Binary Relations

Part One

Outline for Today

- ***Binary Relations***
  - Reasoning about connections between objects.
- ***Equivalence Relations***
  - Reasoning about clusters.
- ***A Fundamental Theorem***
  - How do we know we have the “right” definition for something?

Relationships

- In CS103, you've seen examples of relationships
  - between sets:

$$A \subseteq B$$

  - between numbers:

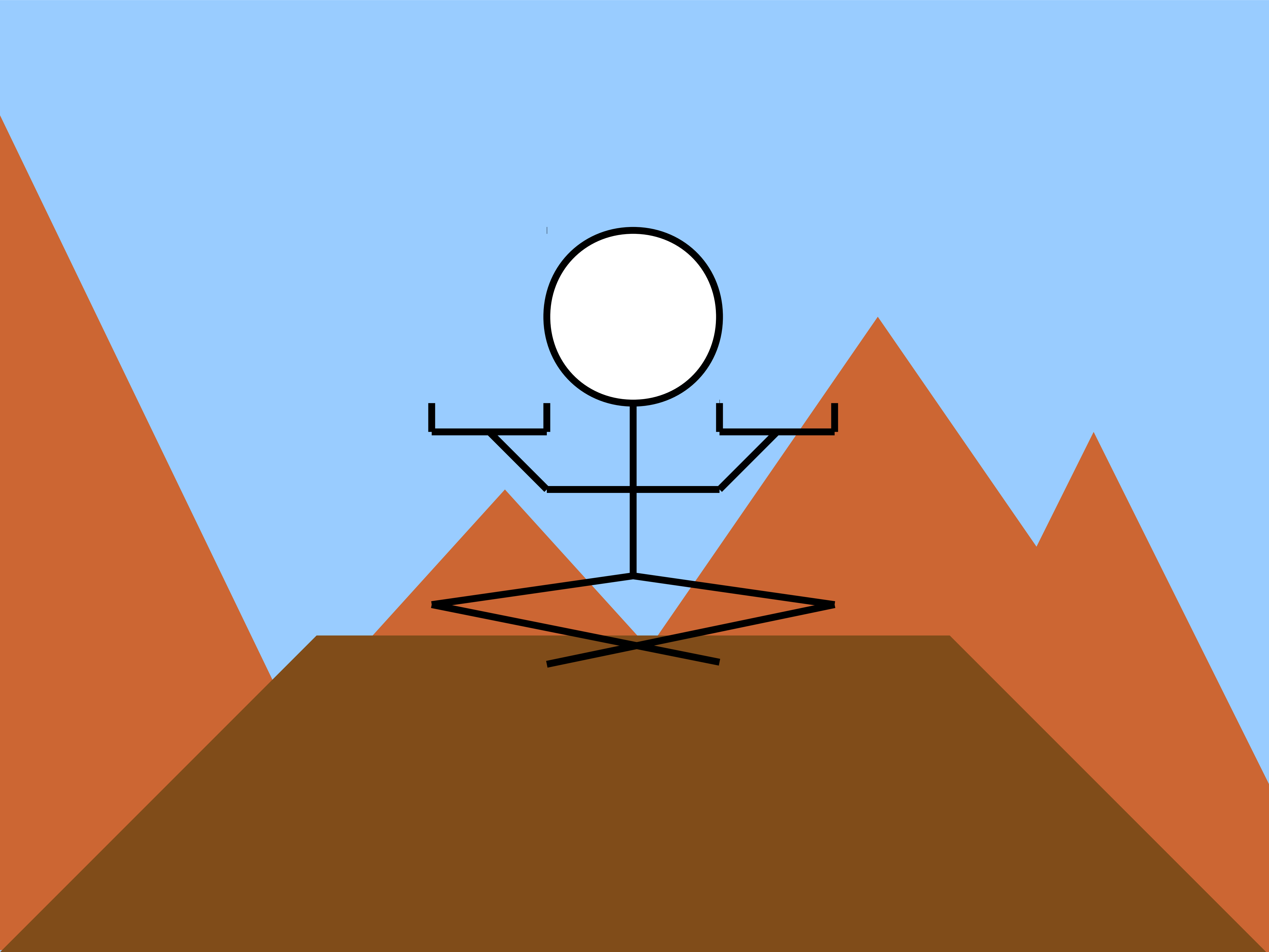
$$x < y \qquad x \equiv_k y \qquad x \leq y$$

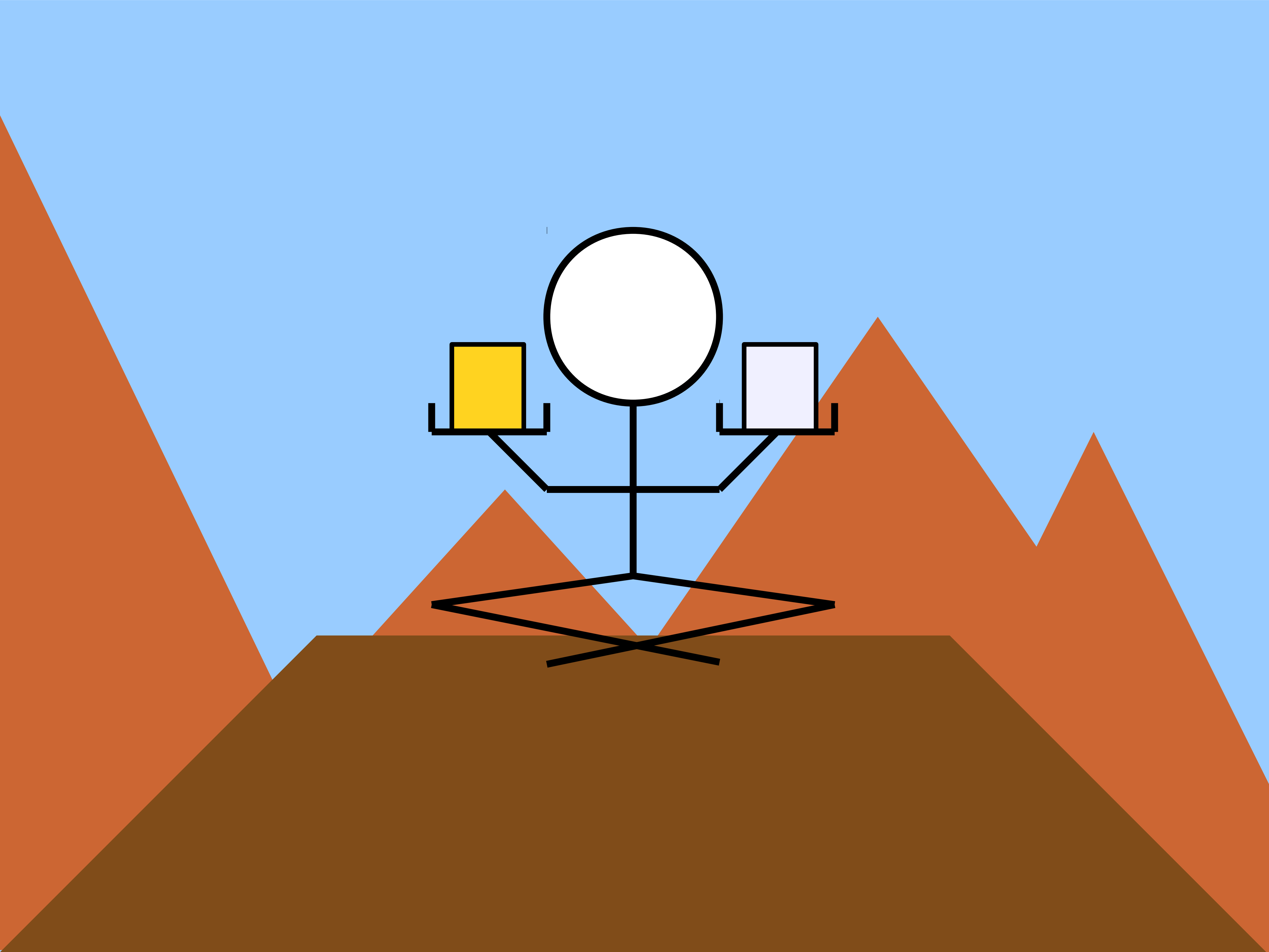
  - between people:

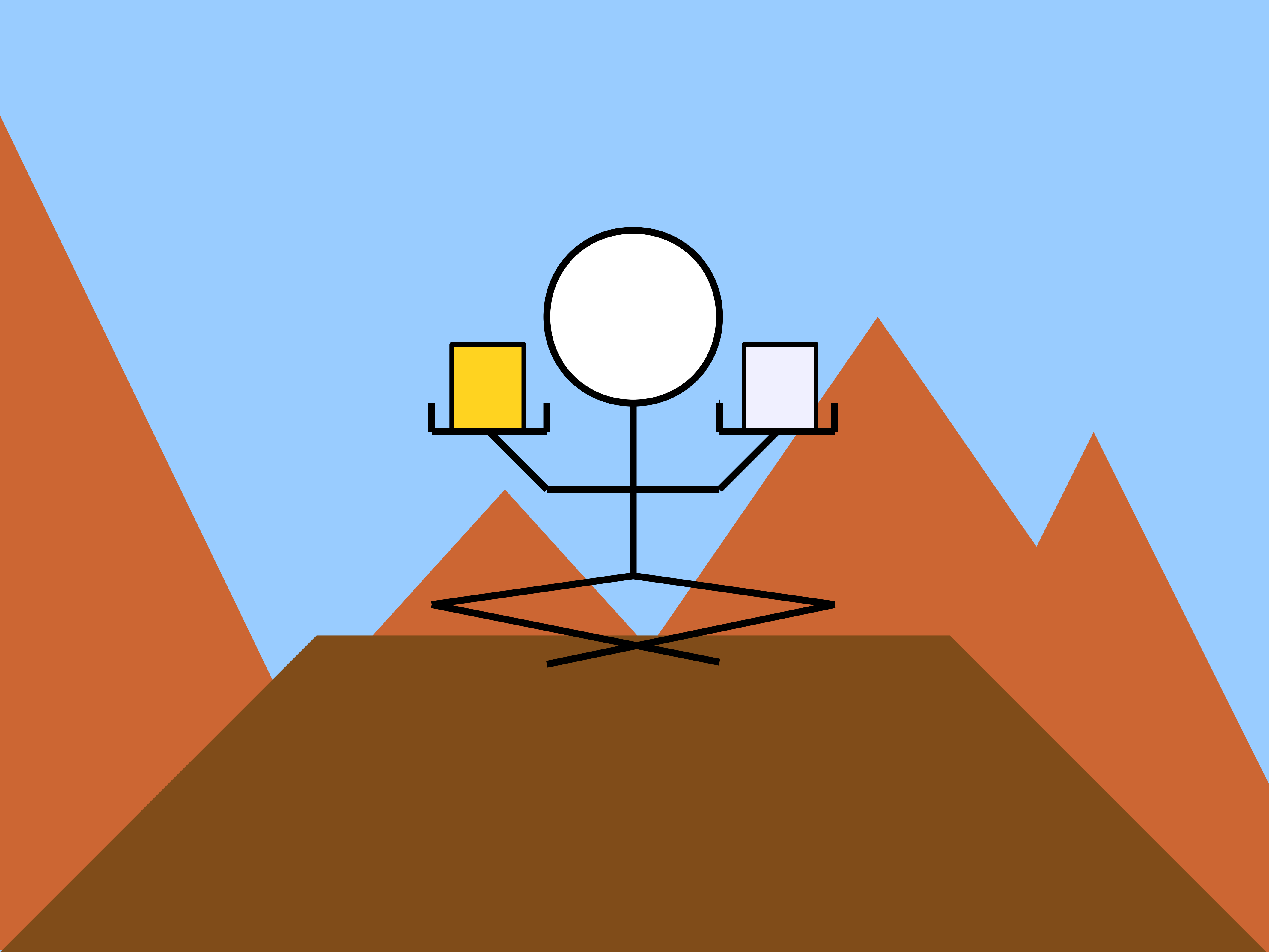
$$p \text{ loves } q$$

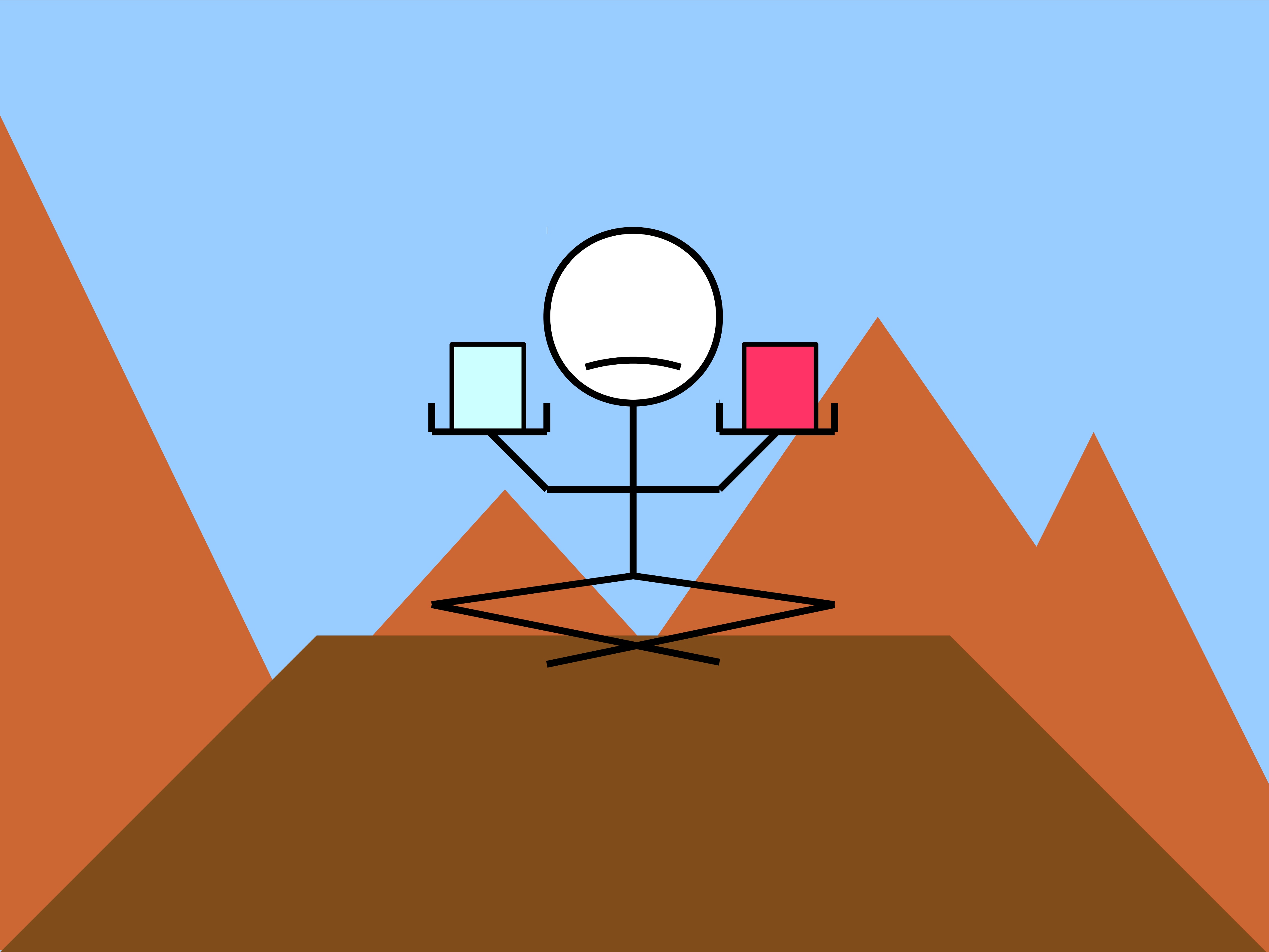
- Since these relations focus on connections between two objects, they are called ***binary relations***.
  - The "binary" here means "pertaining to two things," not "made of zeros and ones."

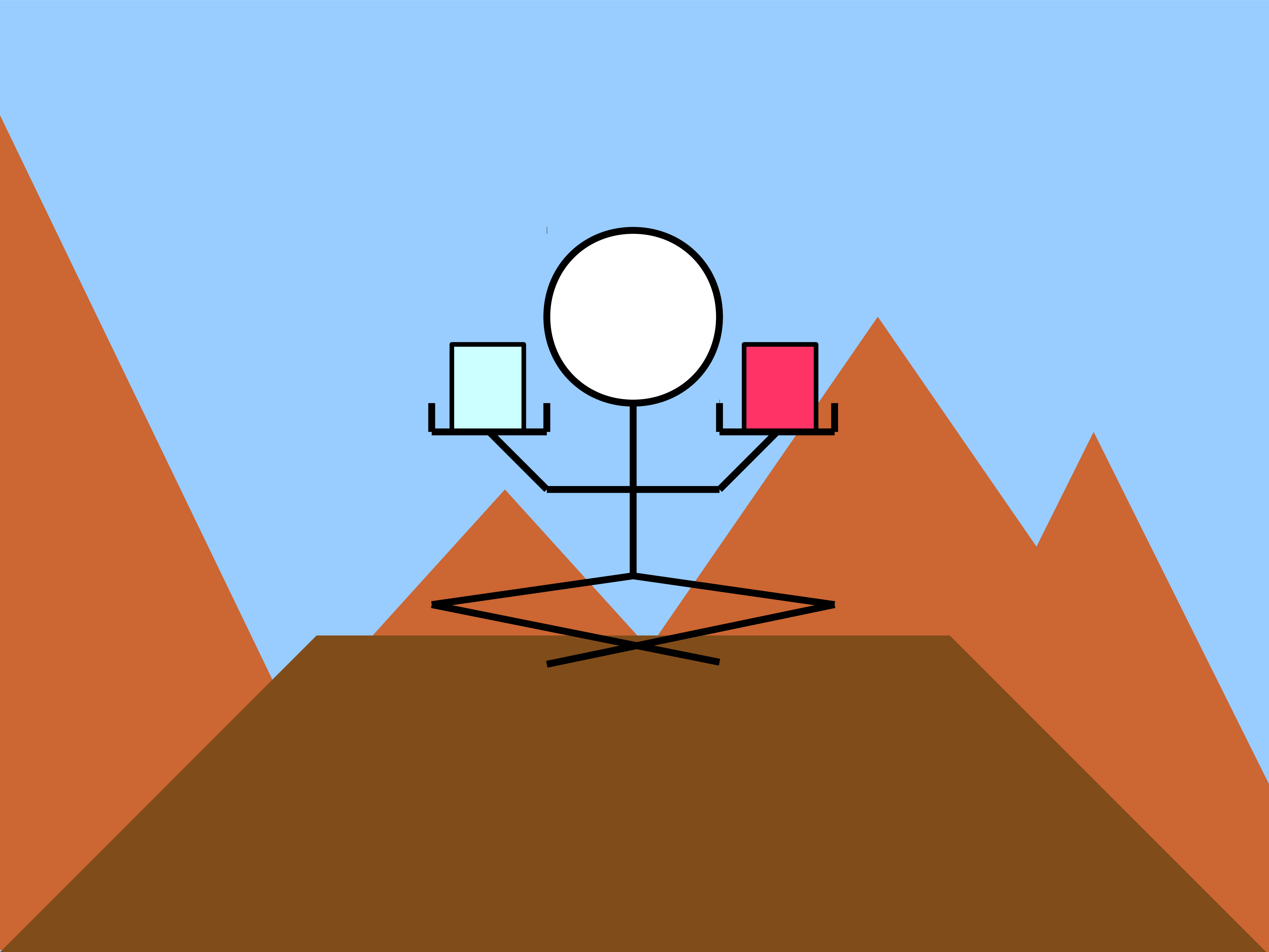
What exactly is a binary relation?

