Let's prove each property independently.

$a \sim b$ if $a+b$ is even

Lemma 1: The binary relation ~ is reflexive.

Proof:

What is the formal definition of reflexivity?

$$\forall a \in \mathbb{Z}.\ a \sim a$$

Therefore, we'll choose an arbitrary integer a, then go prove that $a \sim a$.

$a \sim b$ if $a+b$ is even

Lemma 1: The binary relation $\sim$ is reflexive.

Proof: Consider an arbitrary $a \in \mathbb{Z}$. We need to prove that $a \sim a$. From the definition of the $\sim$ relation, this means that we need to prove that $a+a$ is even.

To see this, notice that $a+a = 2a$, so the sum $a+a$ can be written as $2k$ for some integer k (namely, a), so $a+a$ is even. Therefore, $a \sim a$ holds, as required. ■

$a \sim b$ if $a+b$ is even

Lemma 2: The binary relation ~ is symmetric.

Proof:

What is the formal definition of symmetry?

$$\forall a \in \mathbb{Z}.\ \forall b \in \mathbb{Z}.\ (a \sim b \to b \sim a)$$

Therefore, we'll choose arbitrary integers a and b where $a \sim b$, then prove that $b \sim a$.

***a~b* if *a+b* is even**

Lemma 2: The binary relation ~ is symmetric.

Proof: Consider any integers *a* and *b* where *a~b*. We need to show that *b~a*.

Since *a~b*, we know that *a+b* is even. Because *a+b* = *b+a*, this means that *b+a* is even. Since *b+a* is even, we know that *b~a*, as required. ■

$a \sim b$ if $a+b$ is even

Lemma 3: The binary relation ~ is transitive.

Proof:

What is the formal definition of transitivity?

$$\forall a \in \mathbb{Z}.\ \forall b \in \mathbb{Z}.\ \forall c \in \mathbb{Z}.\ (a \sim b \land b \sim c \to a \sim c)$$

Therefore, we'll choose arbitrary integers a, b, and c where $a \sim b$ and $b \sim c$, then prove that $a \sim c$.

***a*~*b* if *a*+*b* is even**

Lemma 3: The binary relation ~ is transitive.

Proof: Consider arbitrary integers a, b and c where $a \sim b$ and $b \sim c$. We need to prove that $a \sim c$, meaning that we need to show that $a+c$ is even.

Since $a \sim b$ and $b \sim c$, we know that $a \sim b$ and $b \sim c$ are even. This means there are integers k and m where $a+b = 2k$ and $b+c = 2m$. Notice that