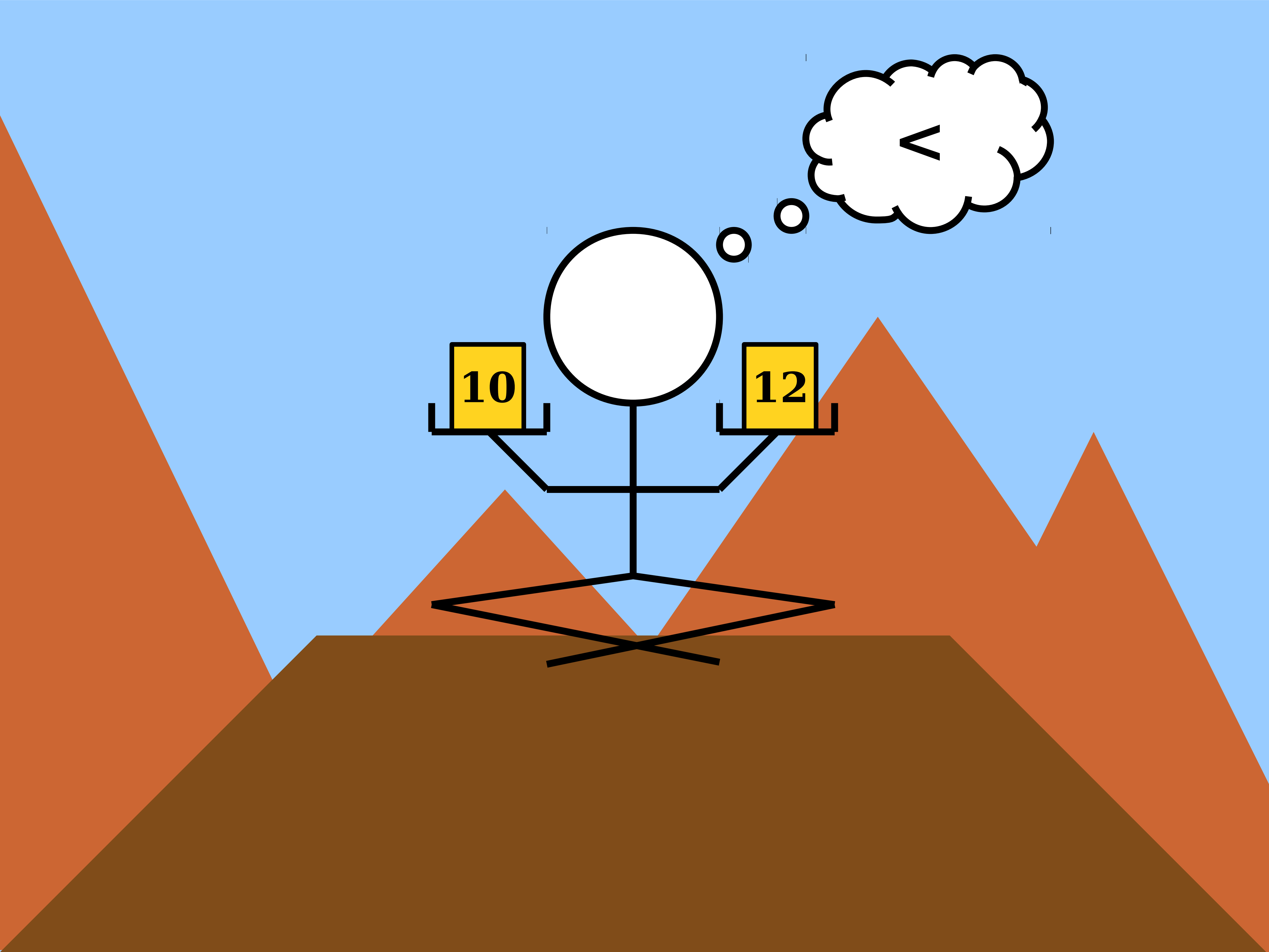


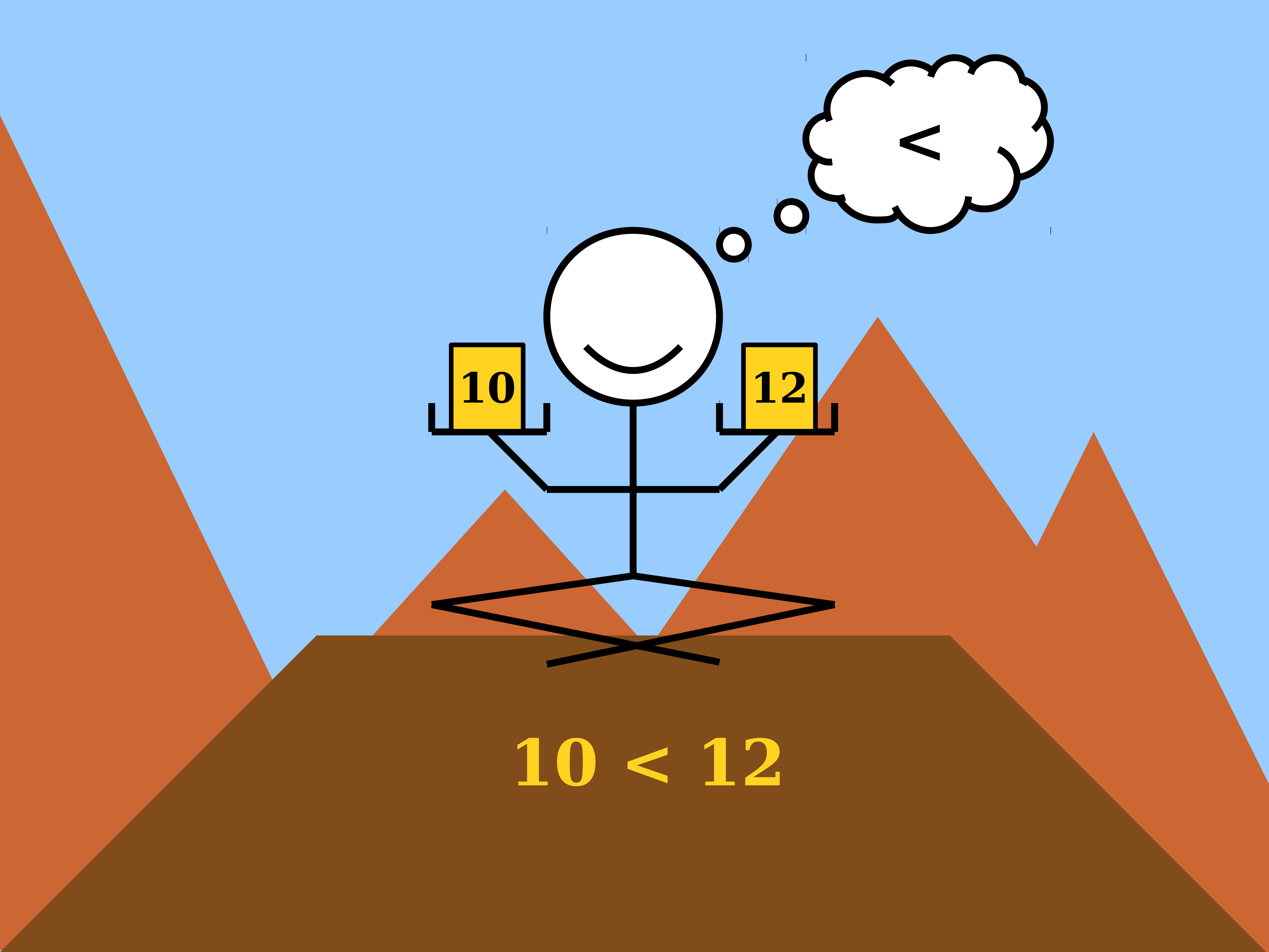


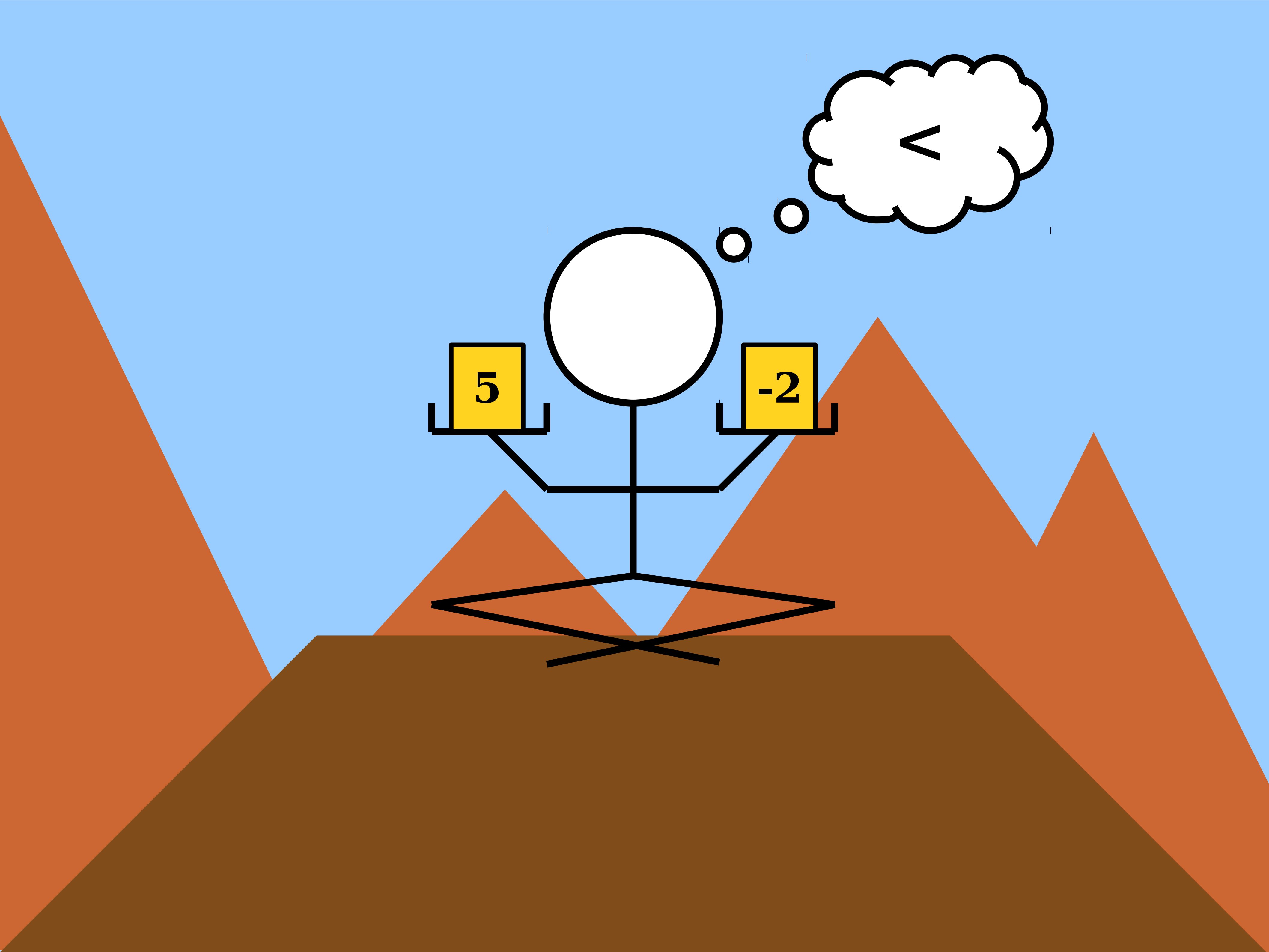


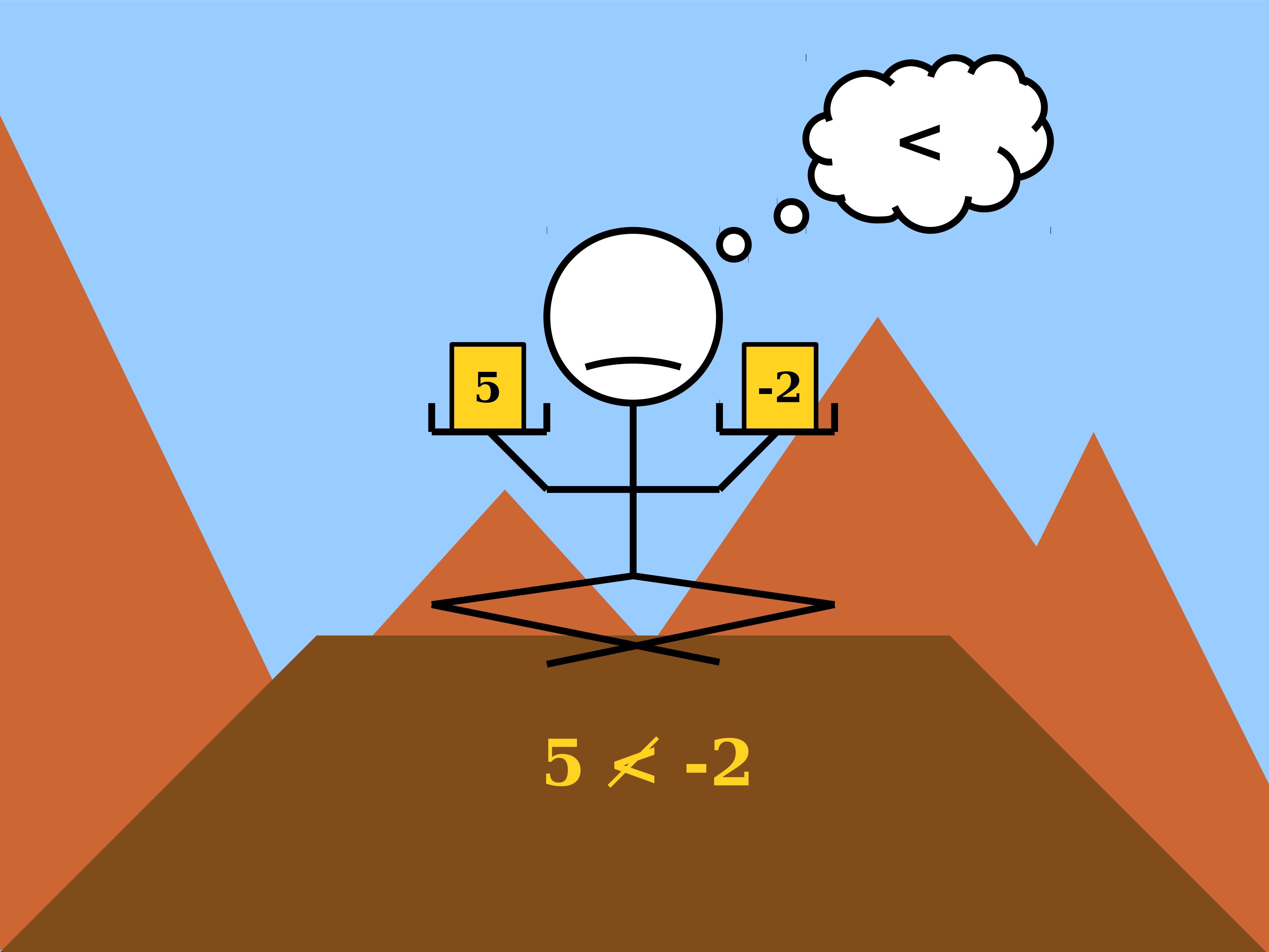


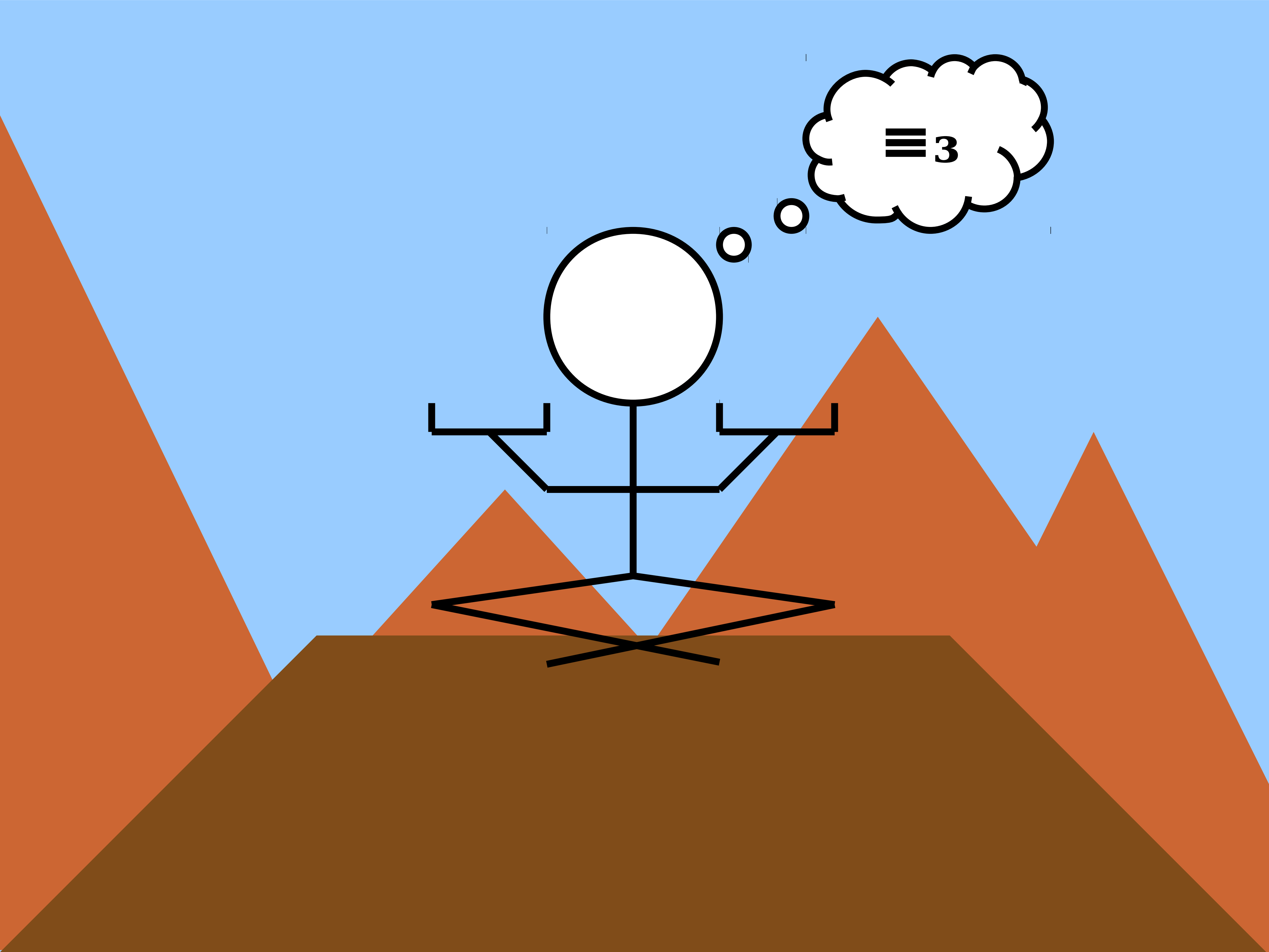


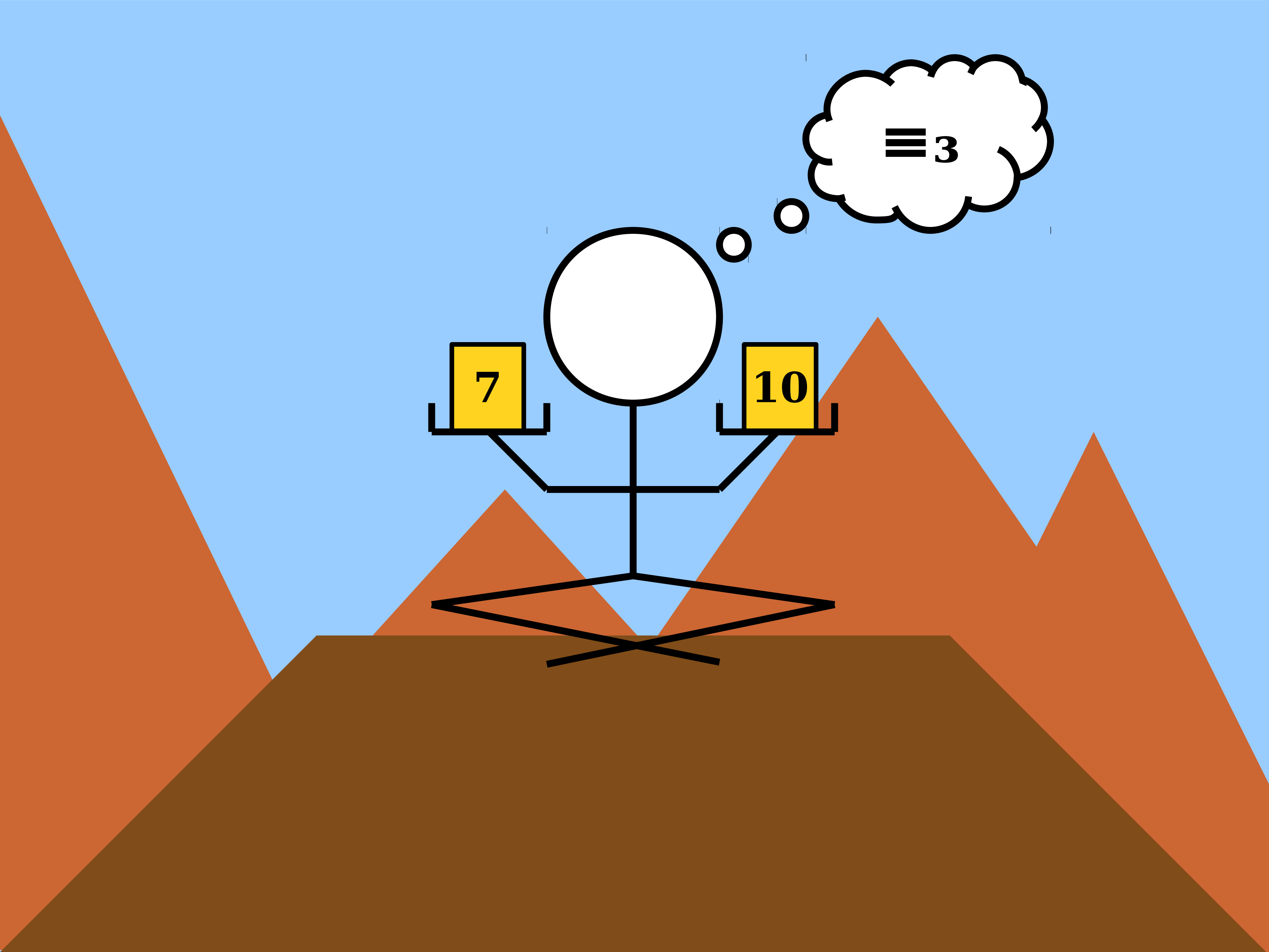

10
12
<

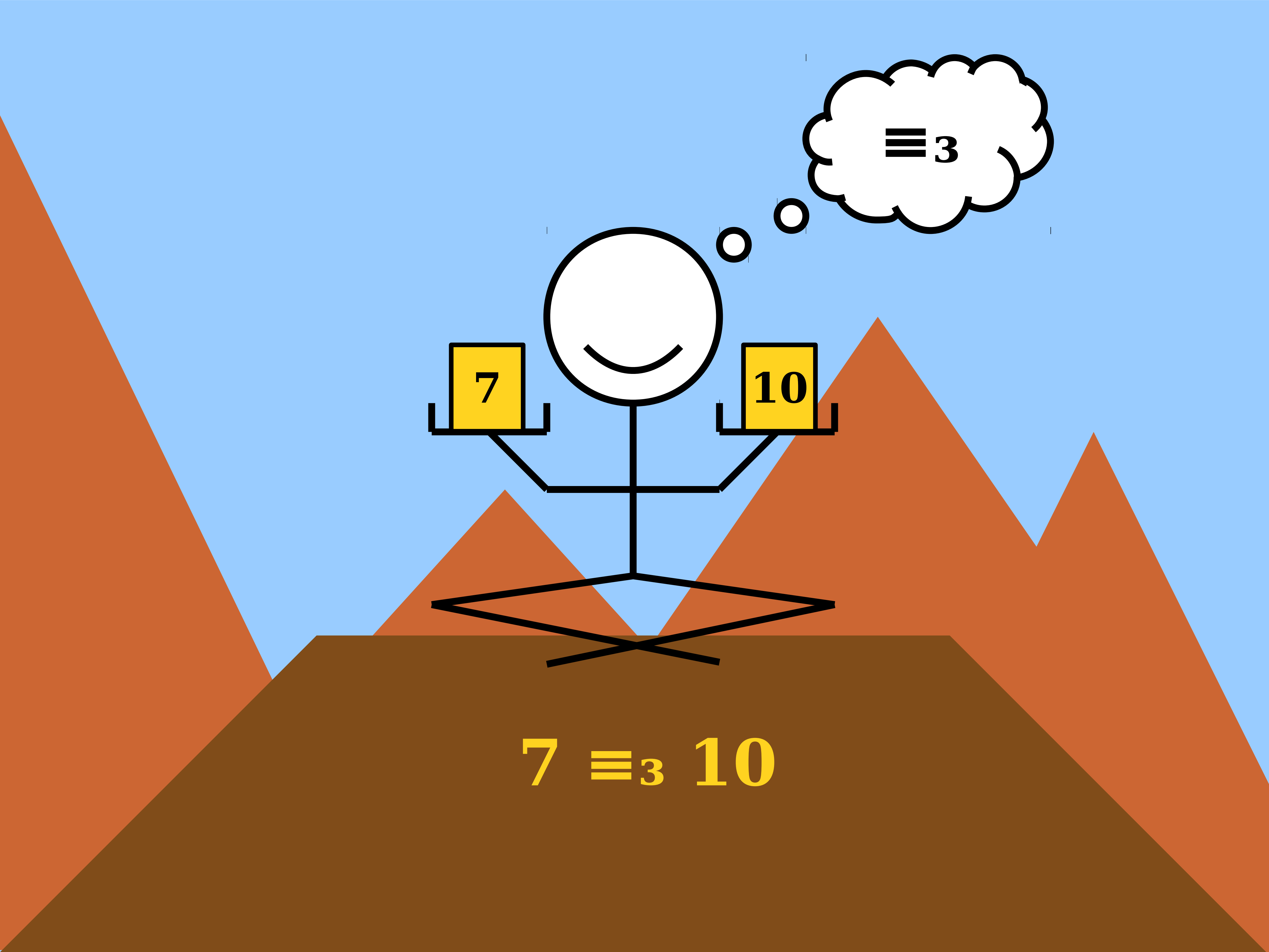

<
10
12
10 < 12

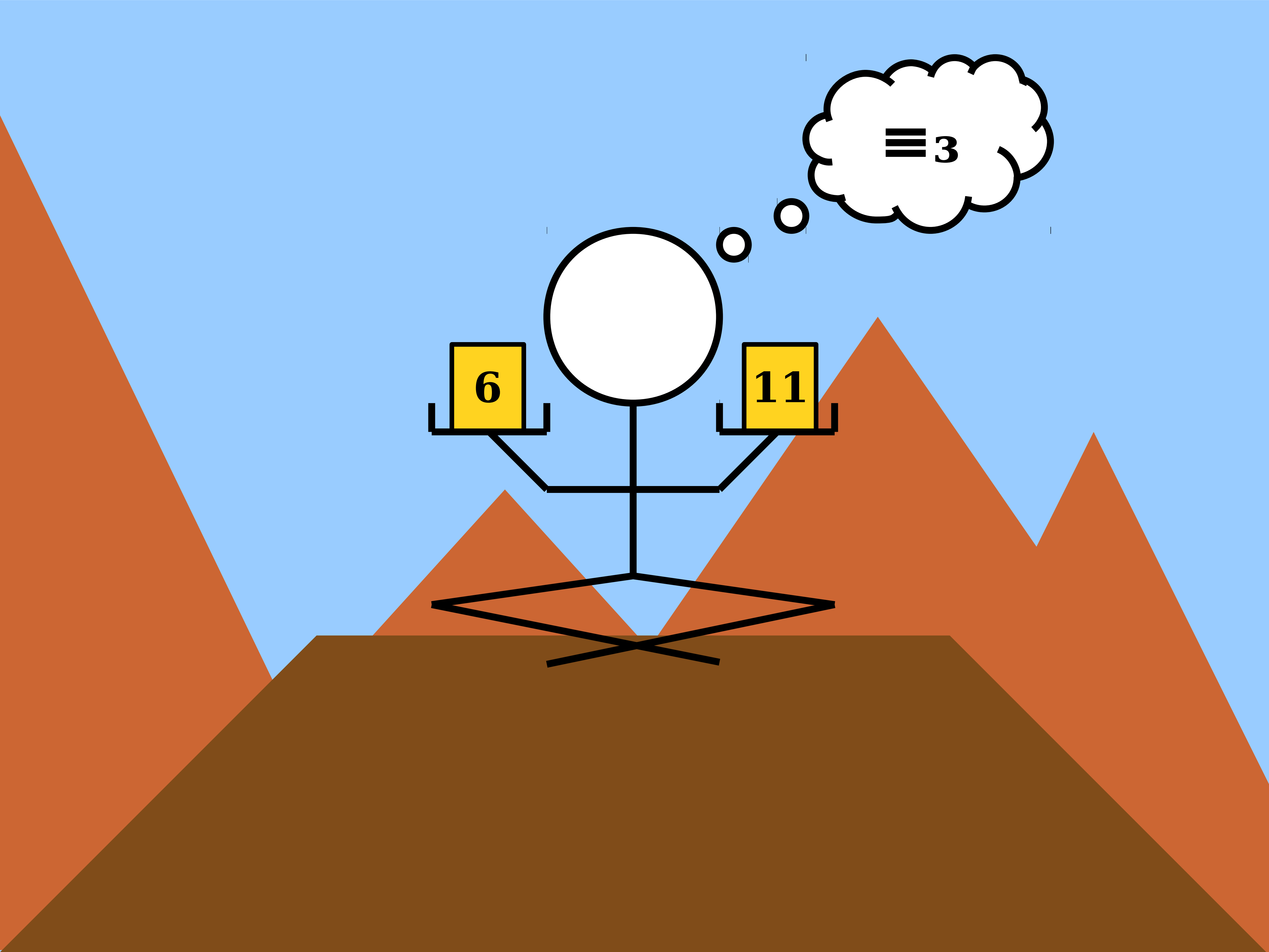

<
5
-2

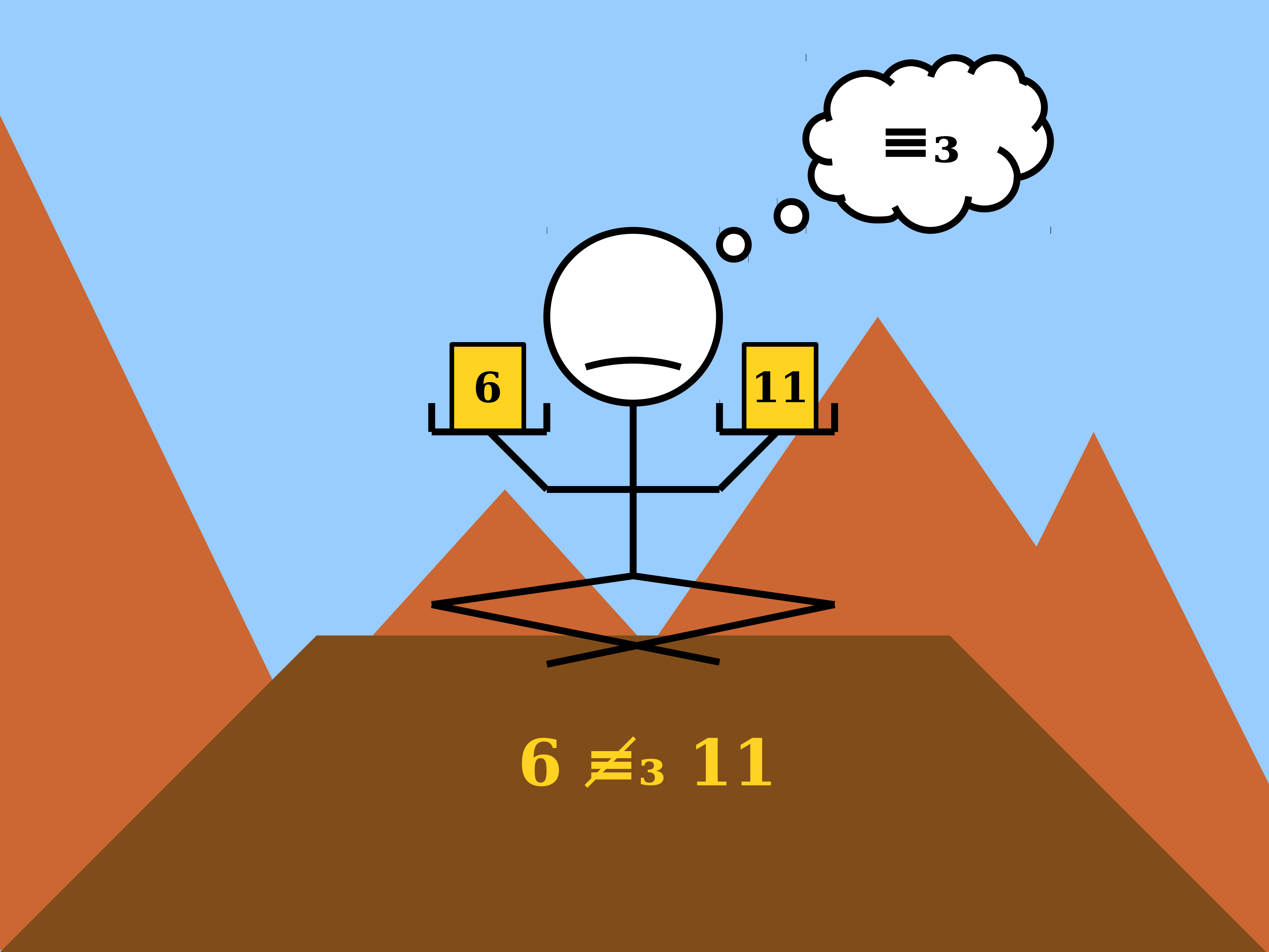

<
5
-2
5 ≮ -2


≡3

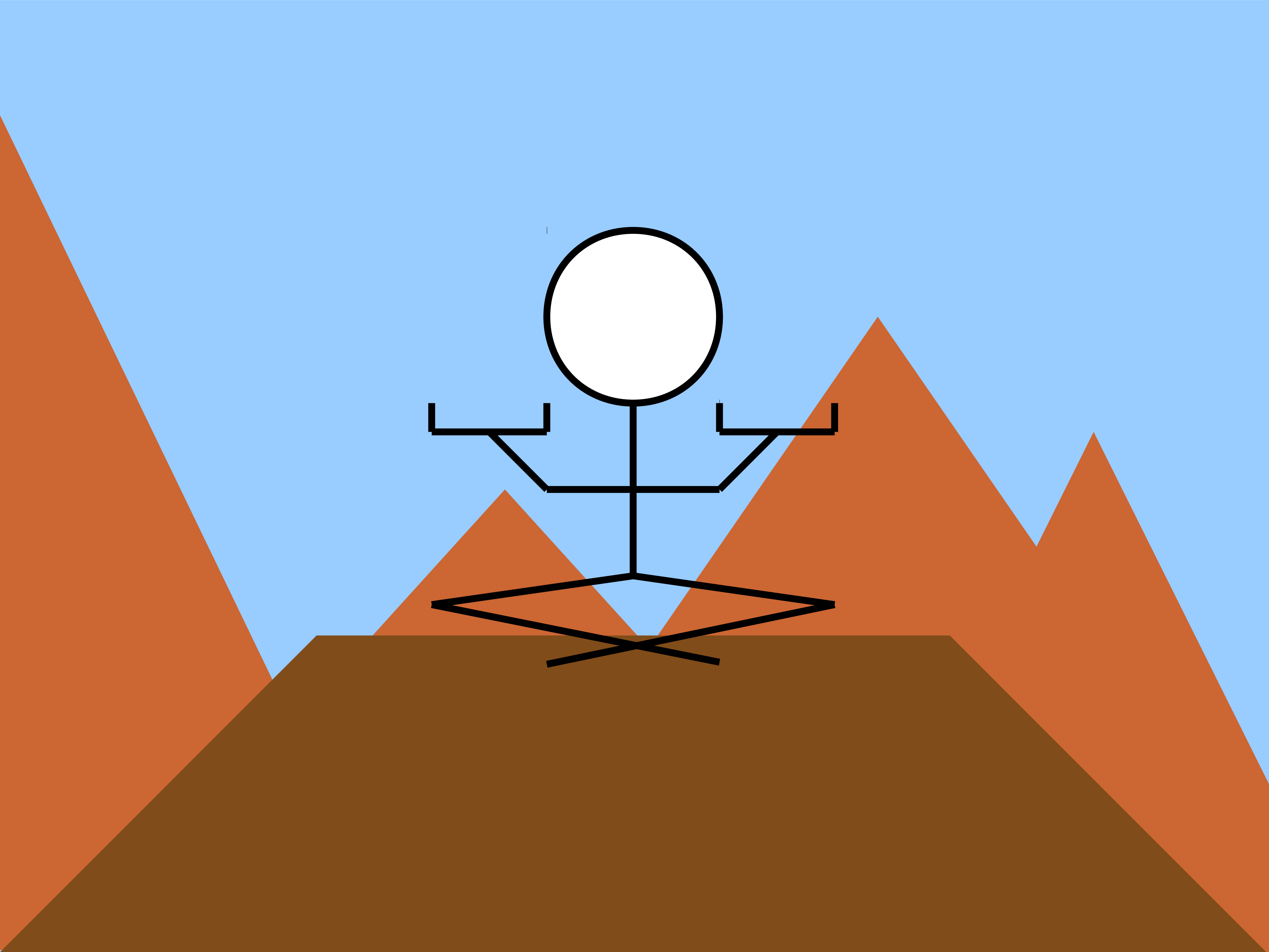

≡3
7
10

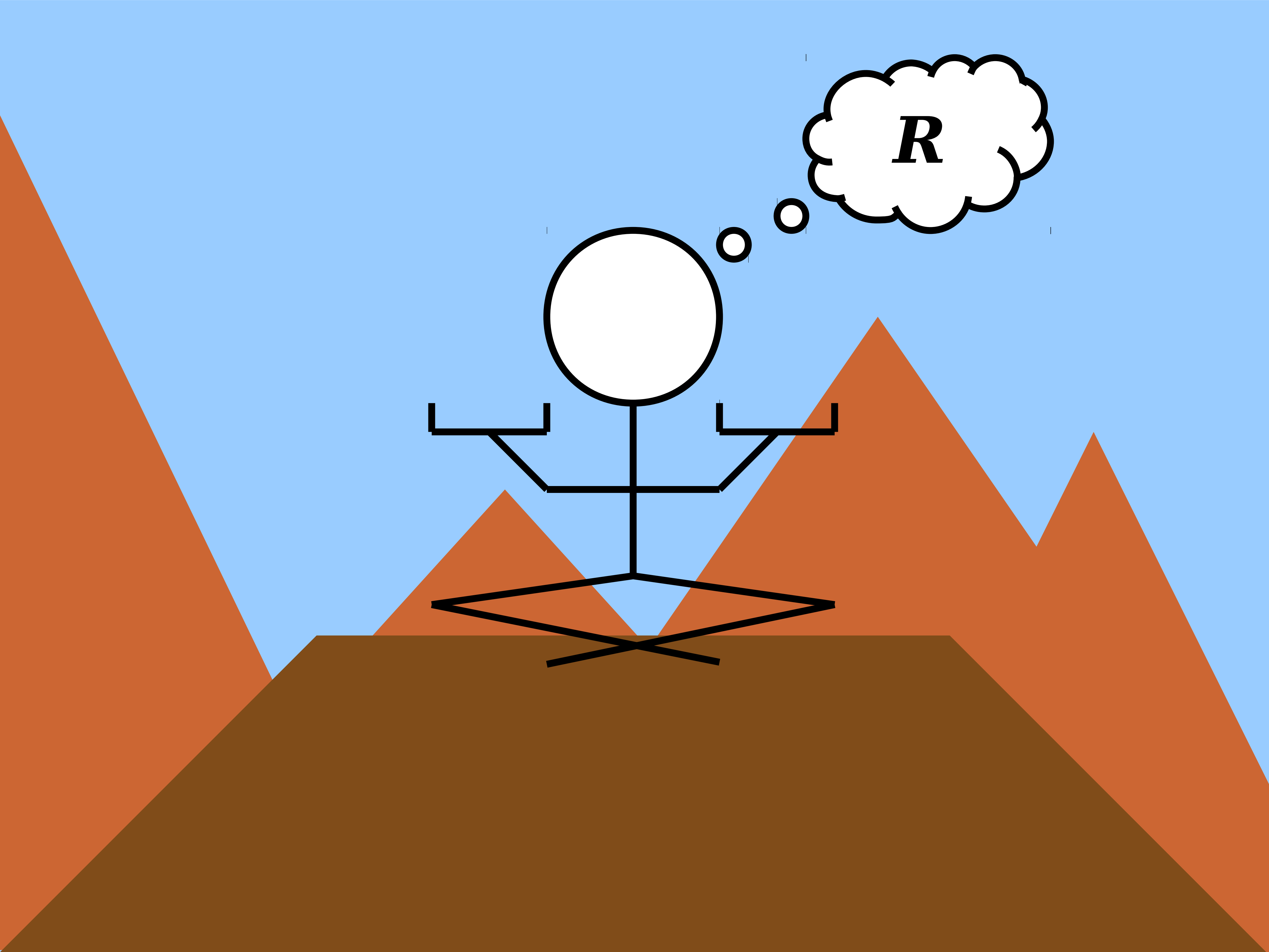

≡₃
7
10
7 ≡₃ 10

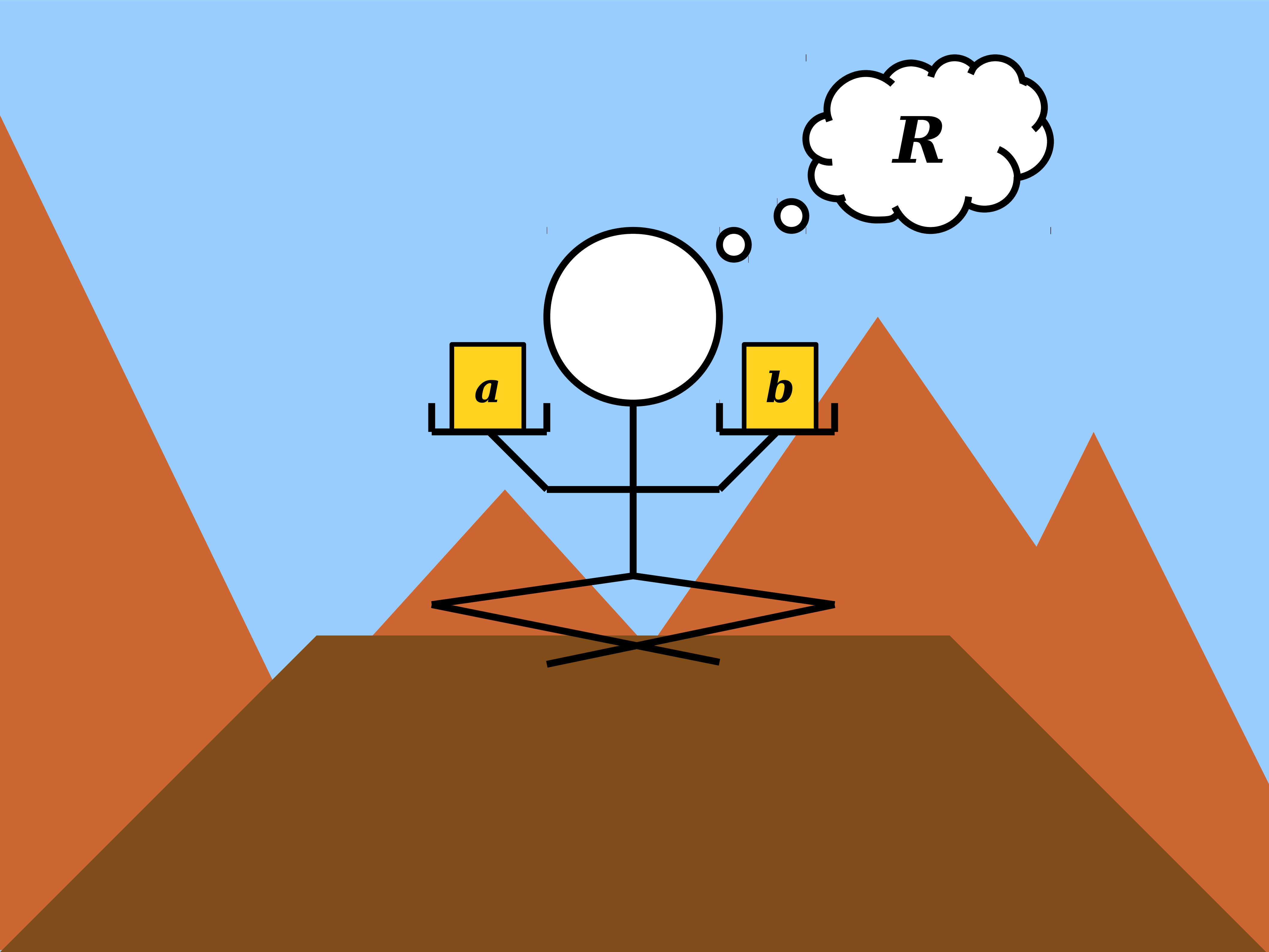

≡₃
6
11

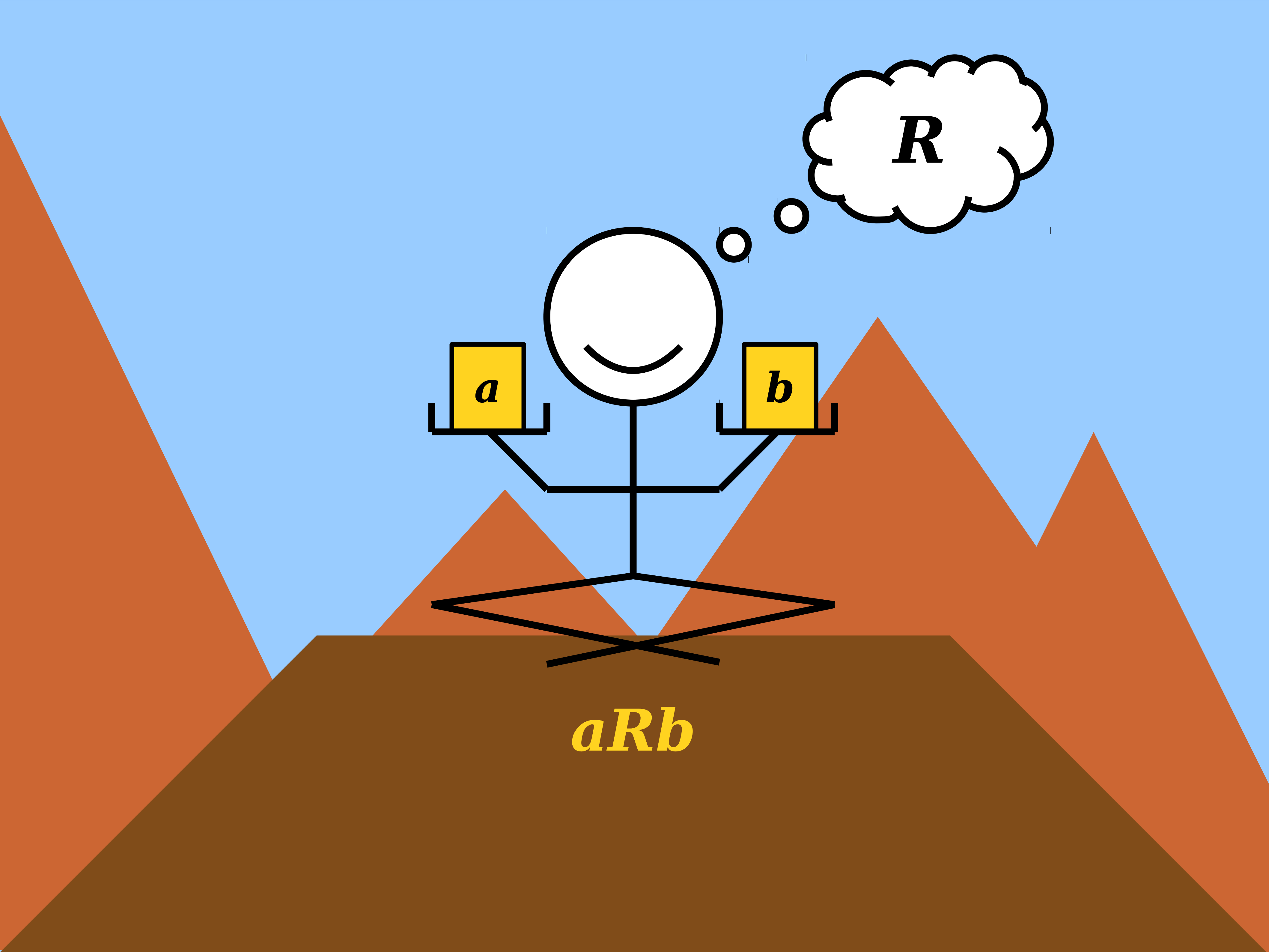


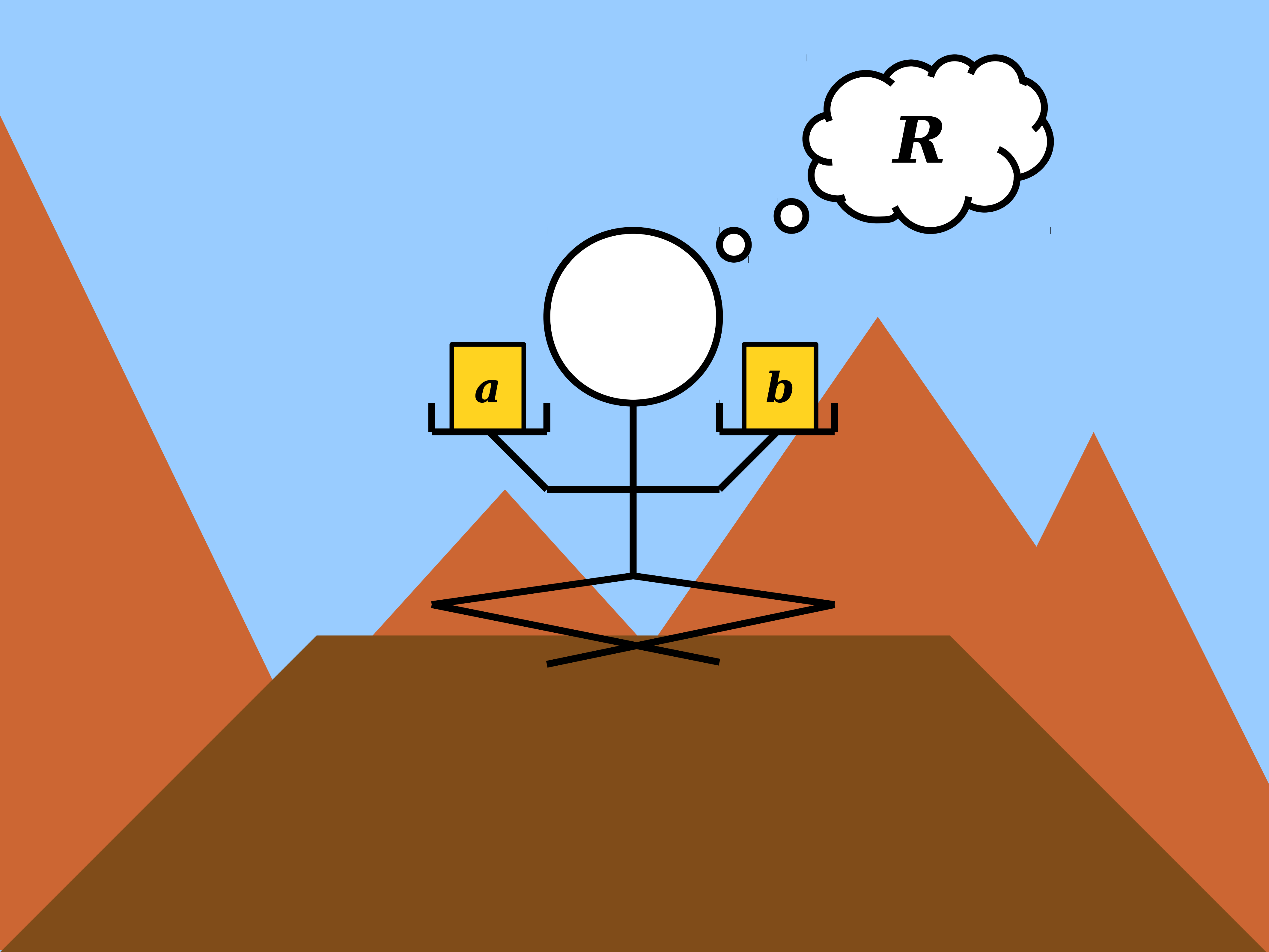
≡₃
6
11
6 ≢₃ 11

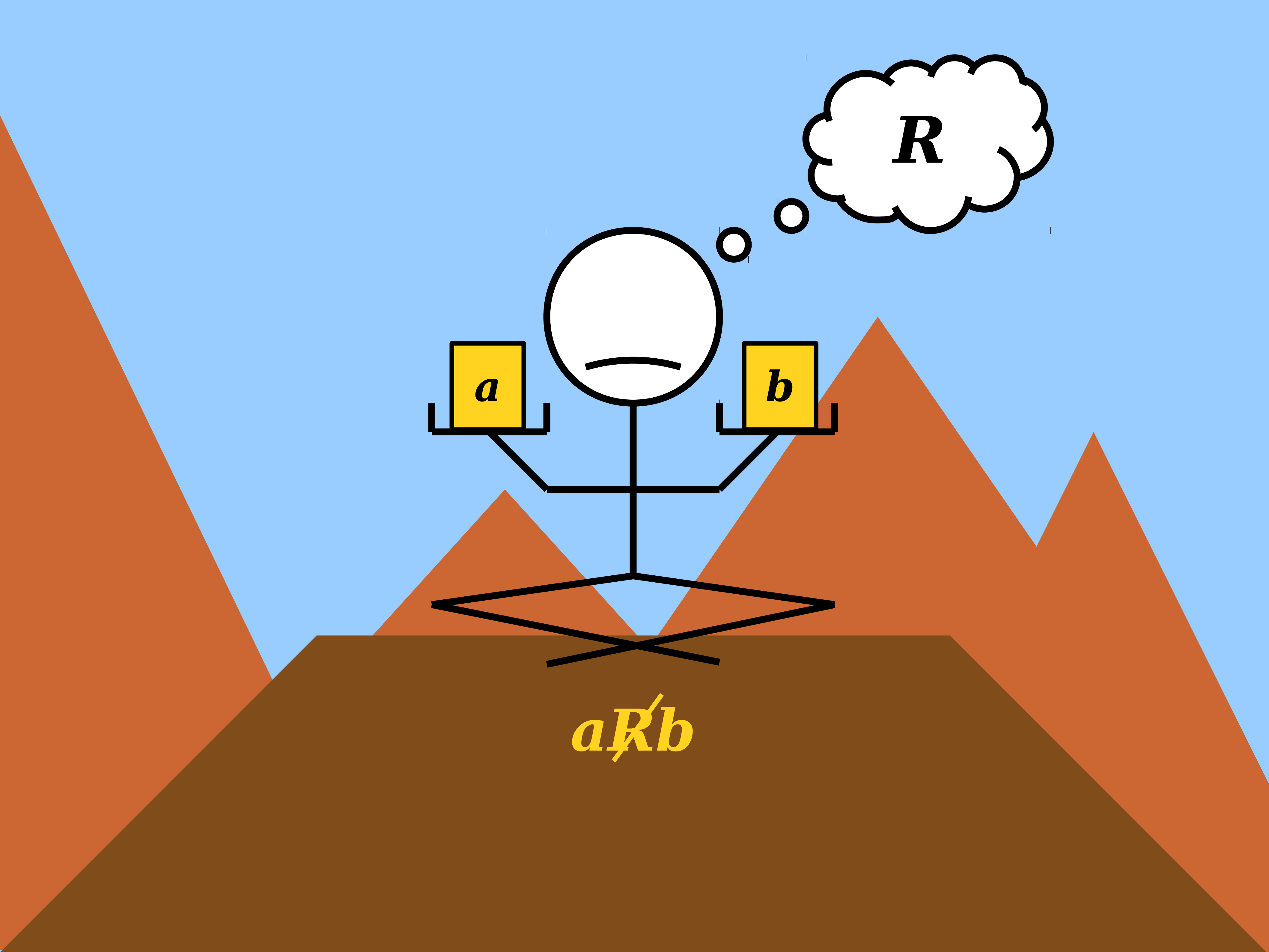



R


R
a
b


R
a
b
aRb


R
a
b


R
a
b
$a \not R b$

Binary Relations

- A ***binary relation over a set A*** is a predicate R that can be applied to pairs of elements drawn from A.
- If R is a binary relation over A and it holds for the pair (a, b), we write $\boldsymbol{aRb}$.

$$3 = 3 \qquad 5 < 7 \qquad \emptyset \subseteq \mathbb{N}$$

- If R is a binary relation over A and it does not hold for the pair (a, b), we write $\boldsymbol{a \not{R} b}$.

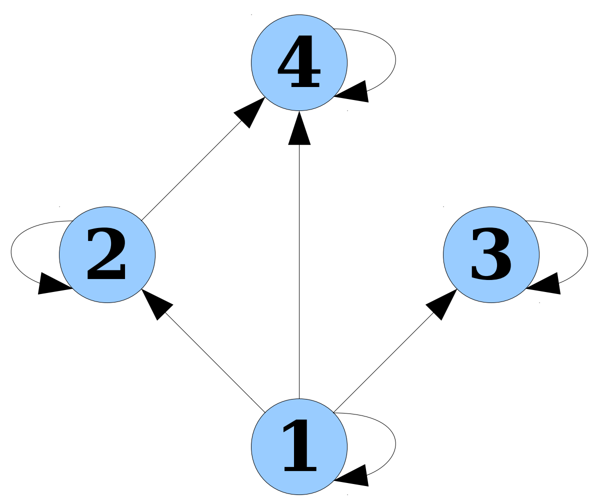
$$4 \neq 3 \qquad 4 \not< 3 \qquad \mathbb{N} \not\subseteq \emptyset$$

Properties of Relations

- Generally speaking, if R is a binary relation over a set A, the order of the operands is significant.
  - For example, $3 < 5$, but $5 \not< 3$.
  - In some relations order is irrelevant; more on that later.
- Relations are always defined relative to some underlying set.
  - It's not meaningful to ask whether ☺ $\subseteq$ 15, for example, since $\subseteq$ is defined over sets, not arbitrary objects.

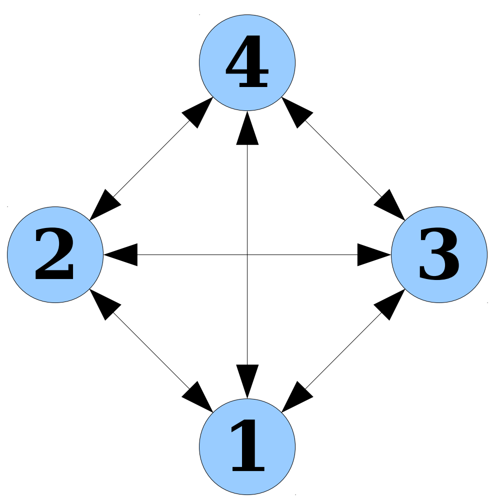
Visualizing Relations

- 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:



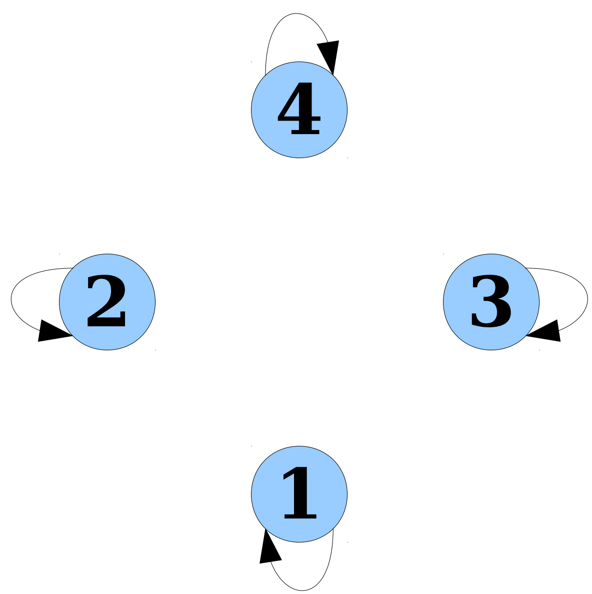
Visualizing Relations

- 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:



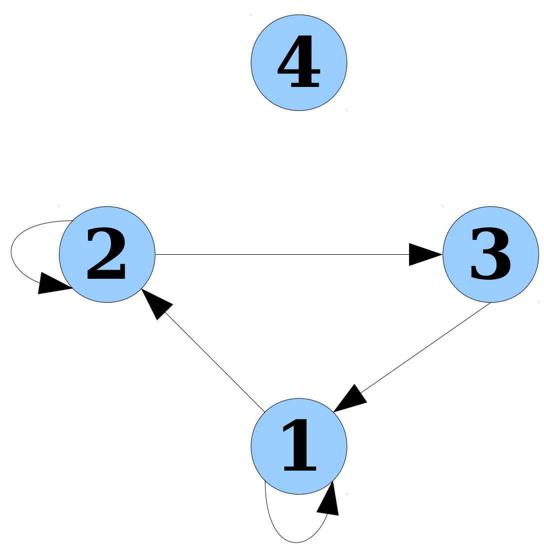
Visualizing Relations

- 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:



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: below is some relation over {1, 2, 3, 4} that's a totally valid relation even though there doesn't appear to be a simple unifying rule.