$$(a+b) + (b+c) = 2k + 2m.$$

Rearranging, we see that

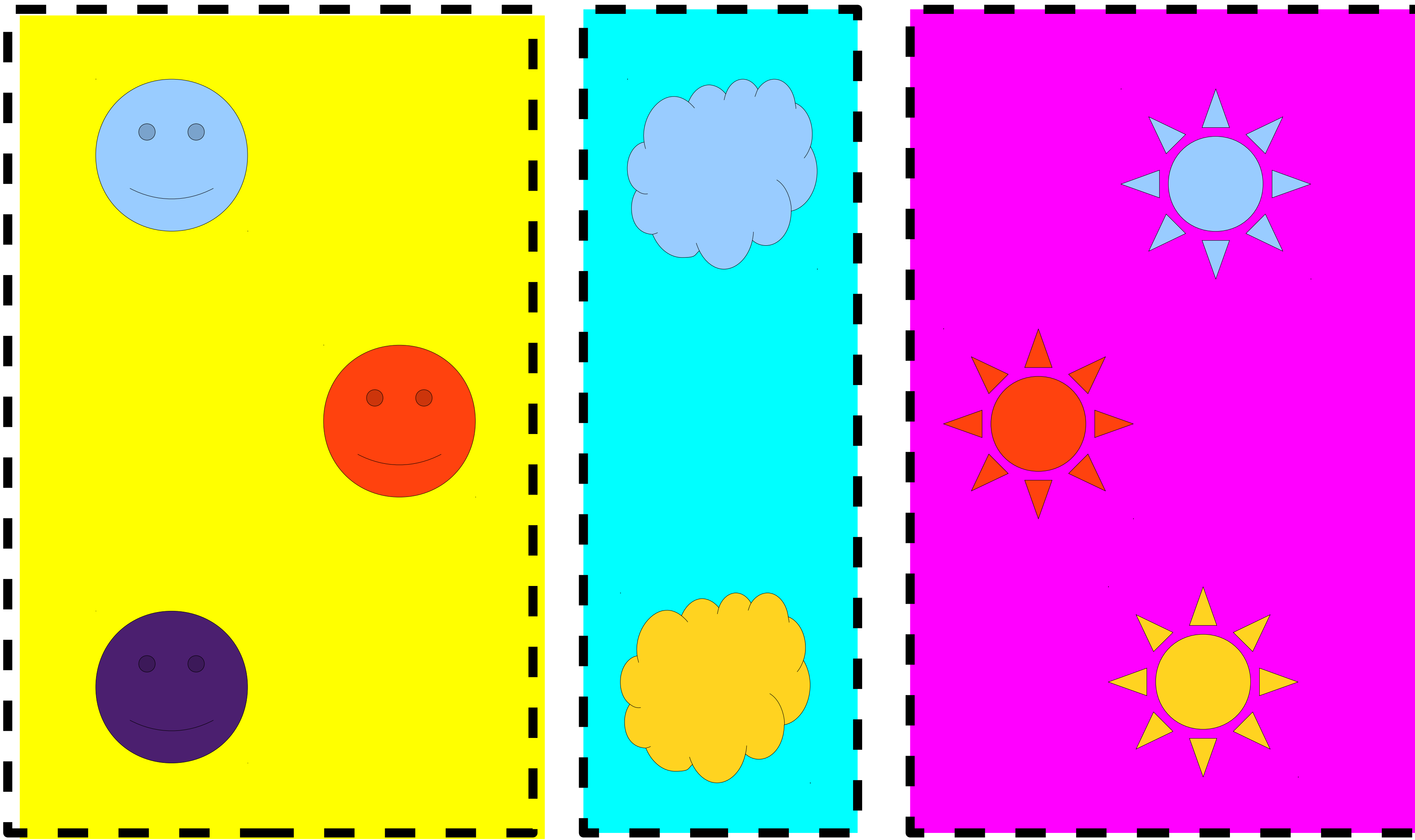
$$a+c + 2b = 2k + 2m,$$

so

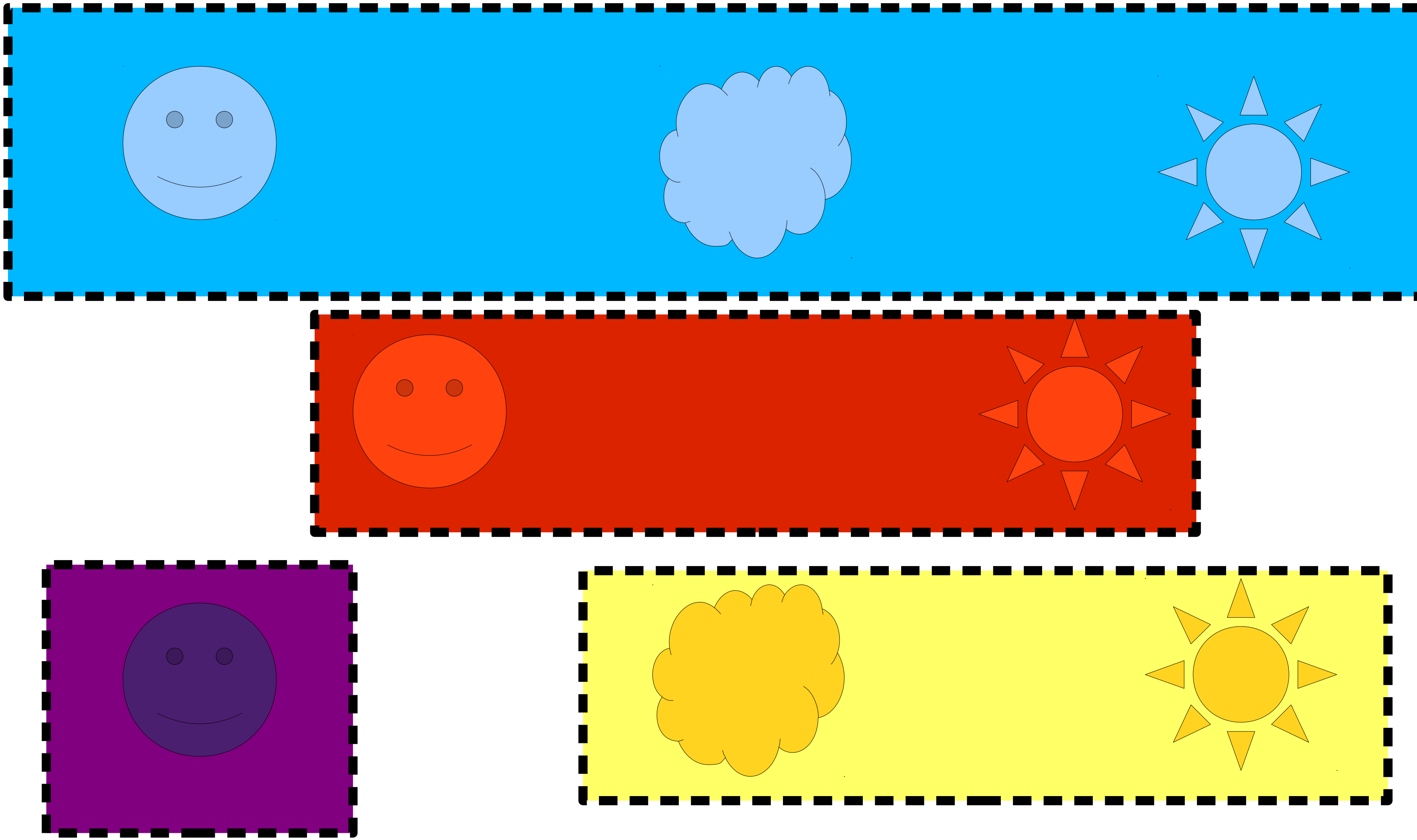
$$a+c = 2k + 2m - 2b = 2(k+m-b).$$

So there is an integer r, namely $k+m-b$, such that $a+c = 2r$. Thus $a+c$ is even, so $a \sim c$, as required. ■

Properties of Equivalence Relations



Let R be “has the same shape as”



Let T be "is the same **color** as"

Equivalences and Partitions

- Our definition of equivalence relations was motivated by the idea of partitioning elements into groups.

Partition of Elements ⇒ Equivalence Relation

- In the previous examples, we've seen several equivalence relations that then give rise to a partition. It turns out that this is not a coincidence!

Equivalence Relation ⇒ Partition of Elements

Equivalence Classes

- Given an equivalence relation R over a set A, for any $x \in A$, the ***equivalence class of x*** is the set

$$[x]_R = \{ y \in A \mid xRy \}$$

- $[x]_R$ is the set of all elements of A that are related to x by relation R.
- ***Theorem:*** If R is an equivalence relation over A, then every $a \in A$ belongs to exactly one equivalence class.