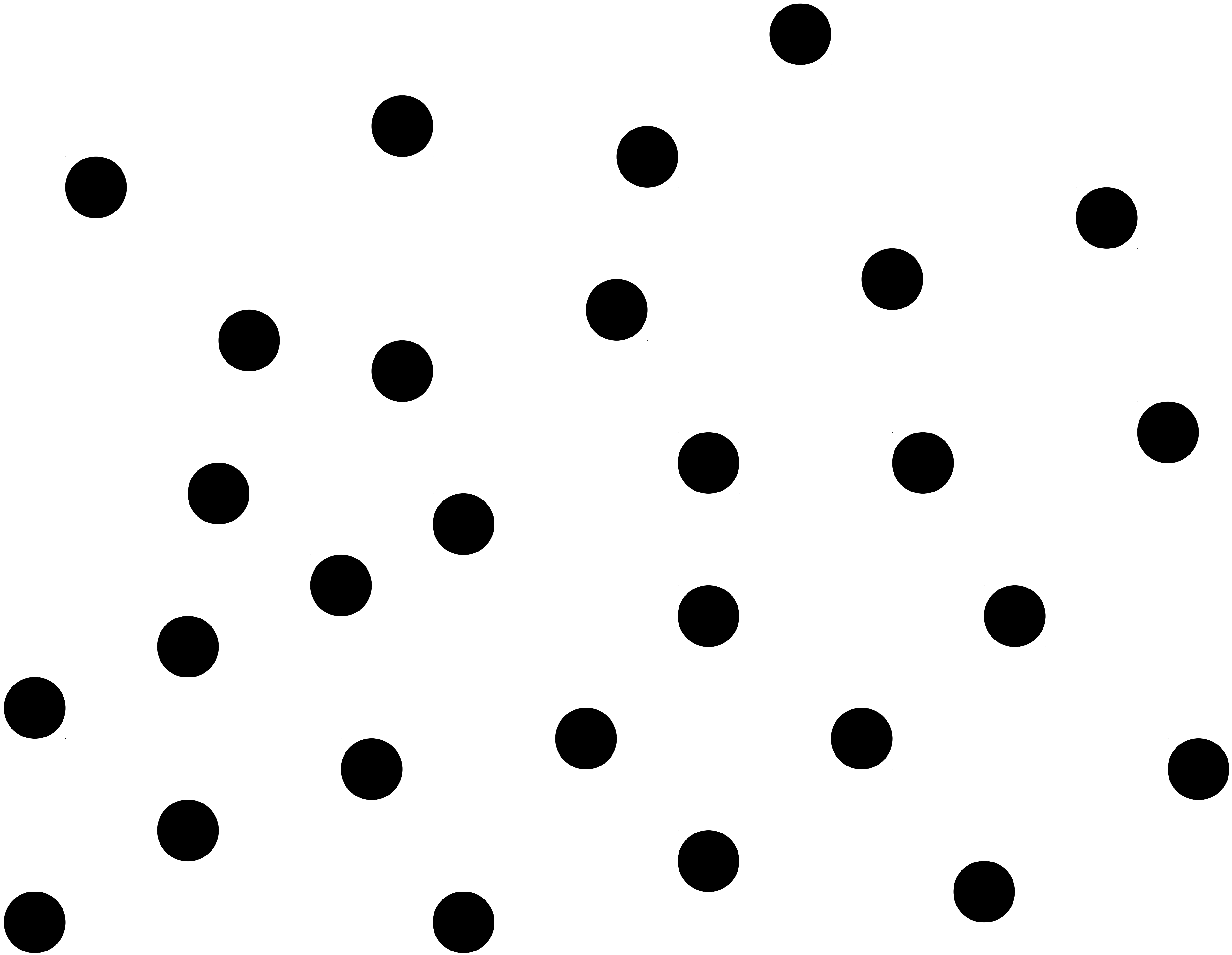


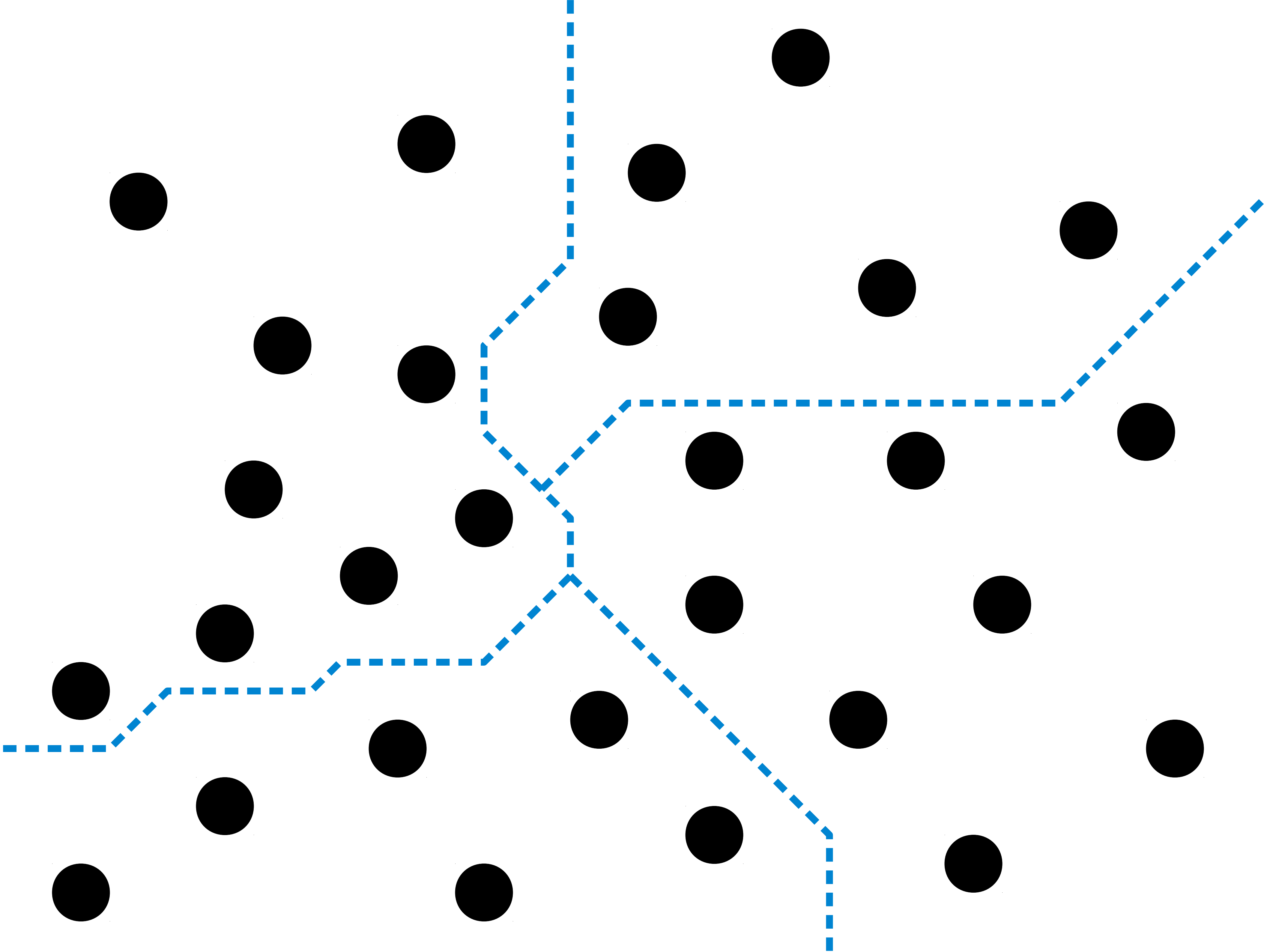
Capturing Structure

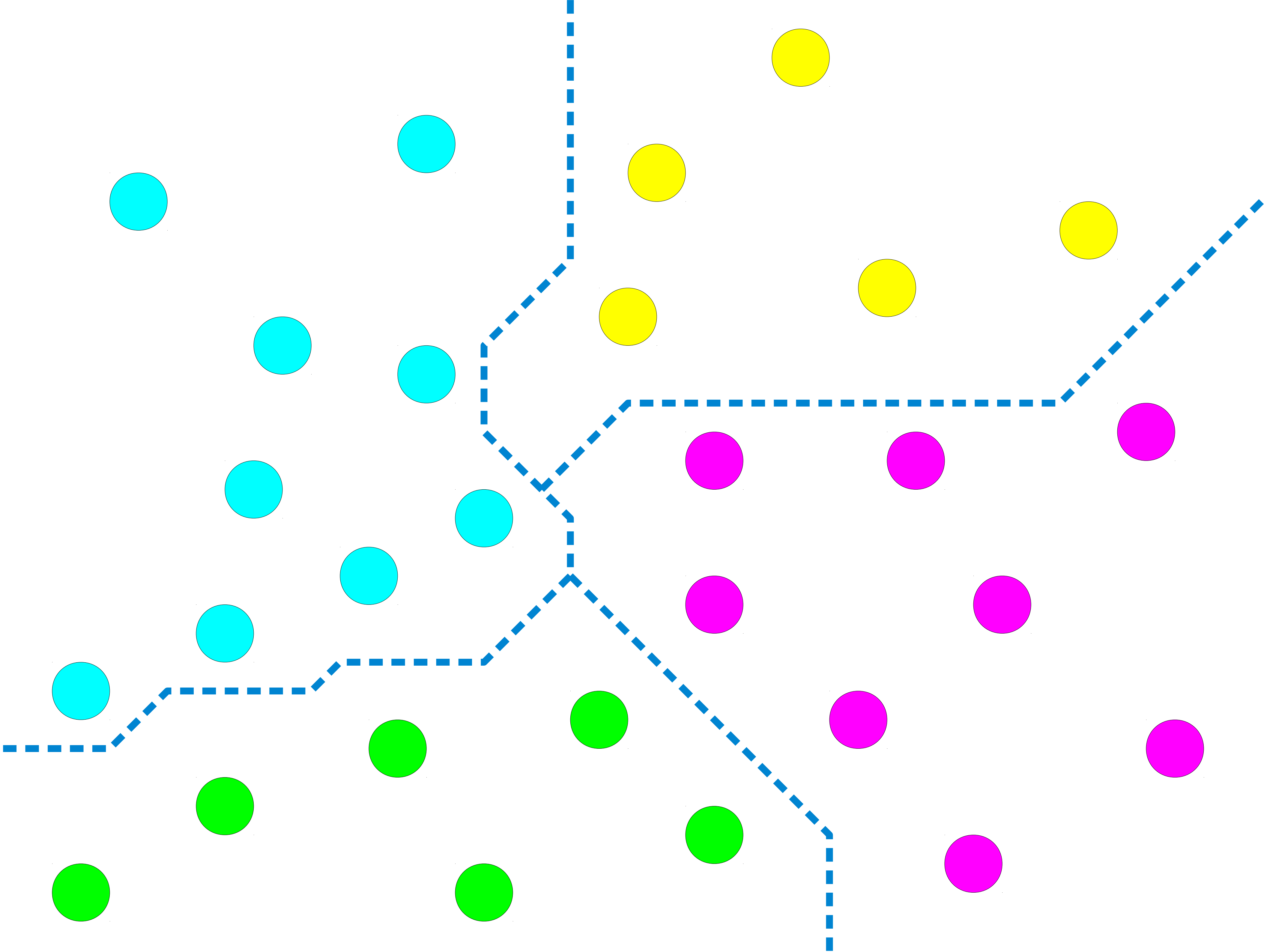
Capturing Structure

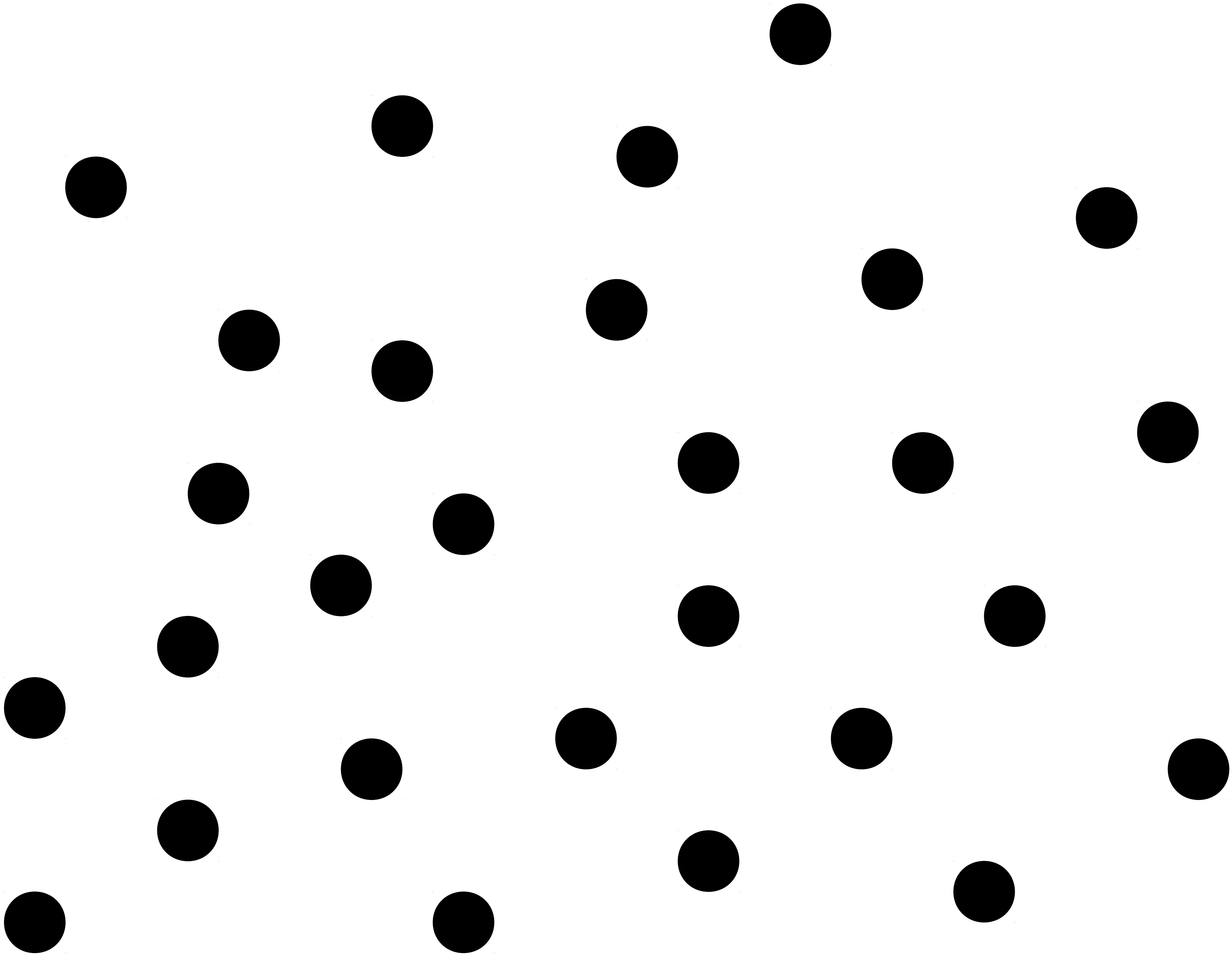
- Binary relations are an excellent way for capturing certain structures that appear in computer science.
- Today, we'll look at one of them (***partitions***), and next time we'll see another (***prerequisites***).
- Along the way, we'll explore how to write proofs about definitions given in first-order logic.

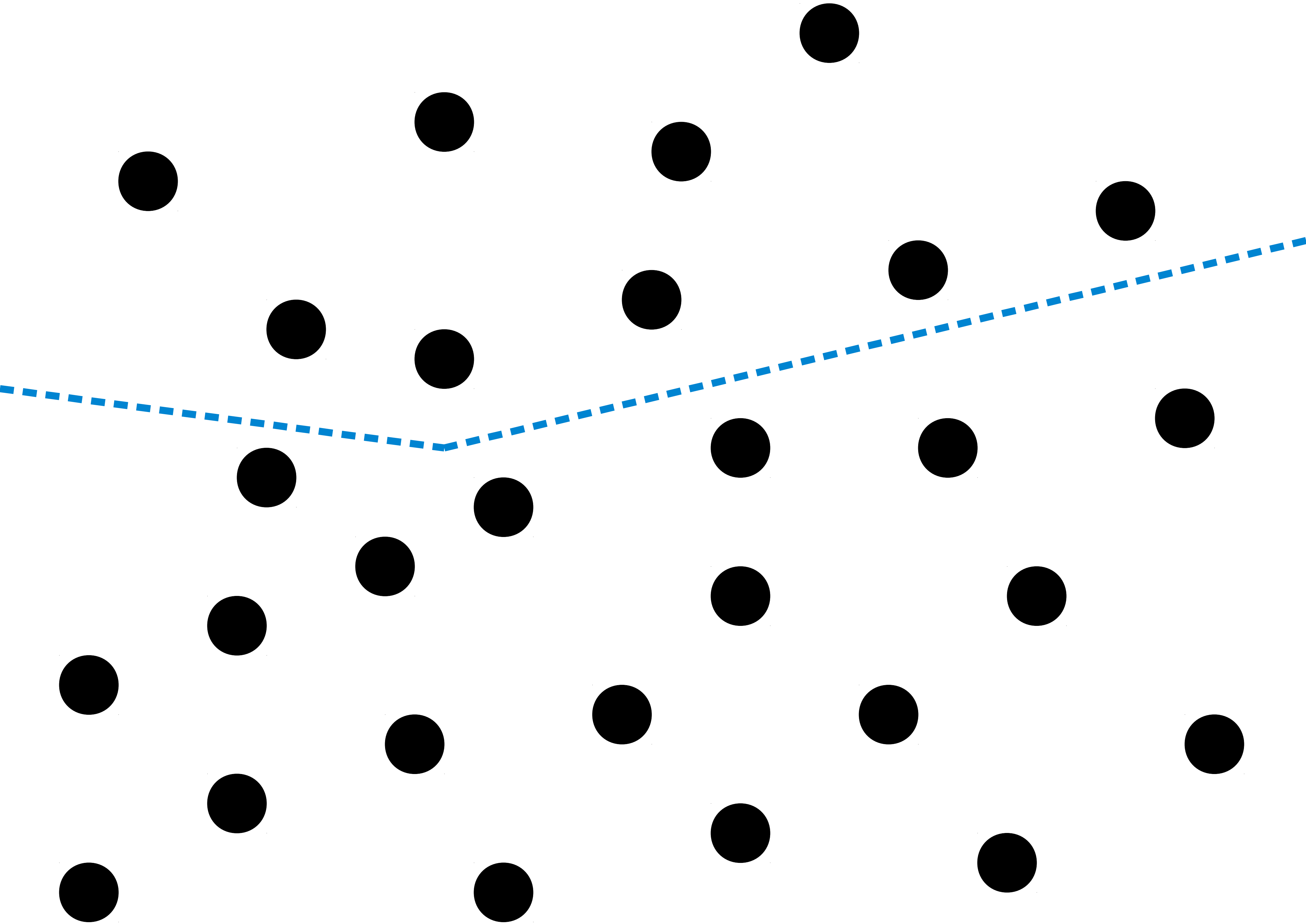
Partitions

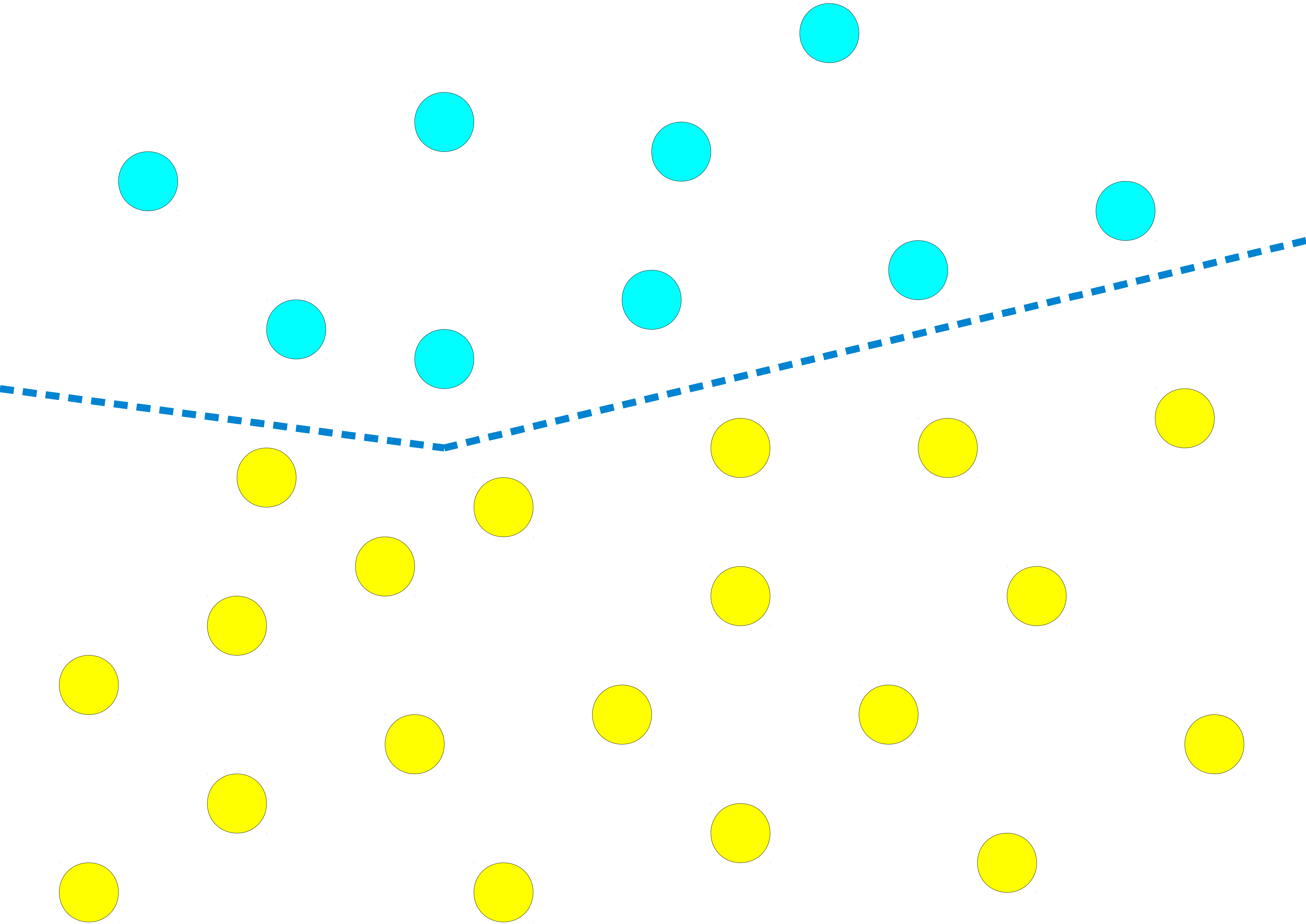


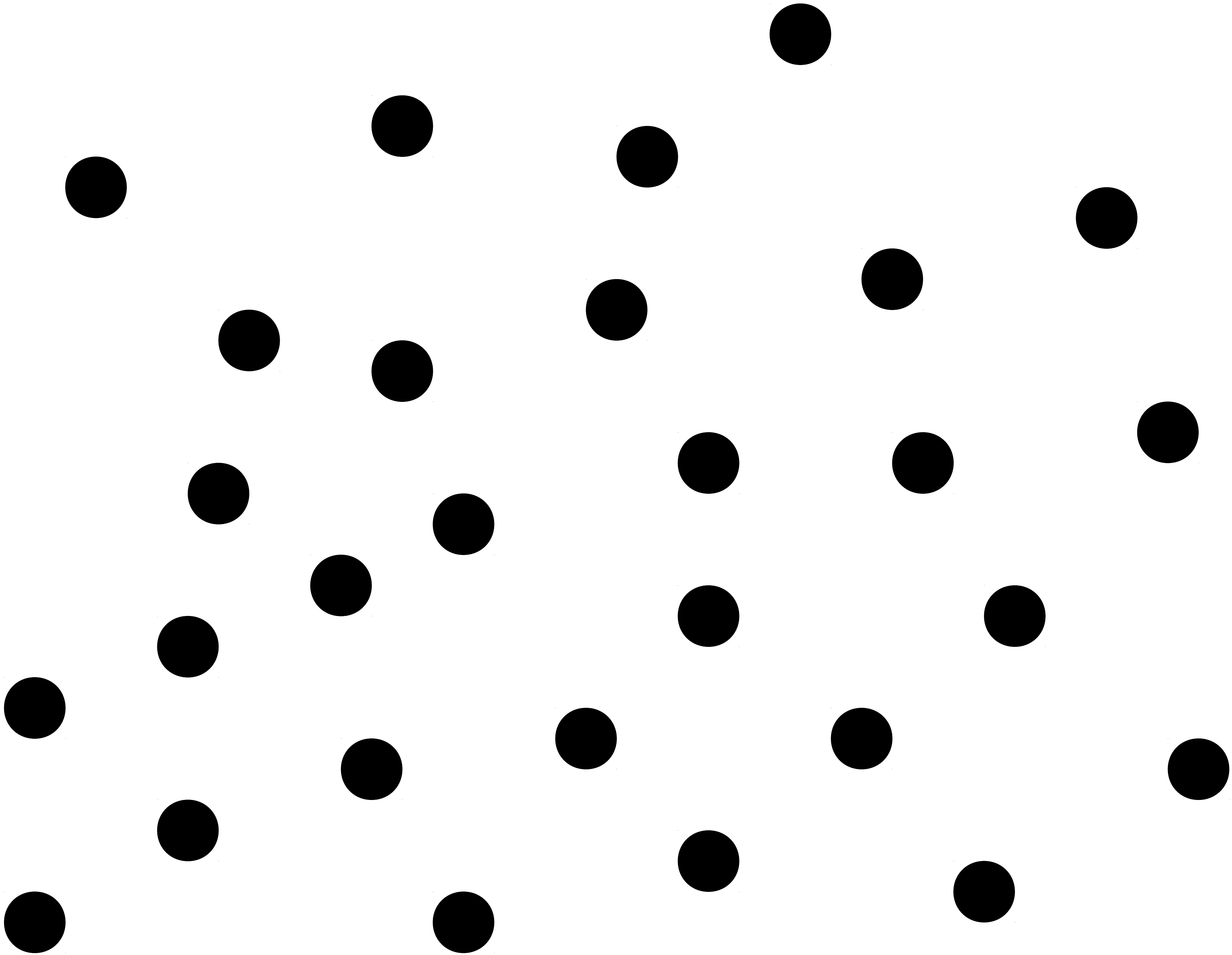


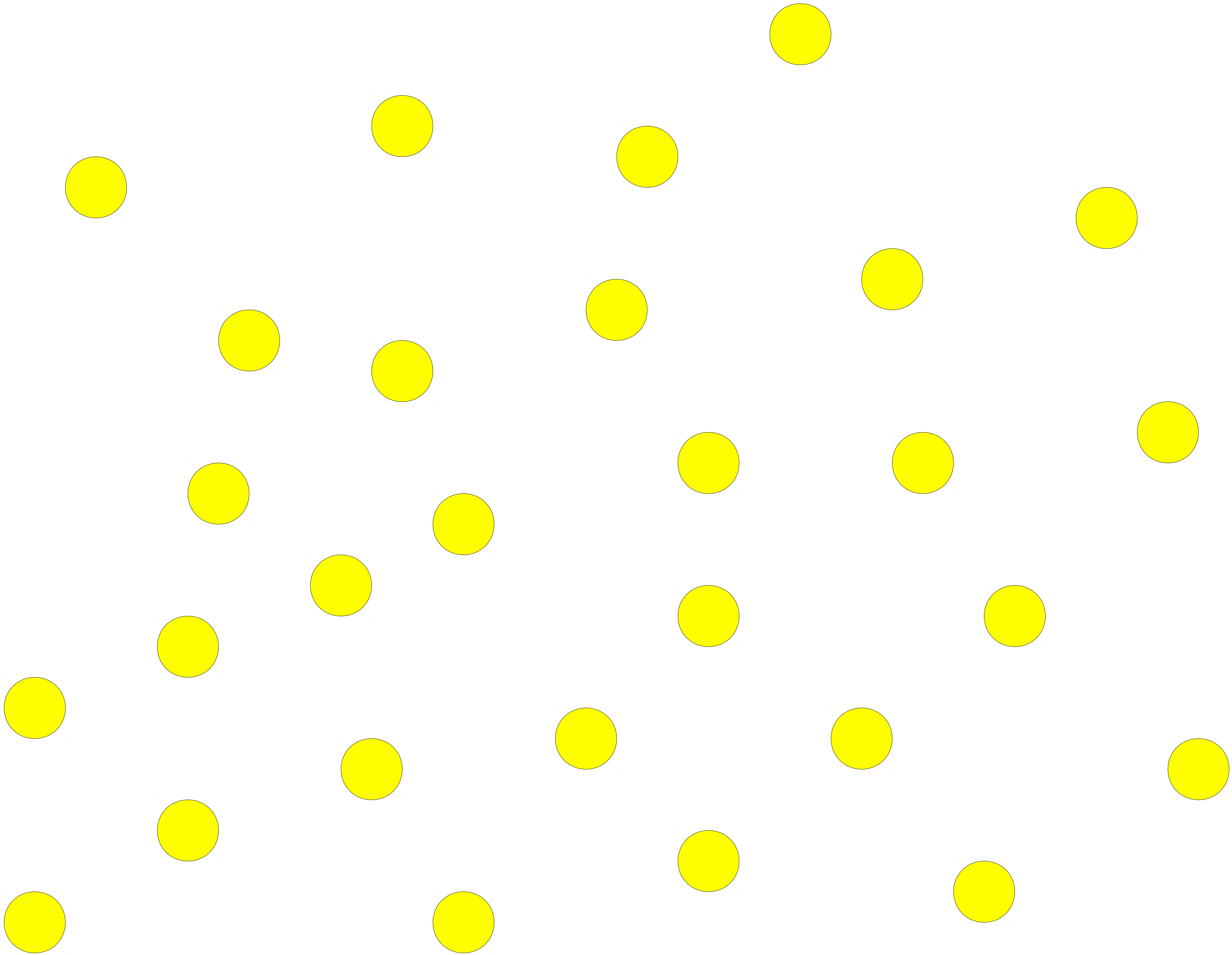


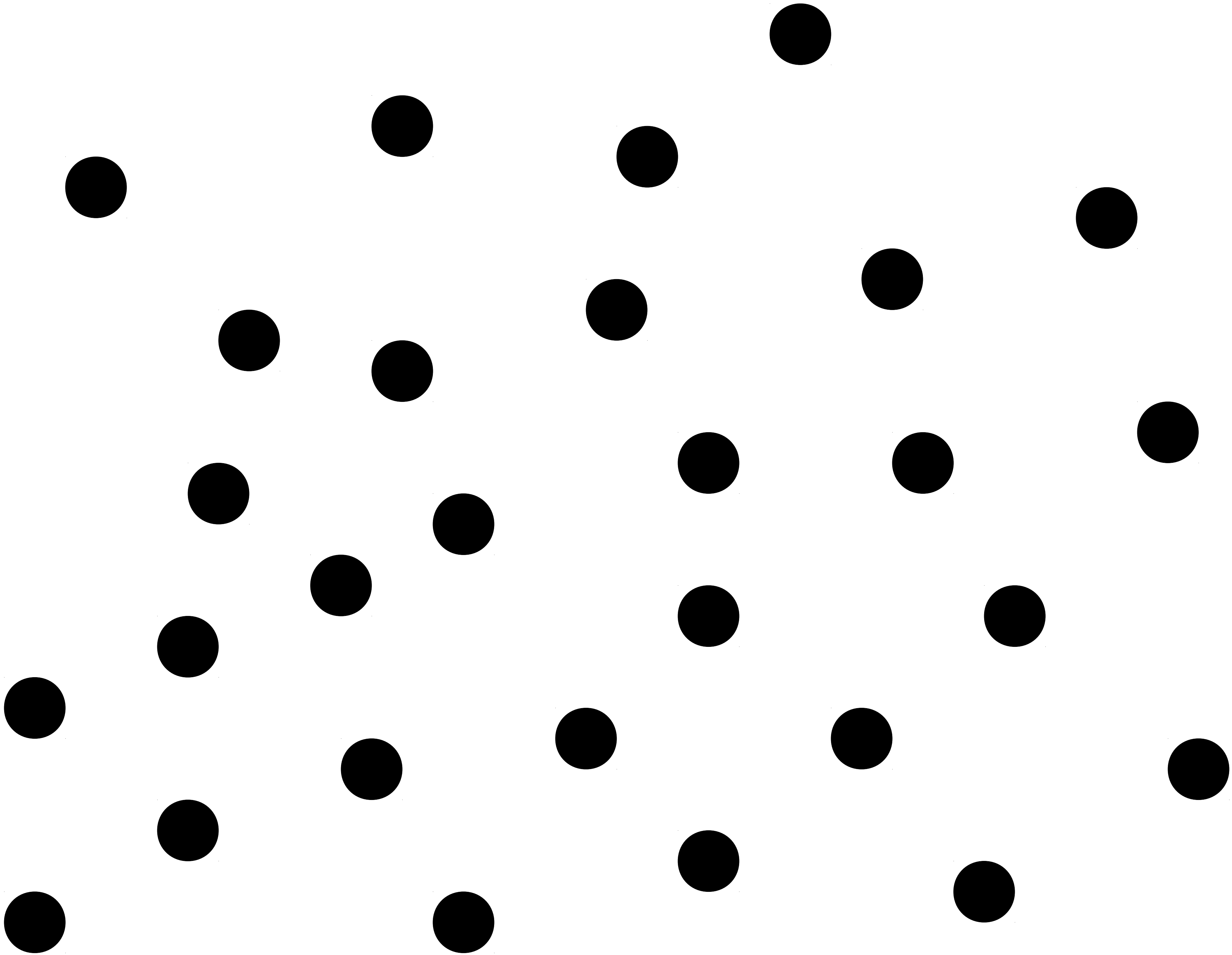


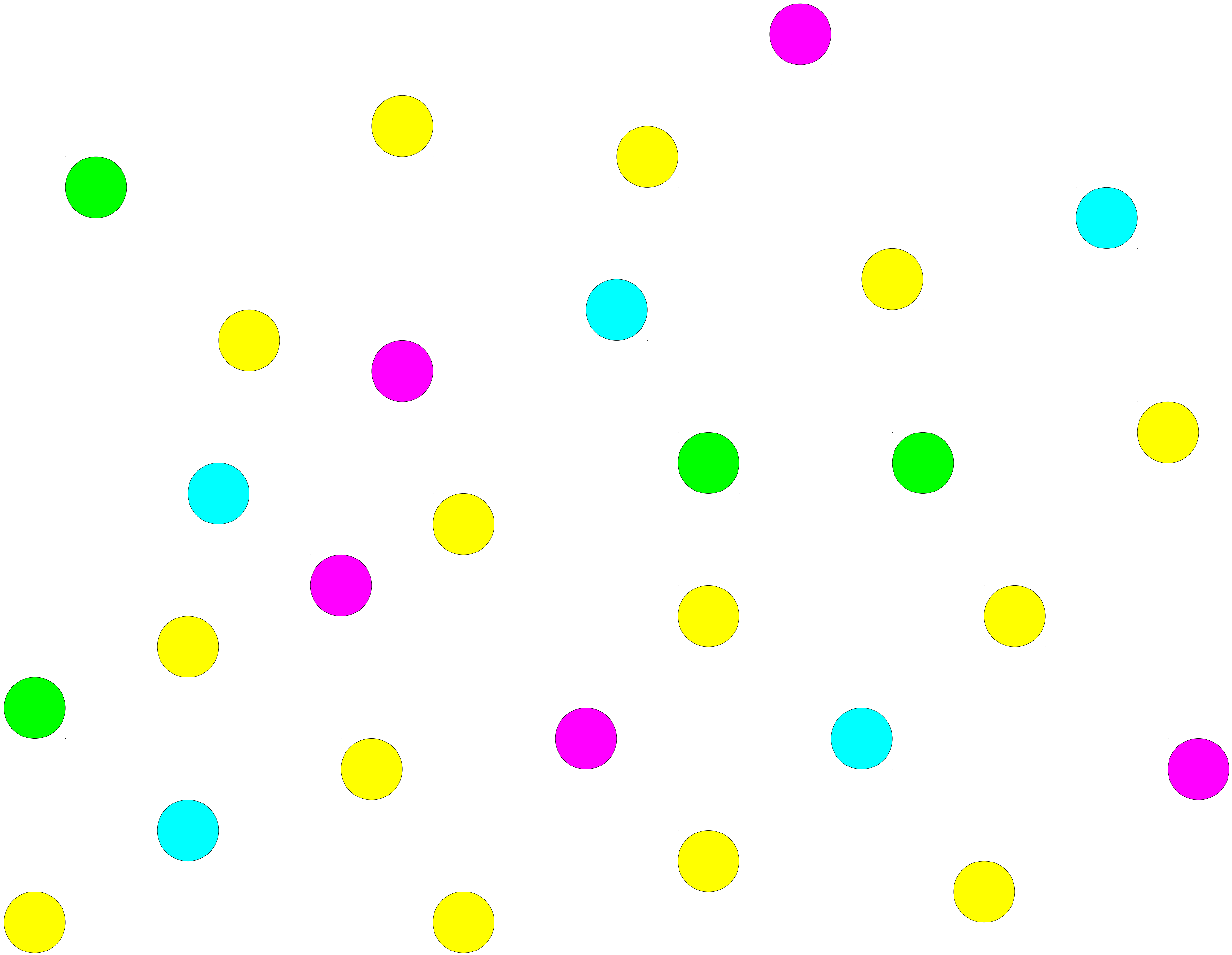












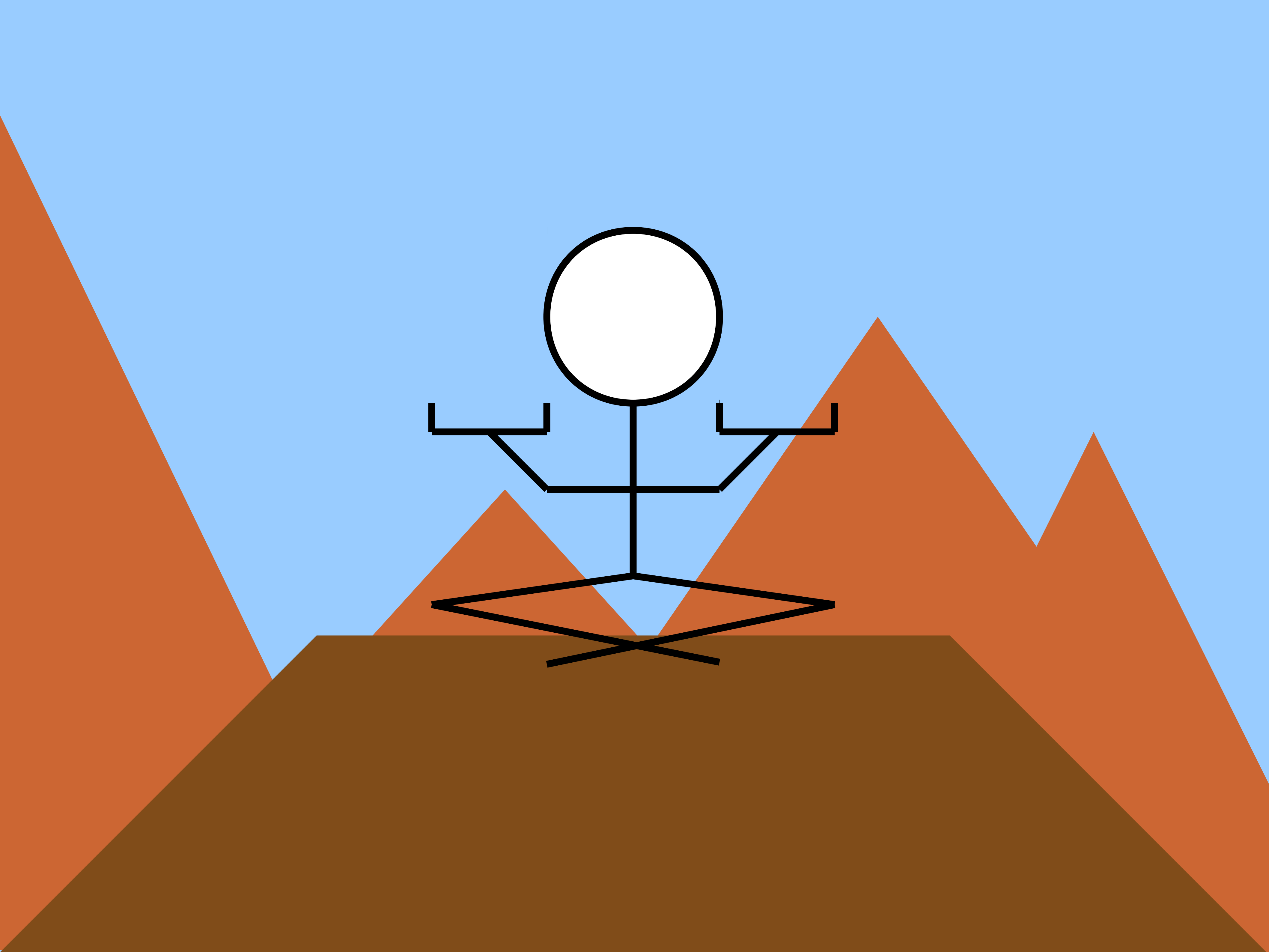
Partitions

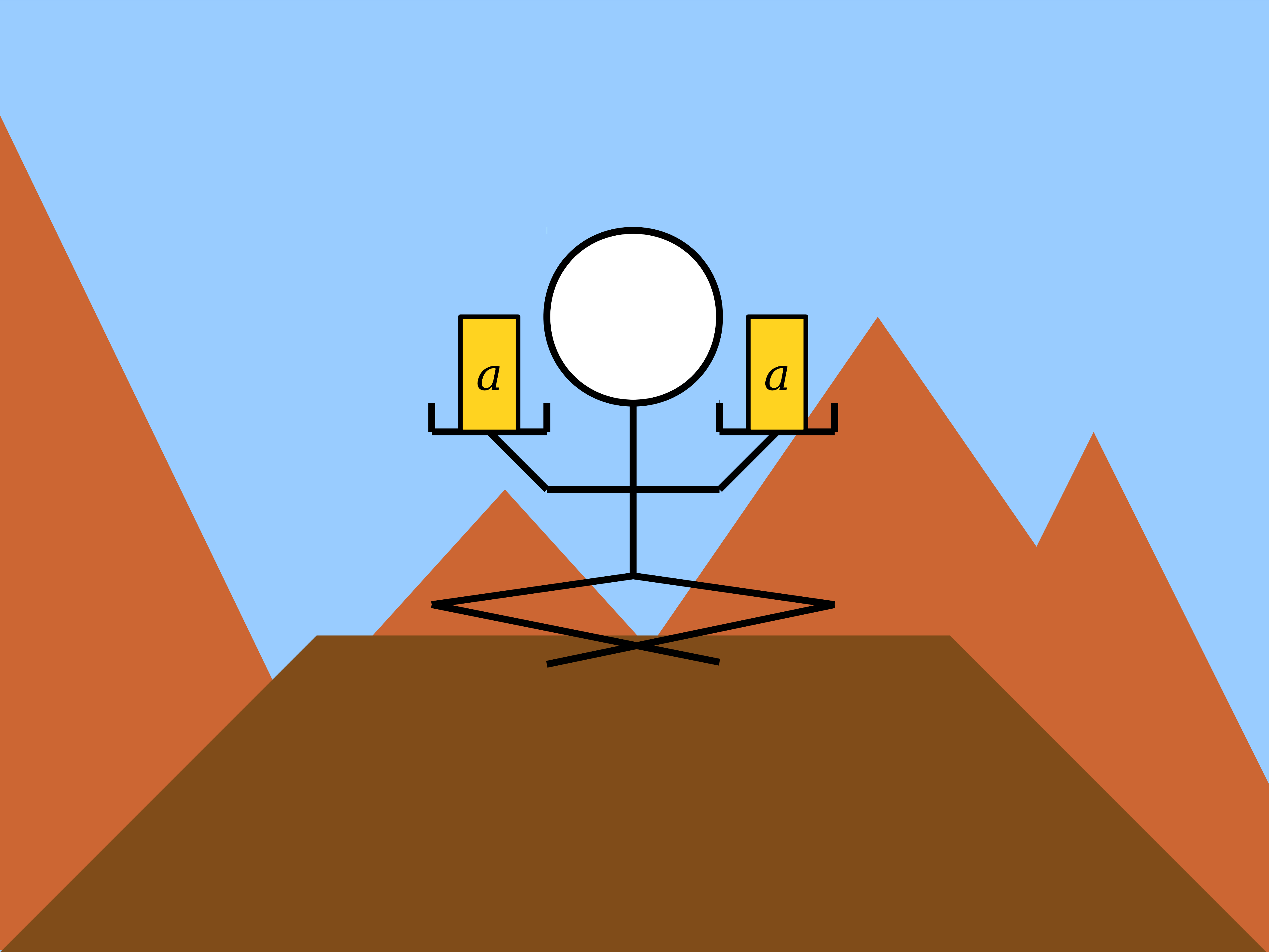
- A ***partition of a set*** is a way of splitting the set into disjoint, nonempty subsets so that every element belongs to exactly one subset.
  - Two sets are ***disjoint*** if their intersection is the empty set; formally, sets S and T are disjoint if $S \cap T = \emptyset$.
- Intuitively, a partition 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.

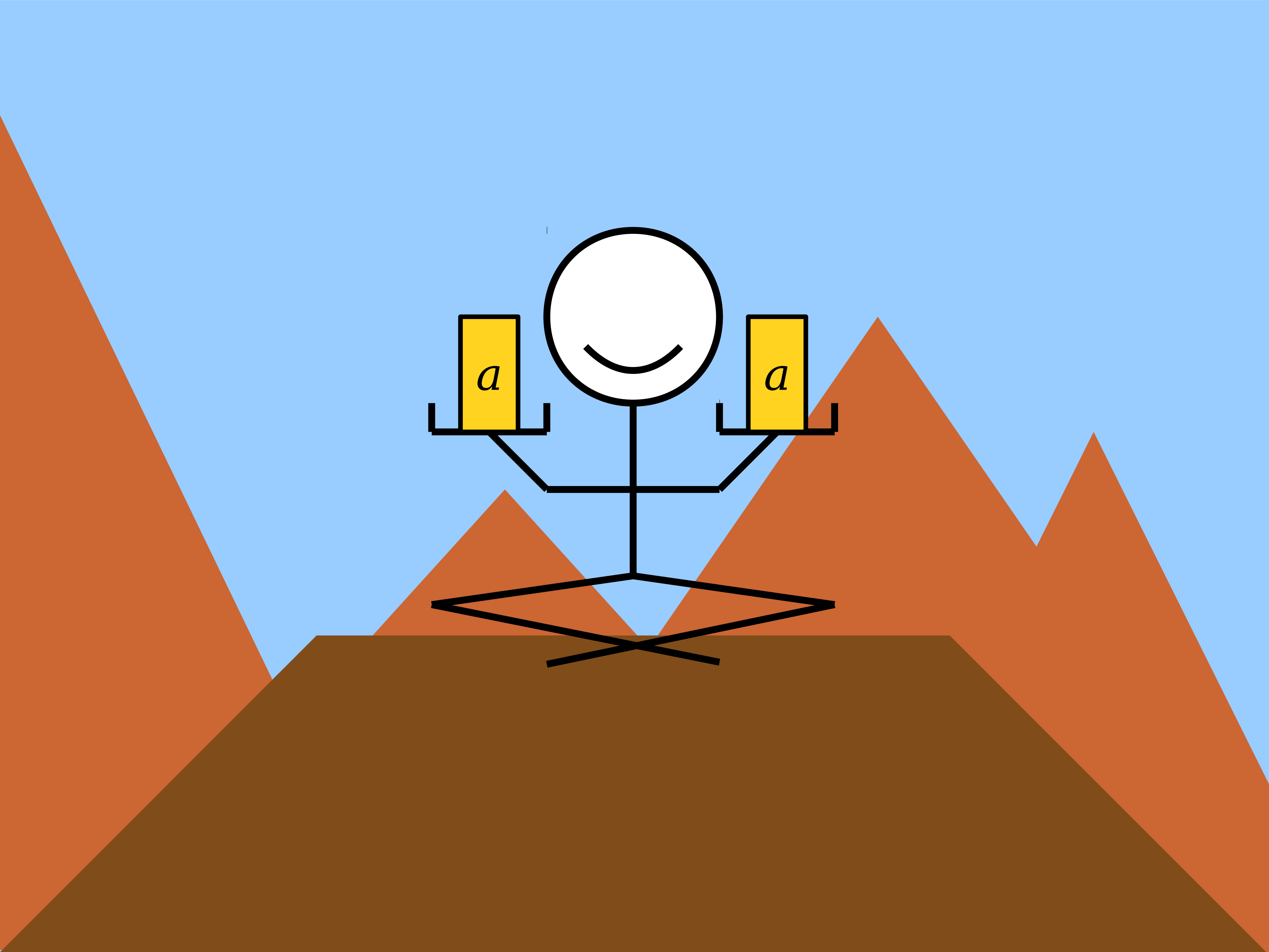
Partitions and Clustering

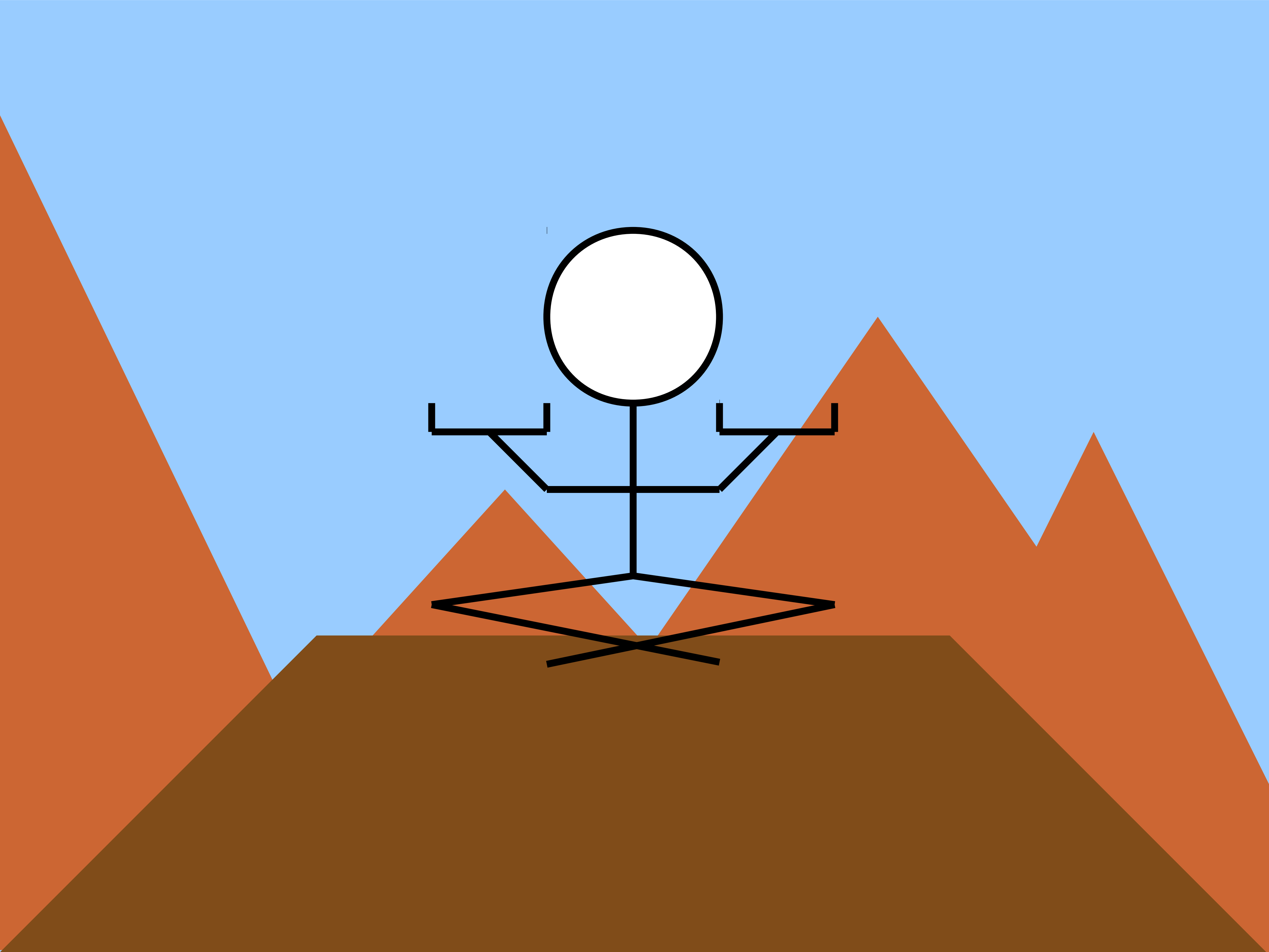
- If you have a set of data, you can often learn something from the data by finding a “good” partition of that data and inspecting the partitions.
  - Usually, the term ***clustering*** is used in data analysis rather than *partitioning*.
- Interested to learn more? Take CS161 or CS246!

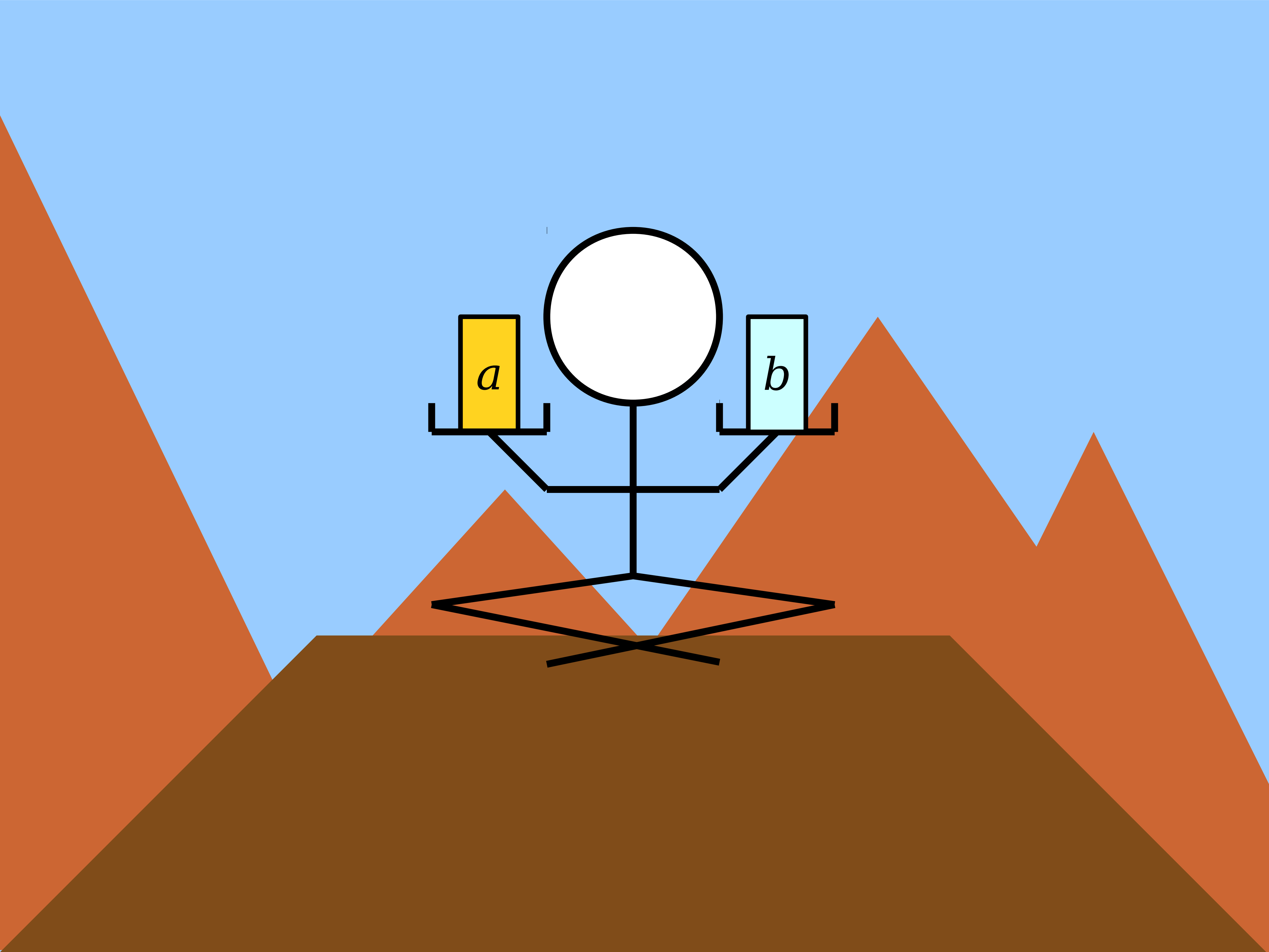
What's the connection between partitions and binary relations?

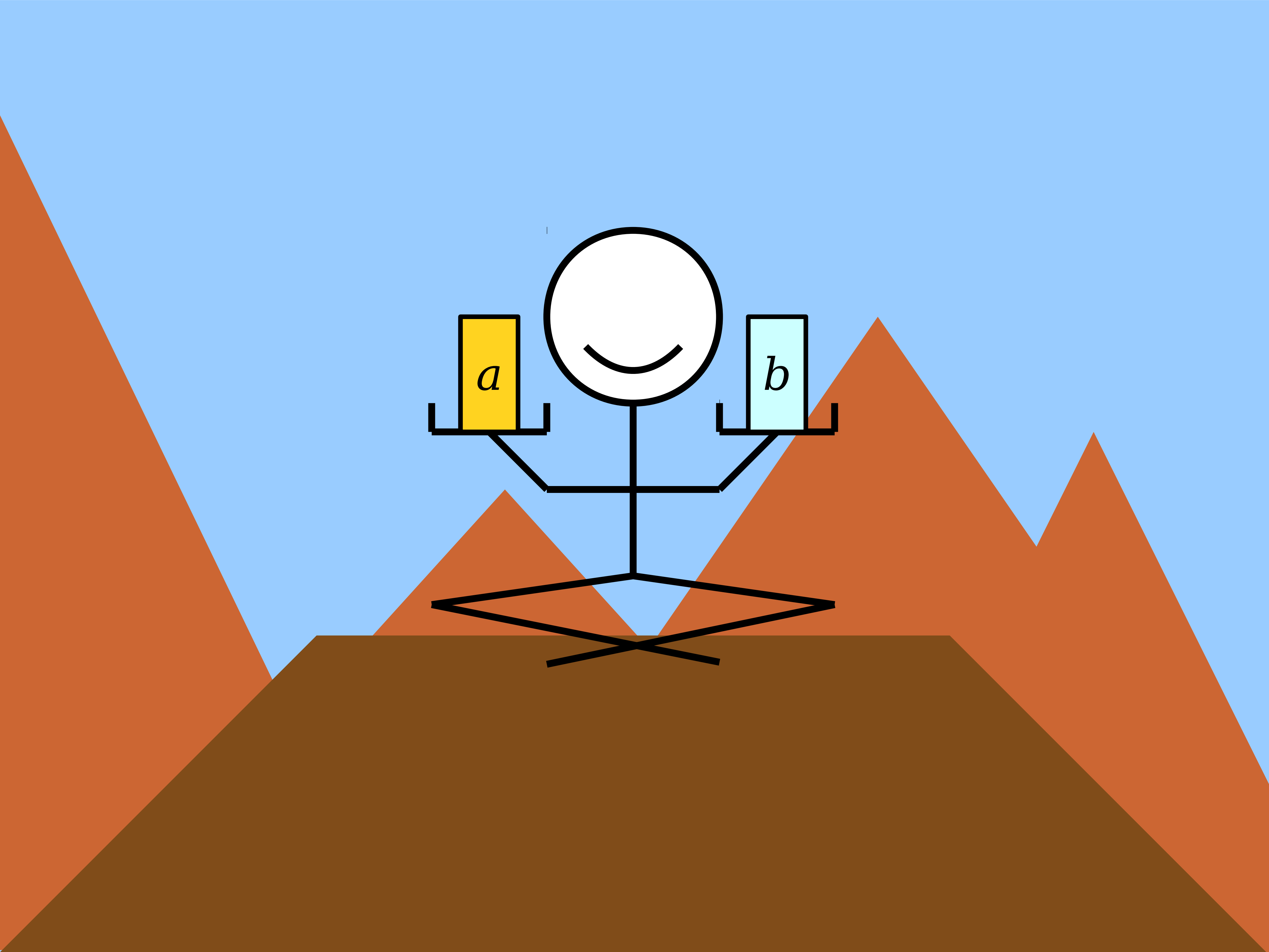


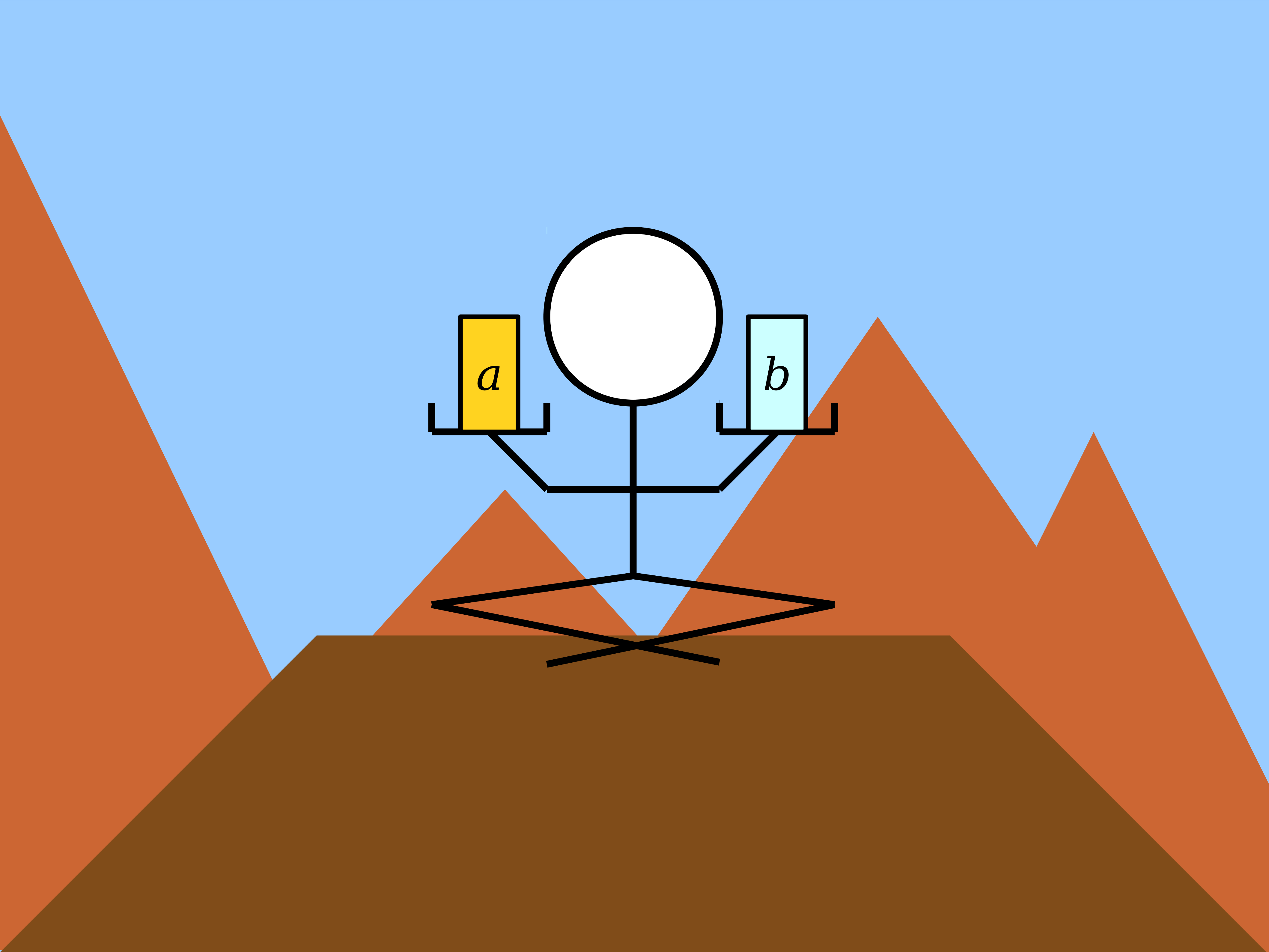

a
a


a
a

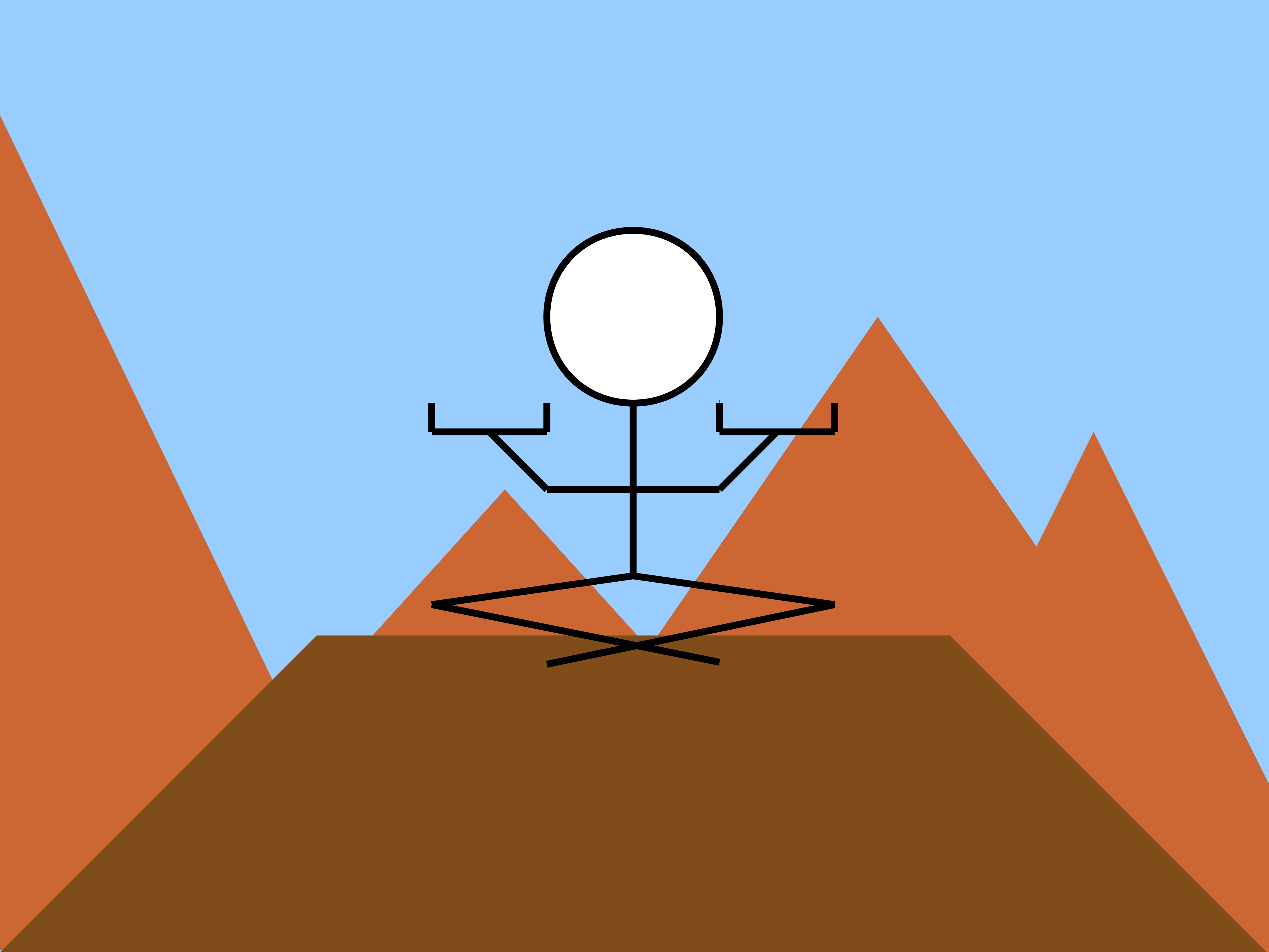


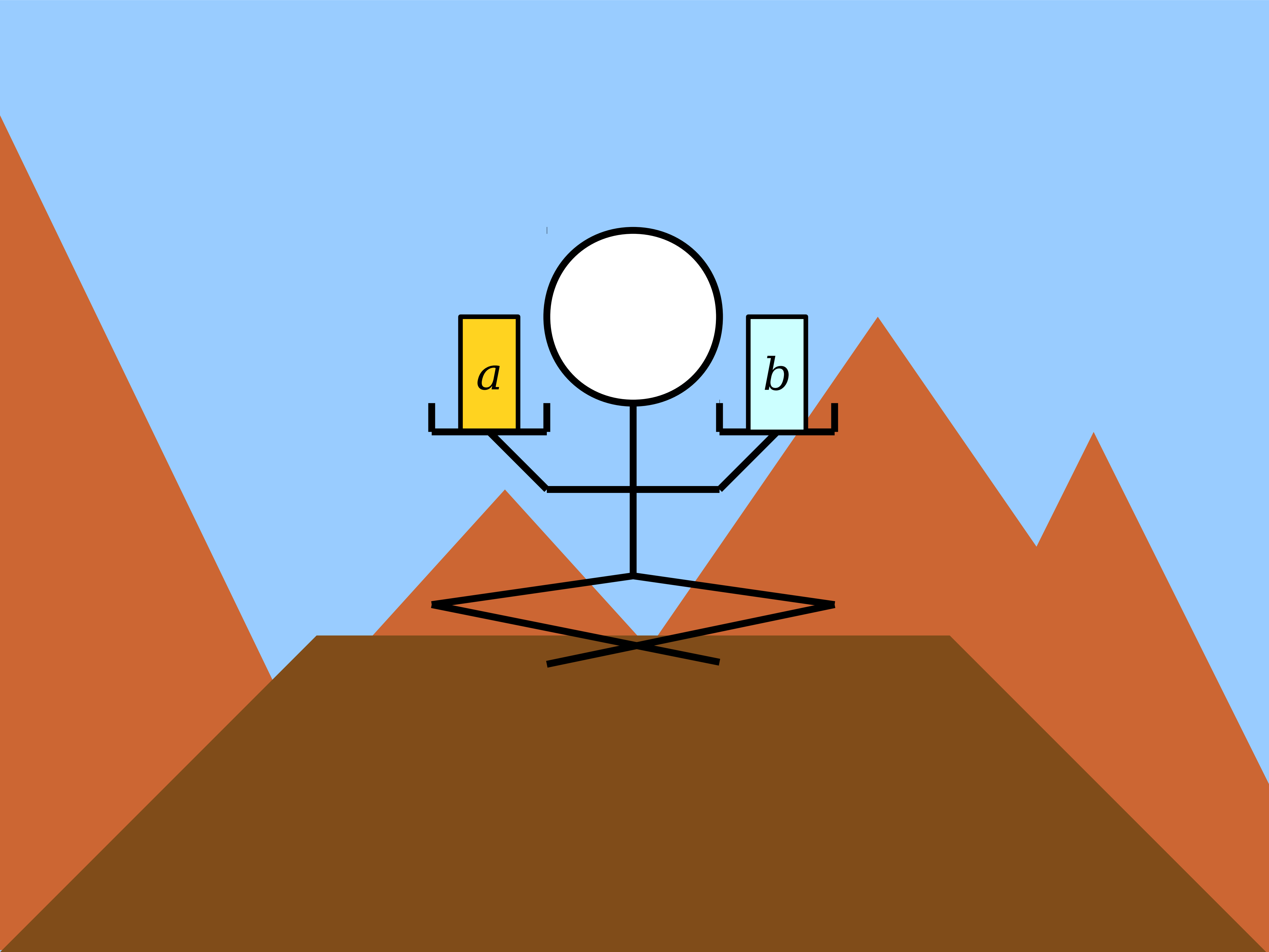

a
b

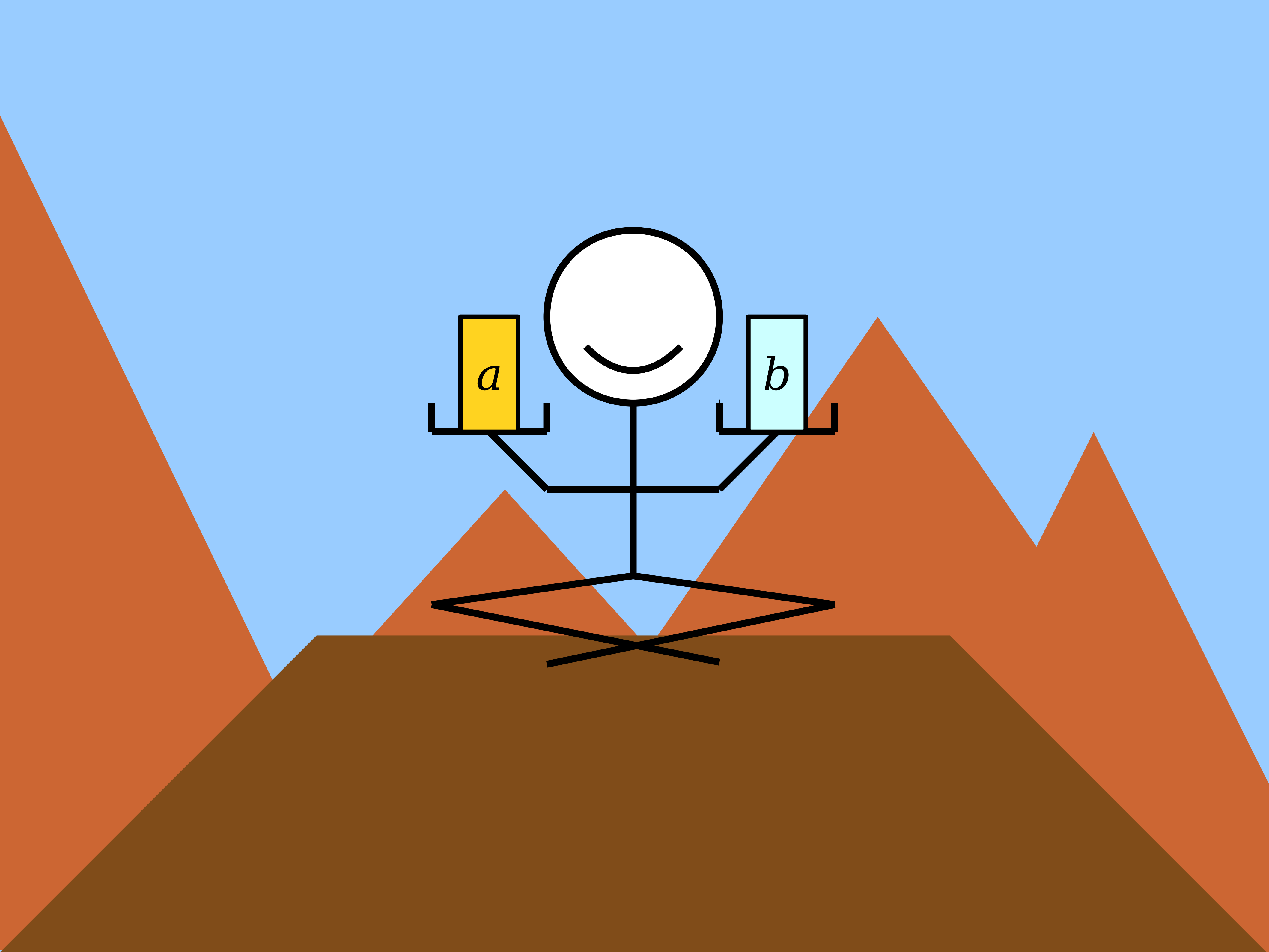

a
b

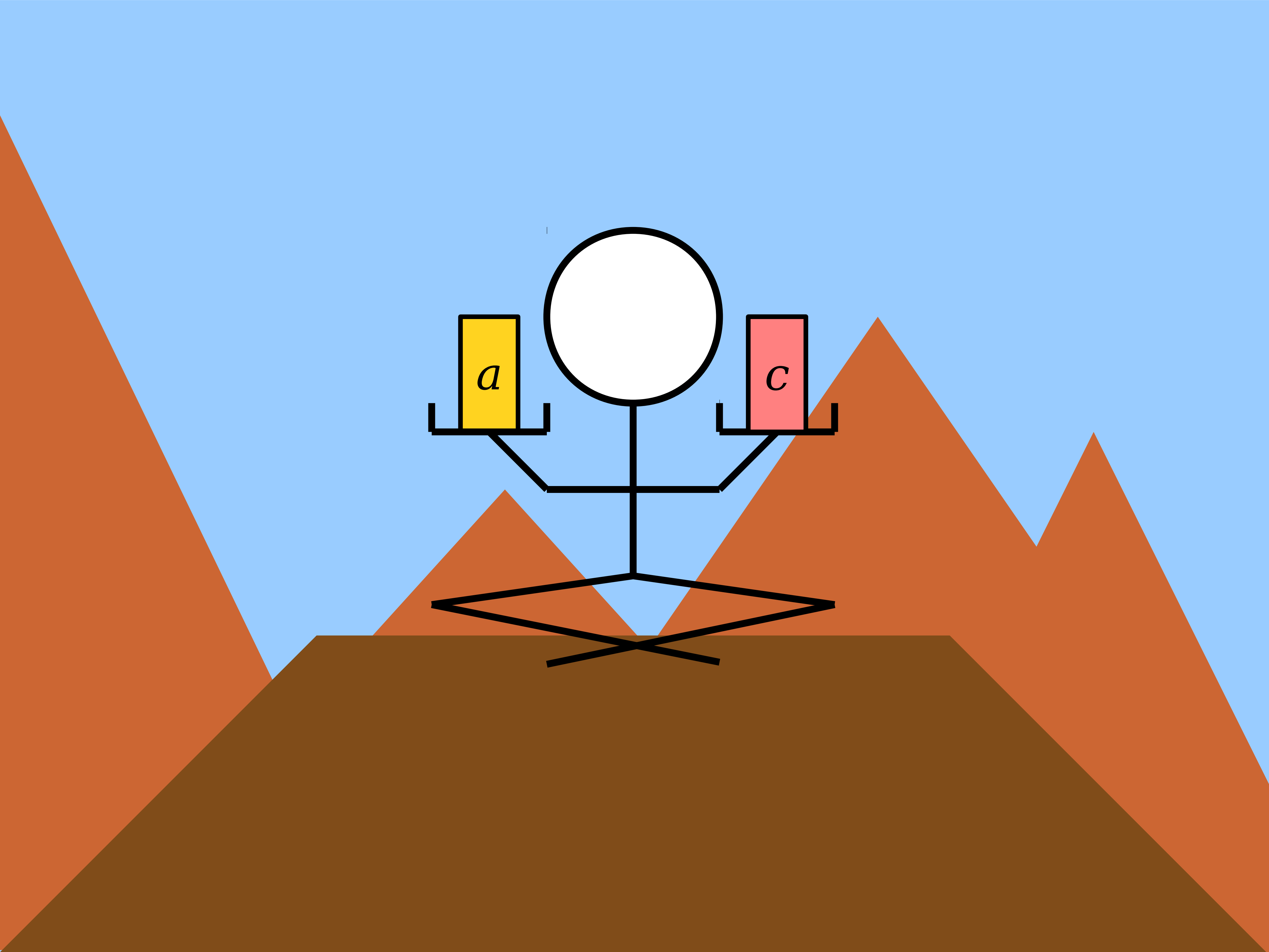

a
b

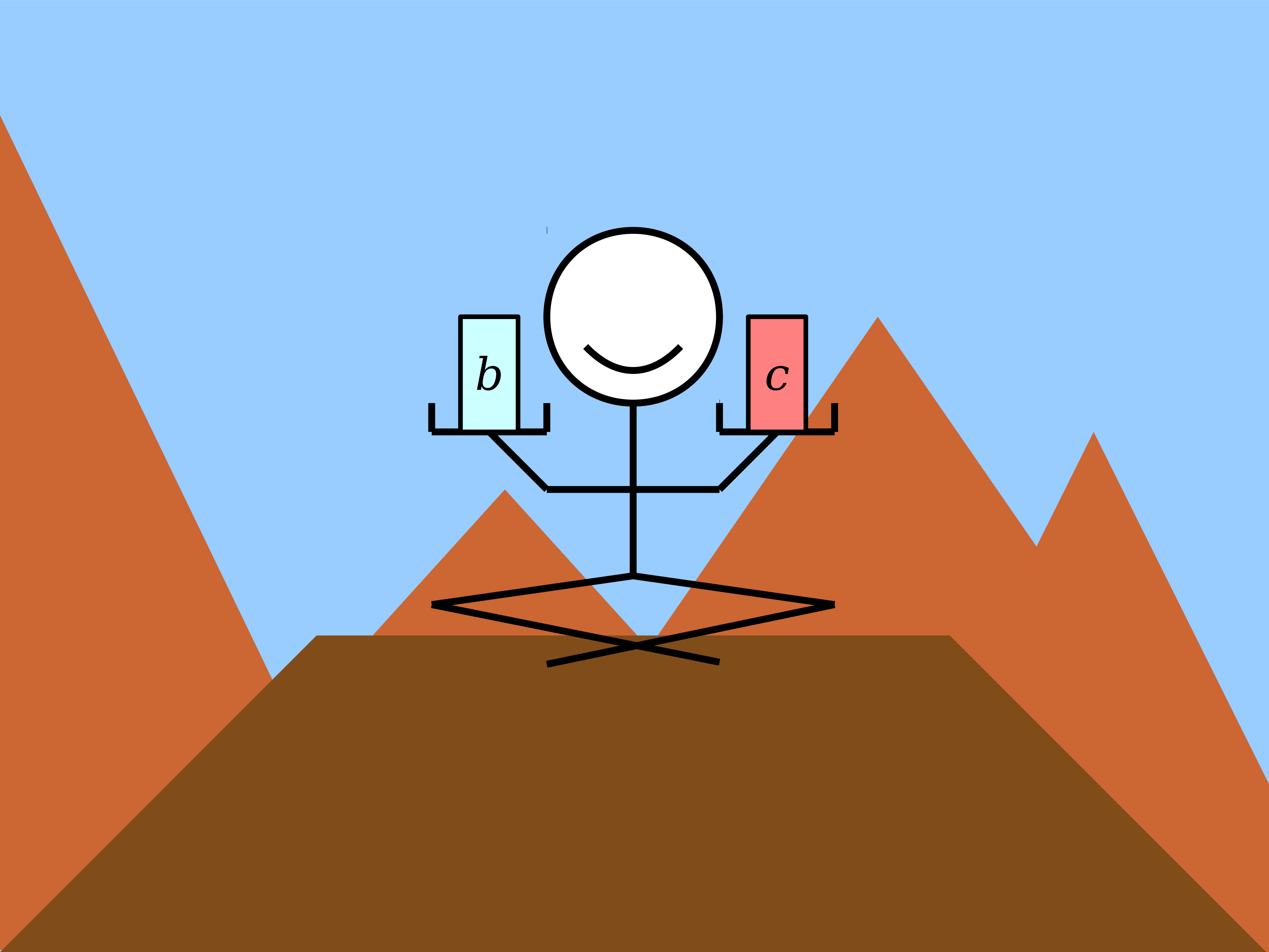

b
a

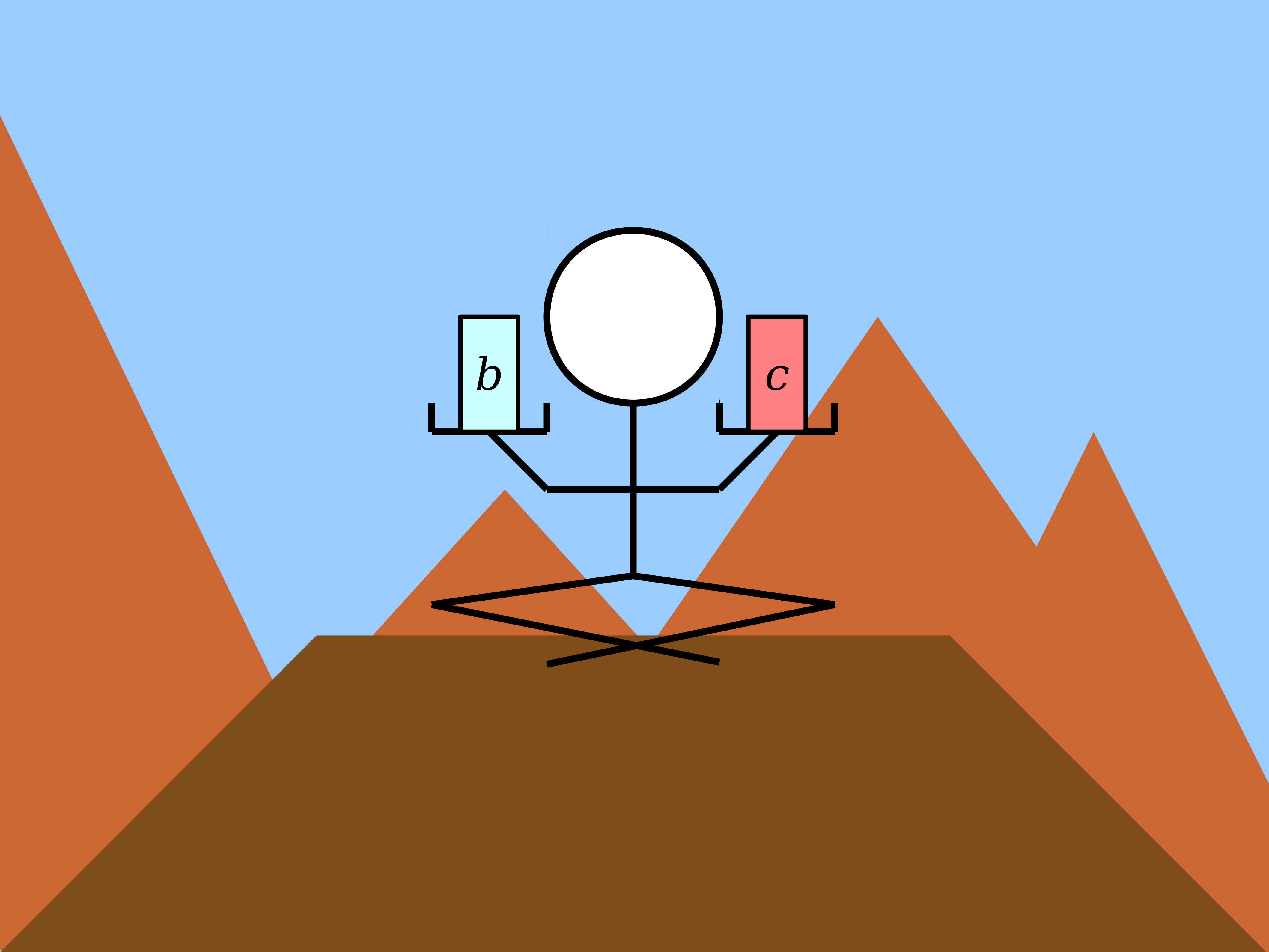


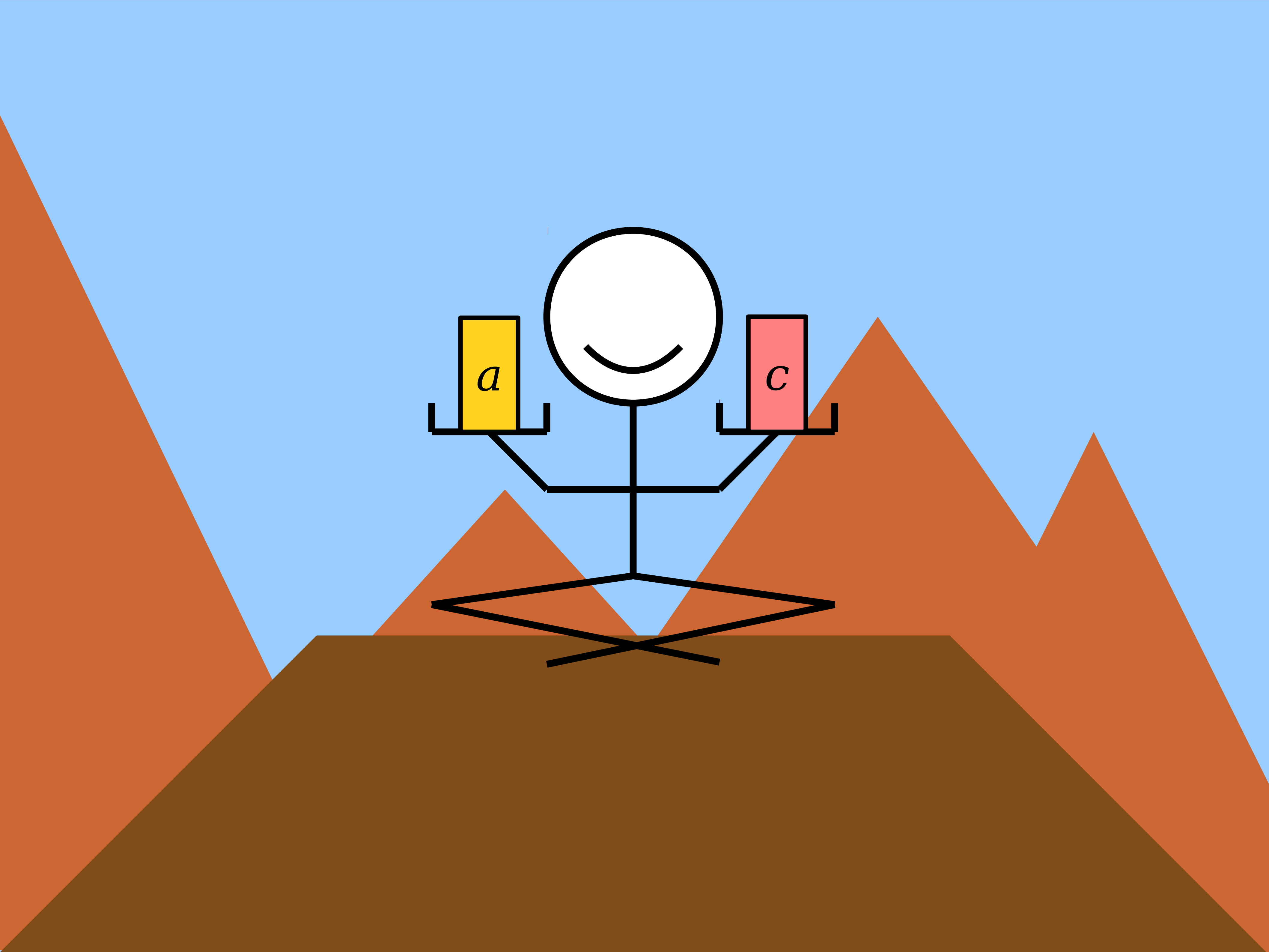

a
b

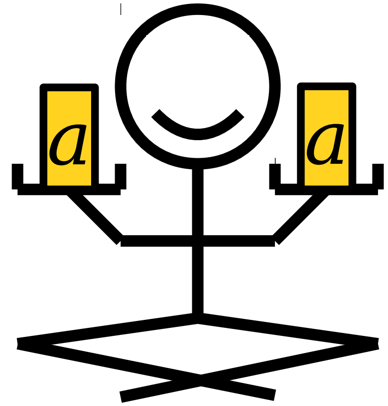

a
b

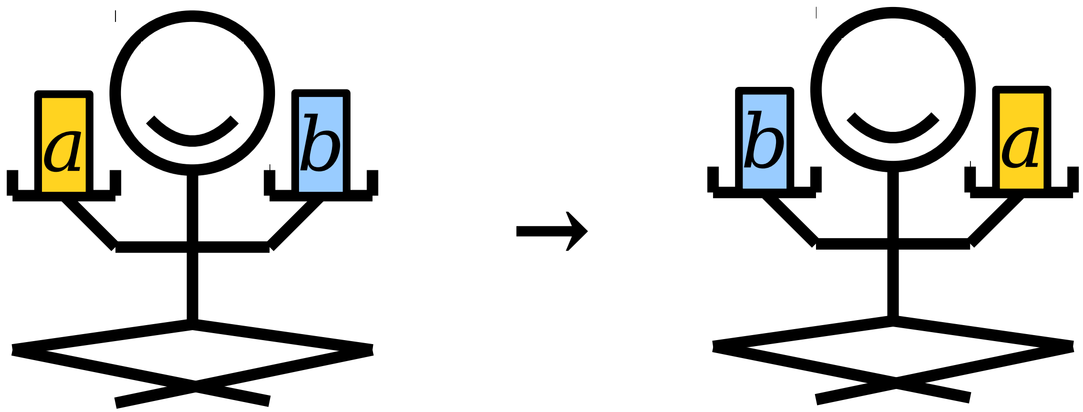

a
c

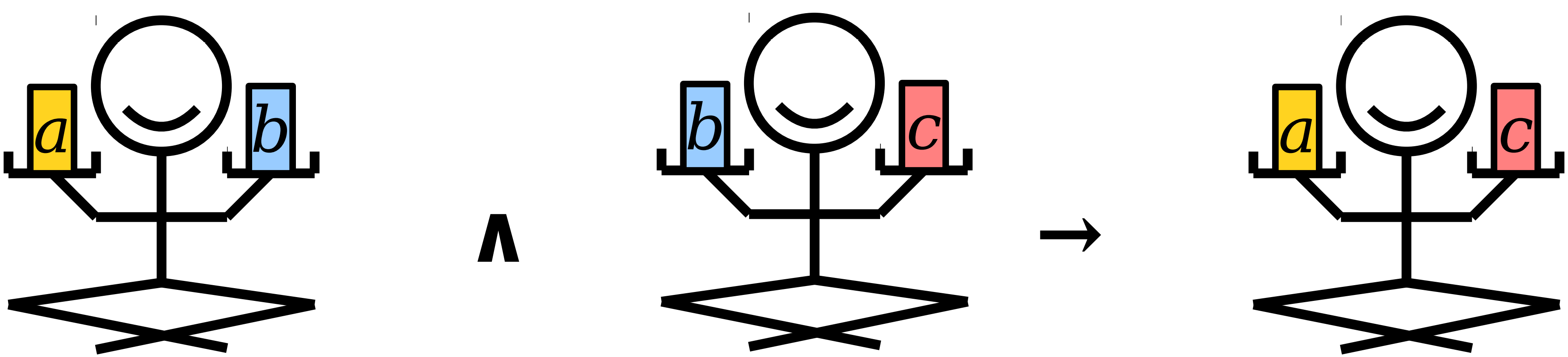

b
c


b
c


a
c


a
a


a
b
→
b
a


a
b
∧
b
c
→
a
c

$$aRa$$

$$aRb \rightarrow bRa$$

$$aRb \wedge bRc \rightarrow aRc$$

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

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

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

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

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

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

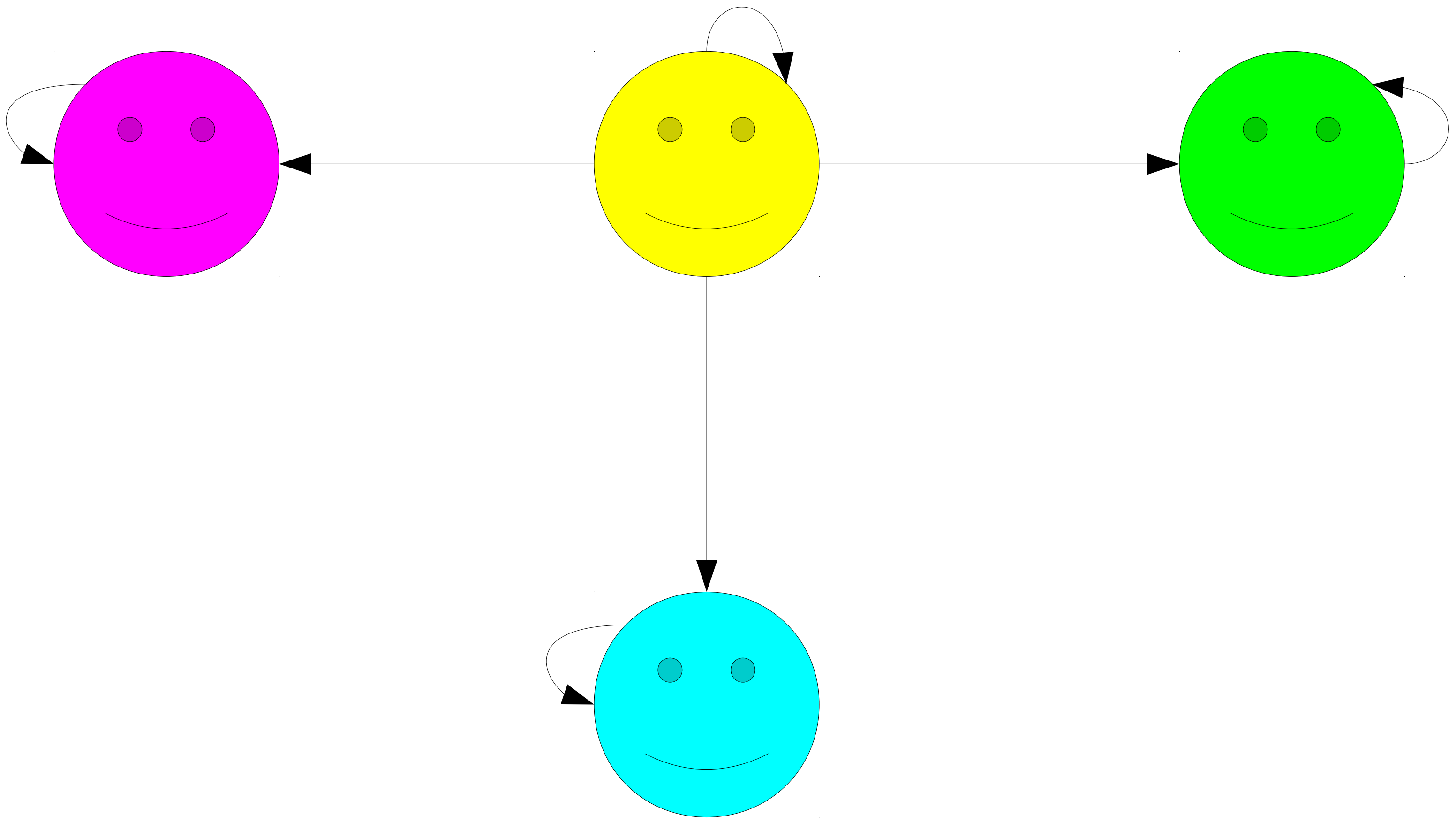
Reflexivity

- 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 first-order statement is true:

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

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

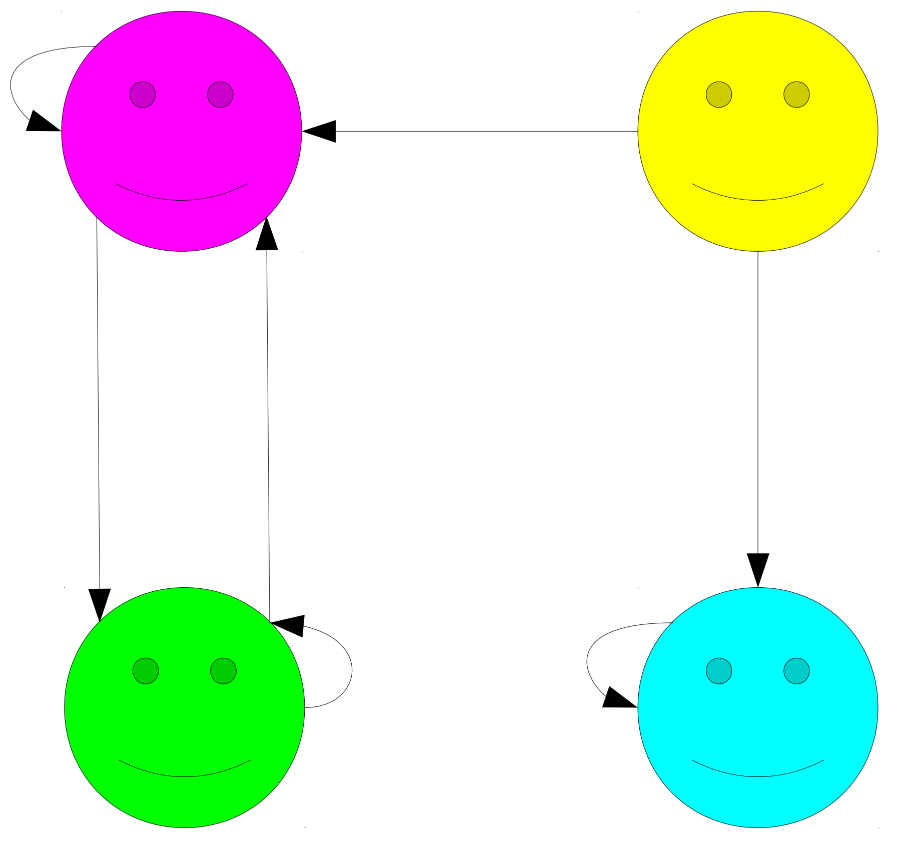
Reflexivity Visualized



$\forall a \in A.\ aRa$

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

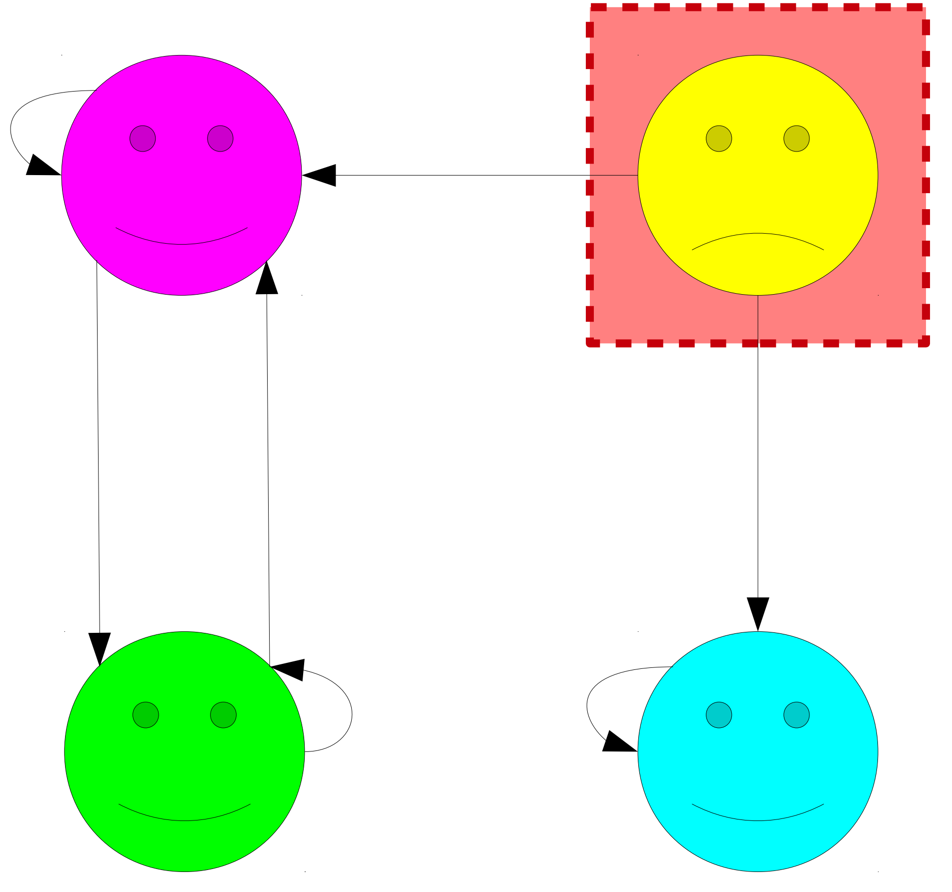
Is This Relation Reflexive?



$\forall a \in A.\ aRa$

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

Is This Relation Reflexive?



$\forall a \in A.\ aRa$

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

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

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

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

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

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

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

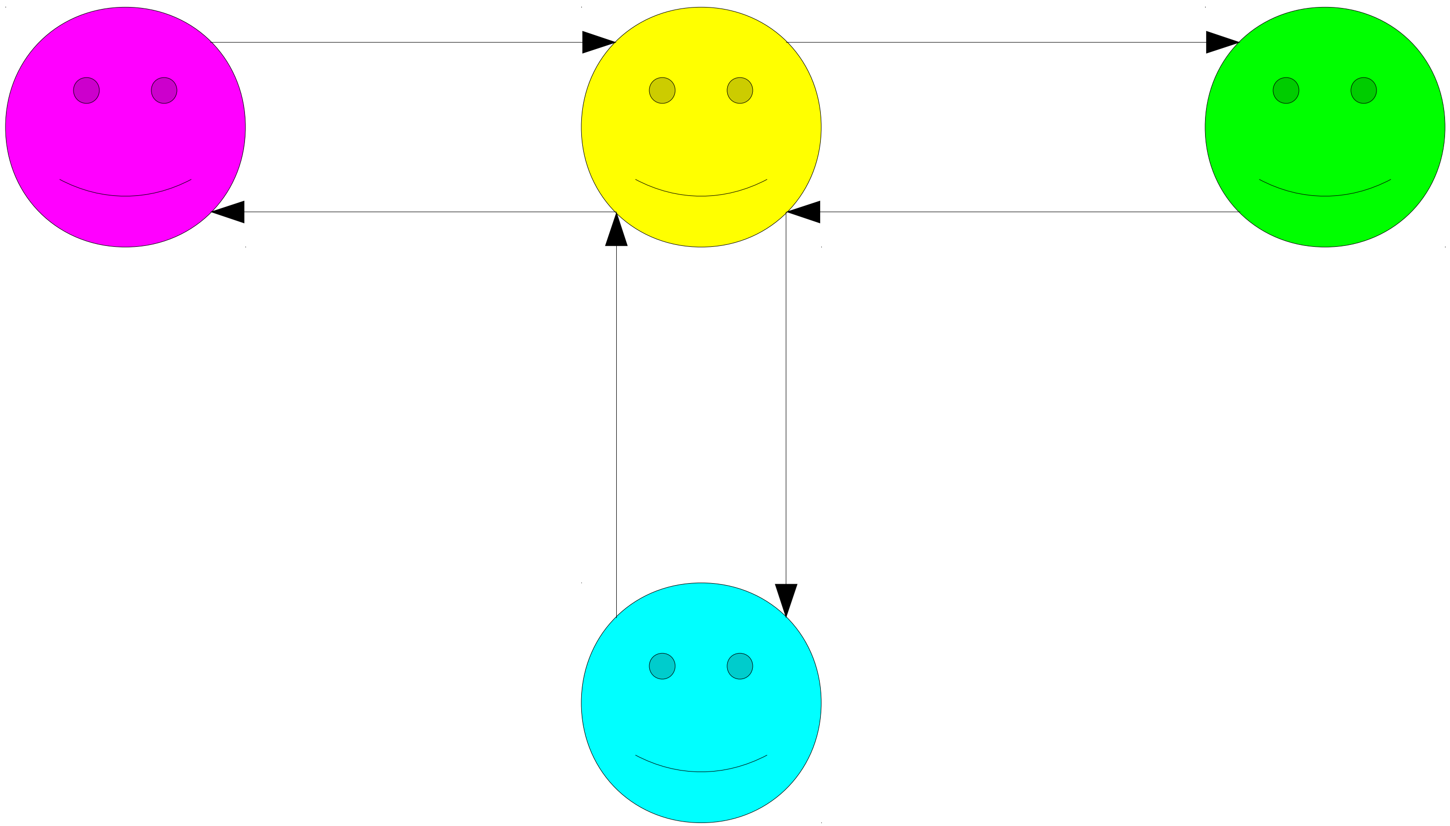
Symmetry

- 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 the following first-order statement is true about R:

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

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

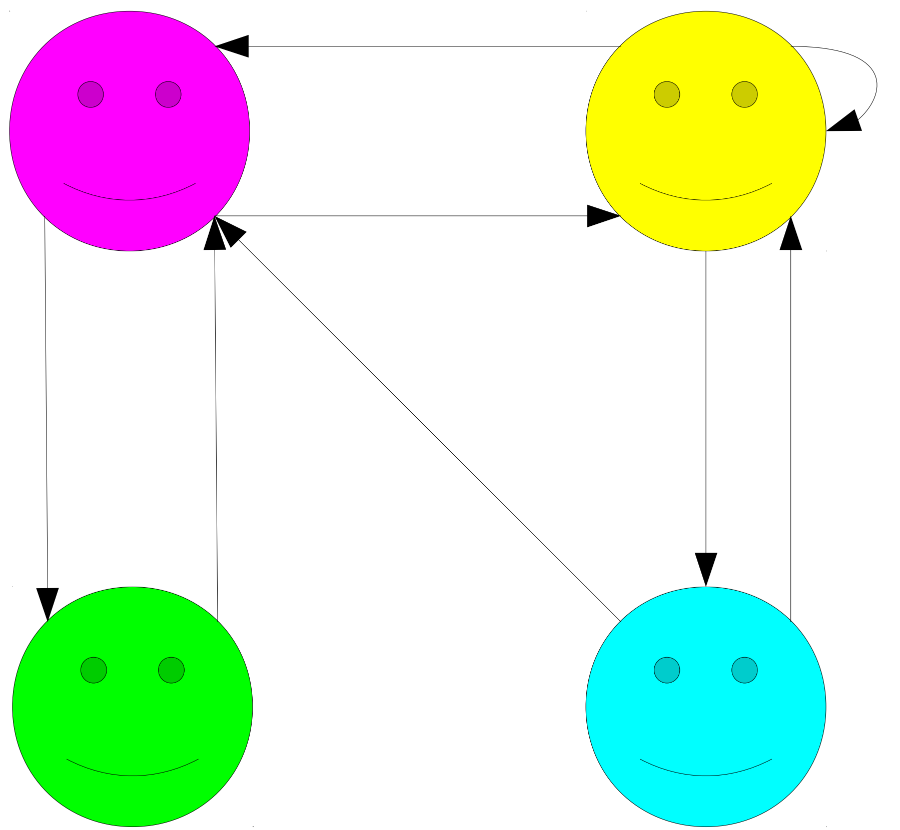
Symmetry Visualized



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

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

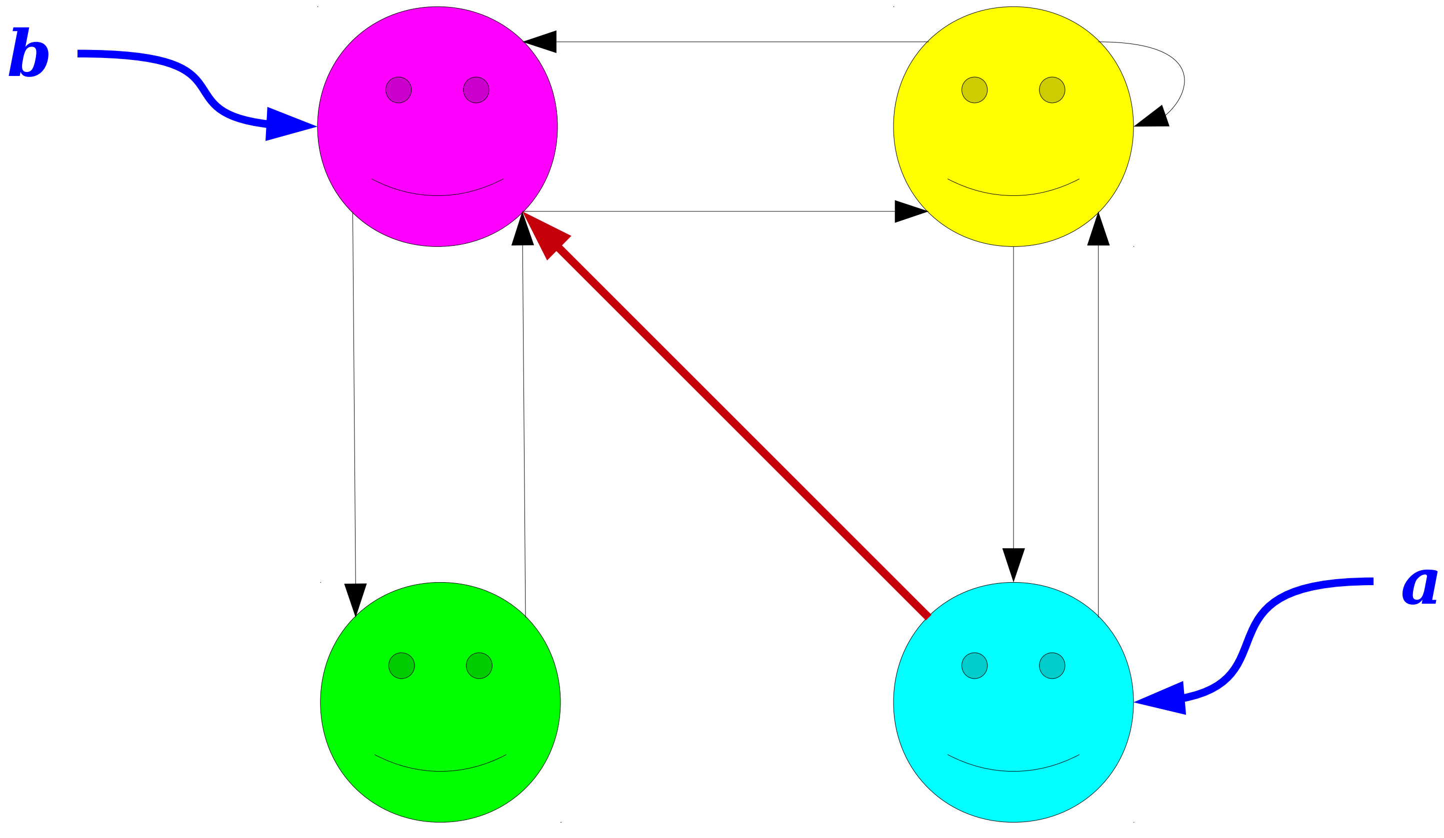
Is This Relation Symmetric?



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

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

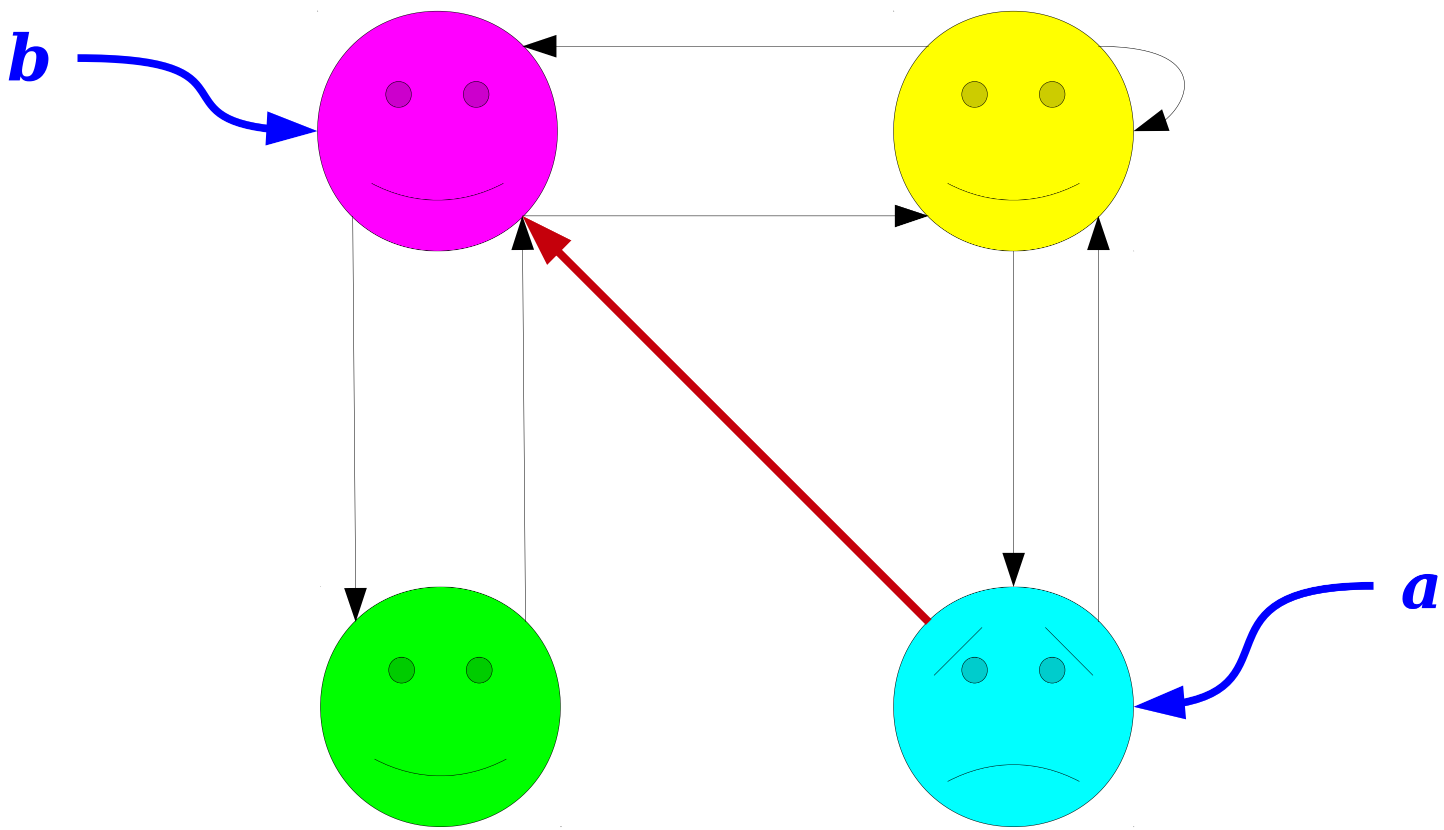
Is This Relation Symmetric?



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

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

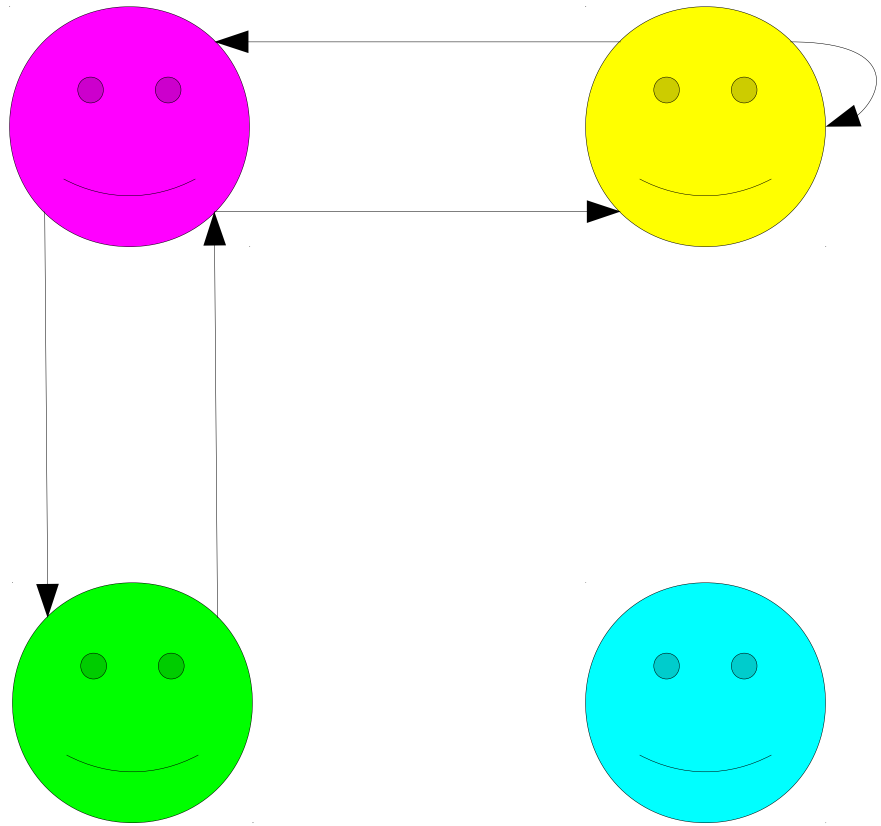
Is This Relation Symmetric?



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

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

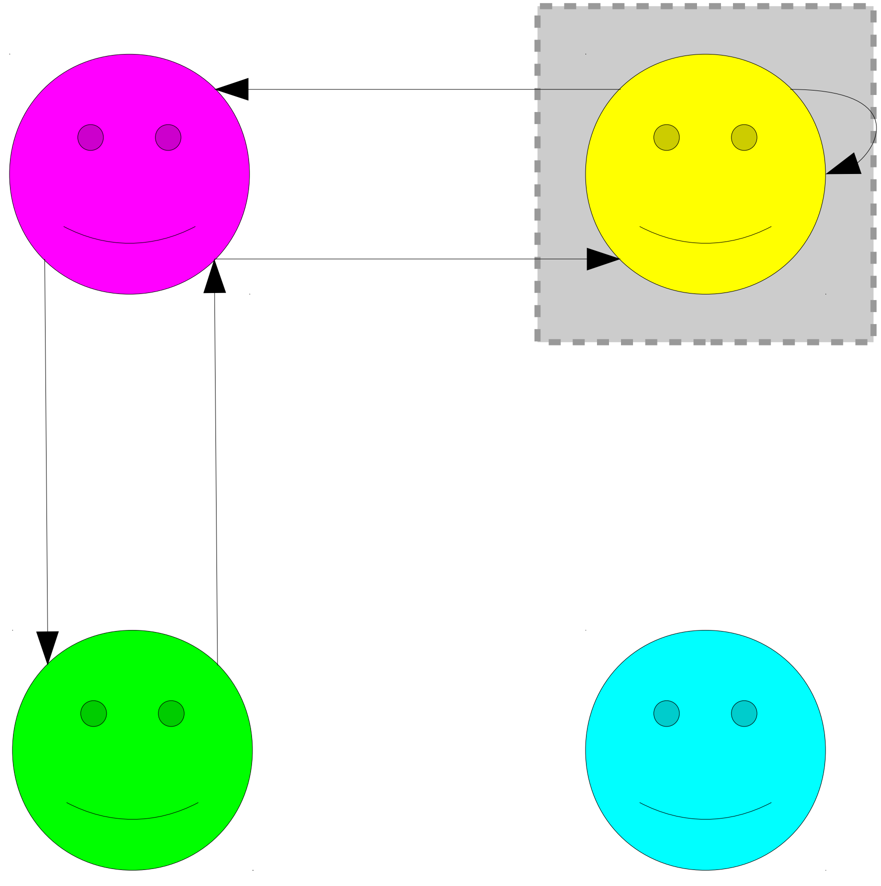
Is This Relation Symmetric?



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

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

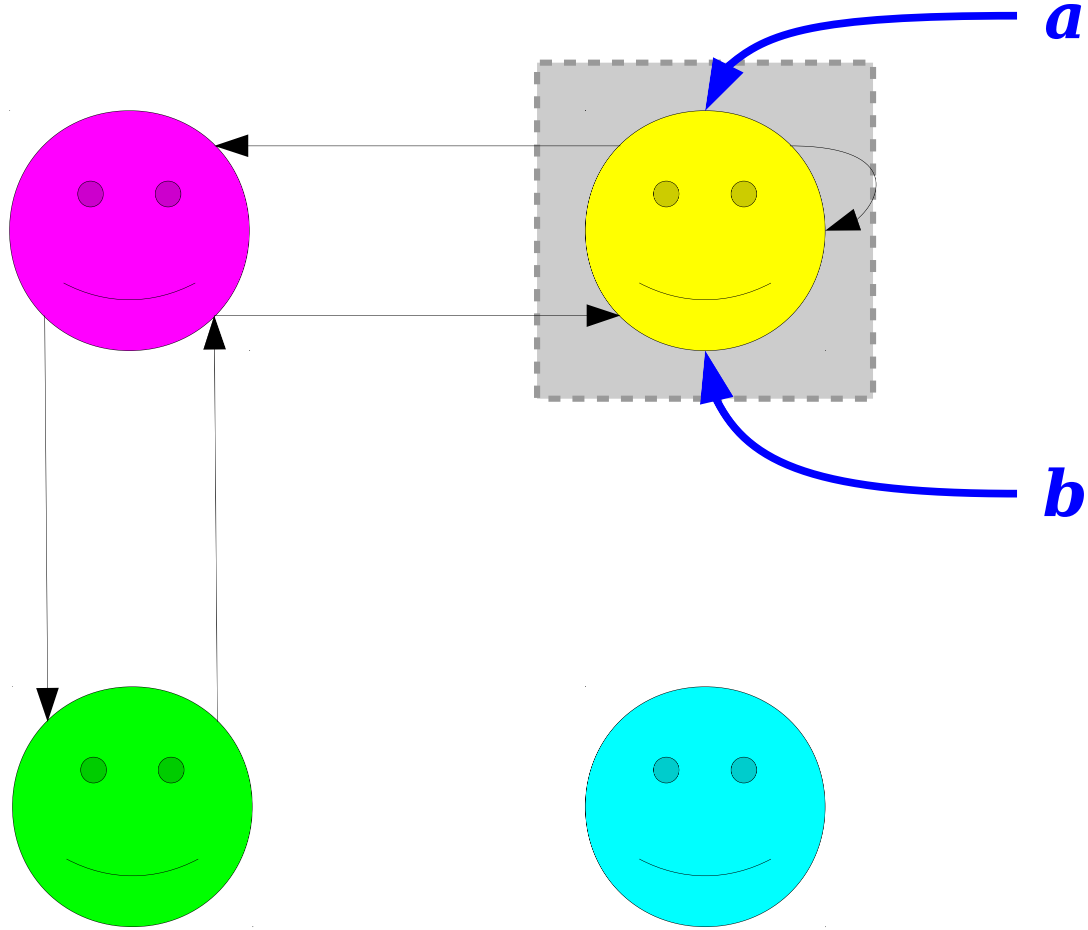
Is This Relation Symmetric?



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

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

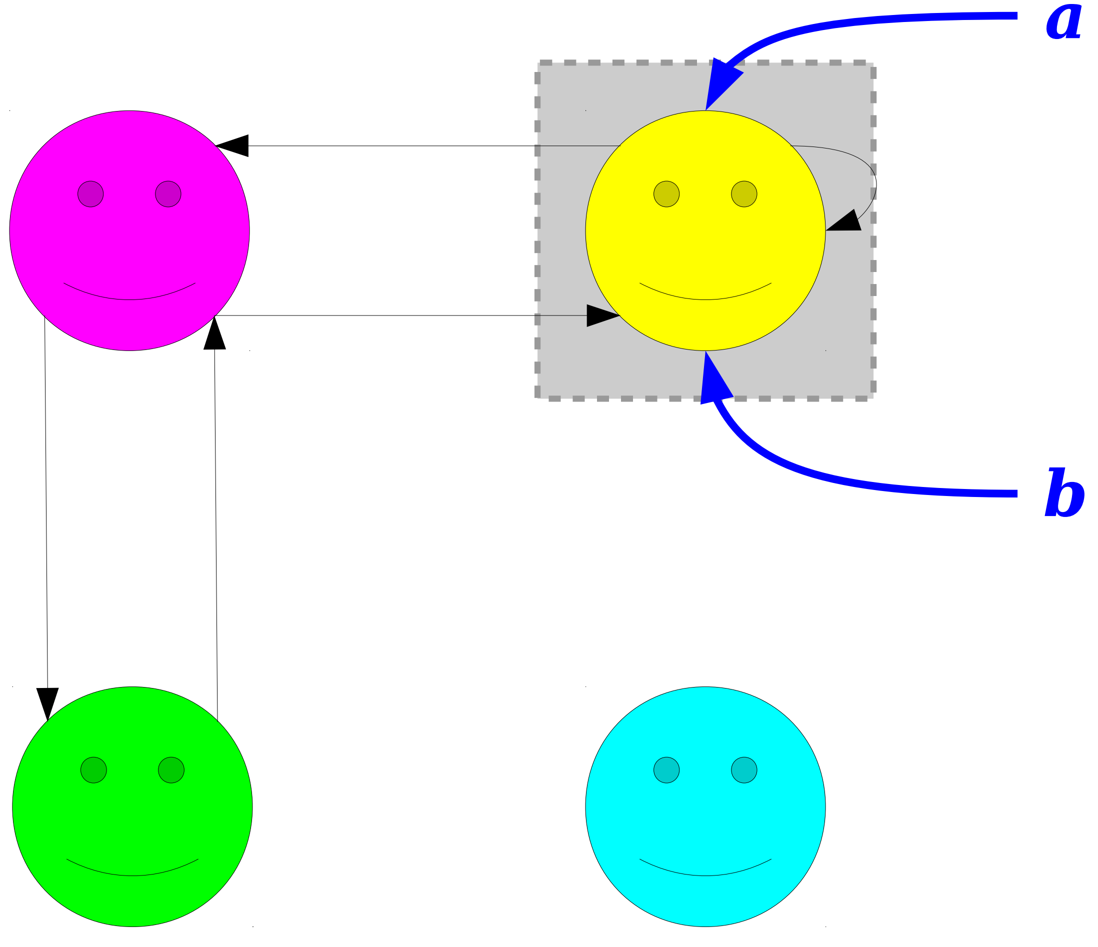
Is This Relation Symmetric?



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

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

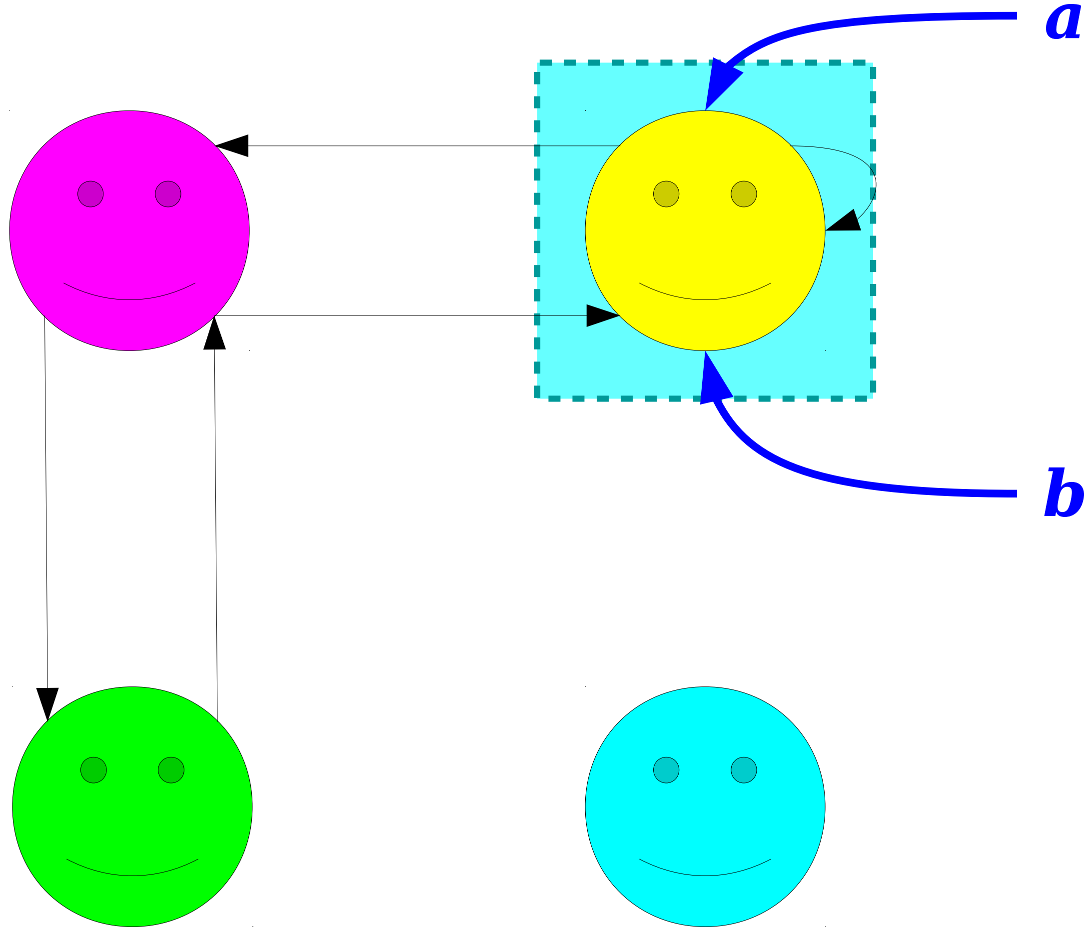
Is This Relation Symmetric?



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

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

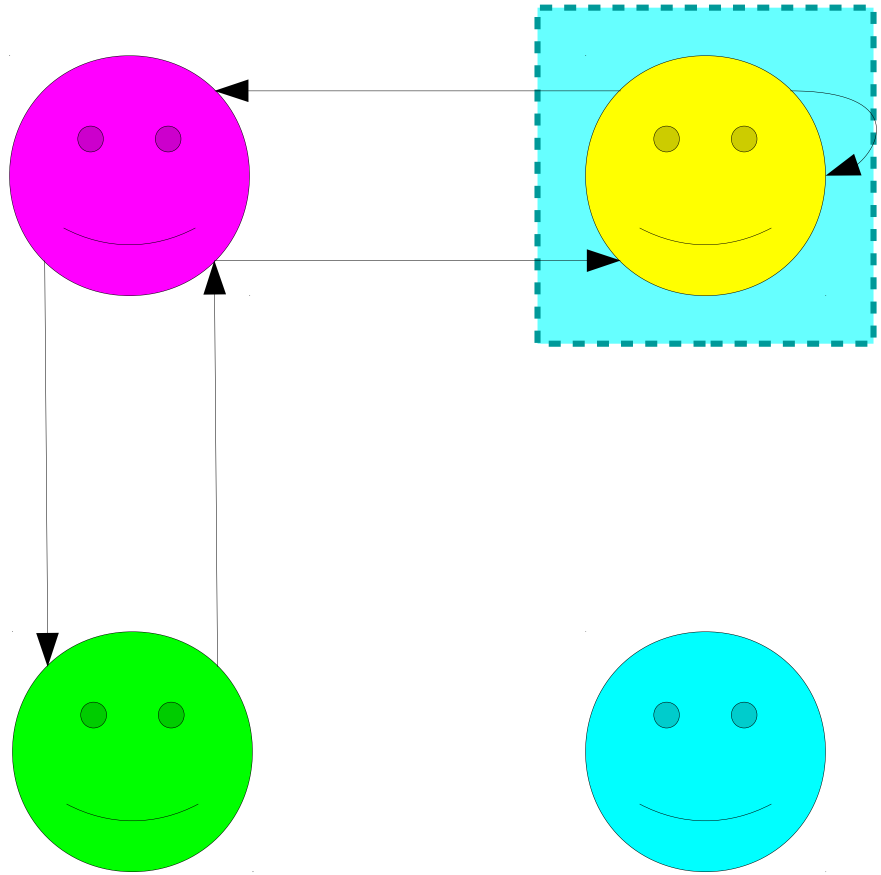
Is This Relation Symmetric?



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

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

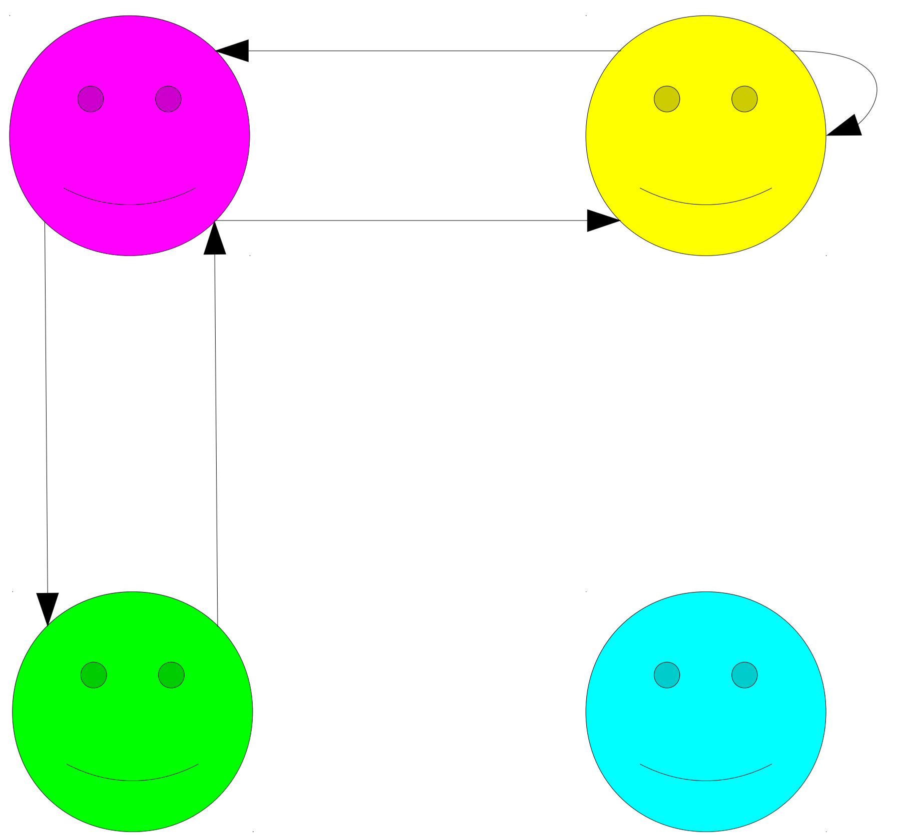
Is This Relation Symmetric?



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

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

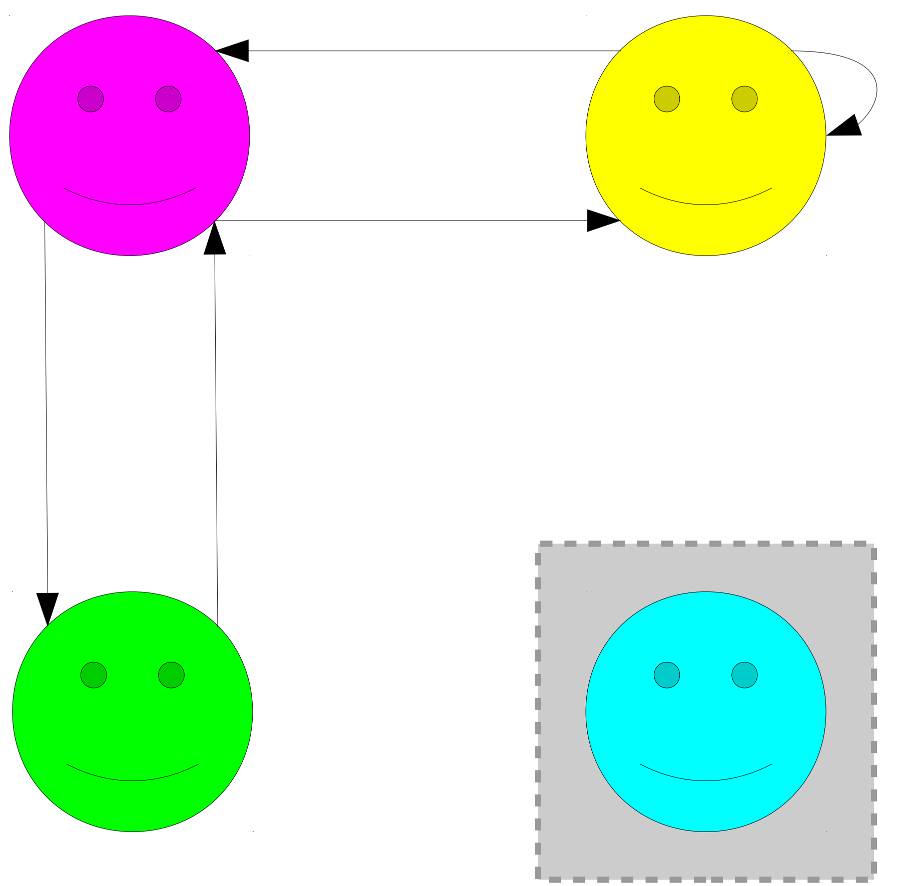
Is This Relation Symmetric?



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

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

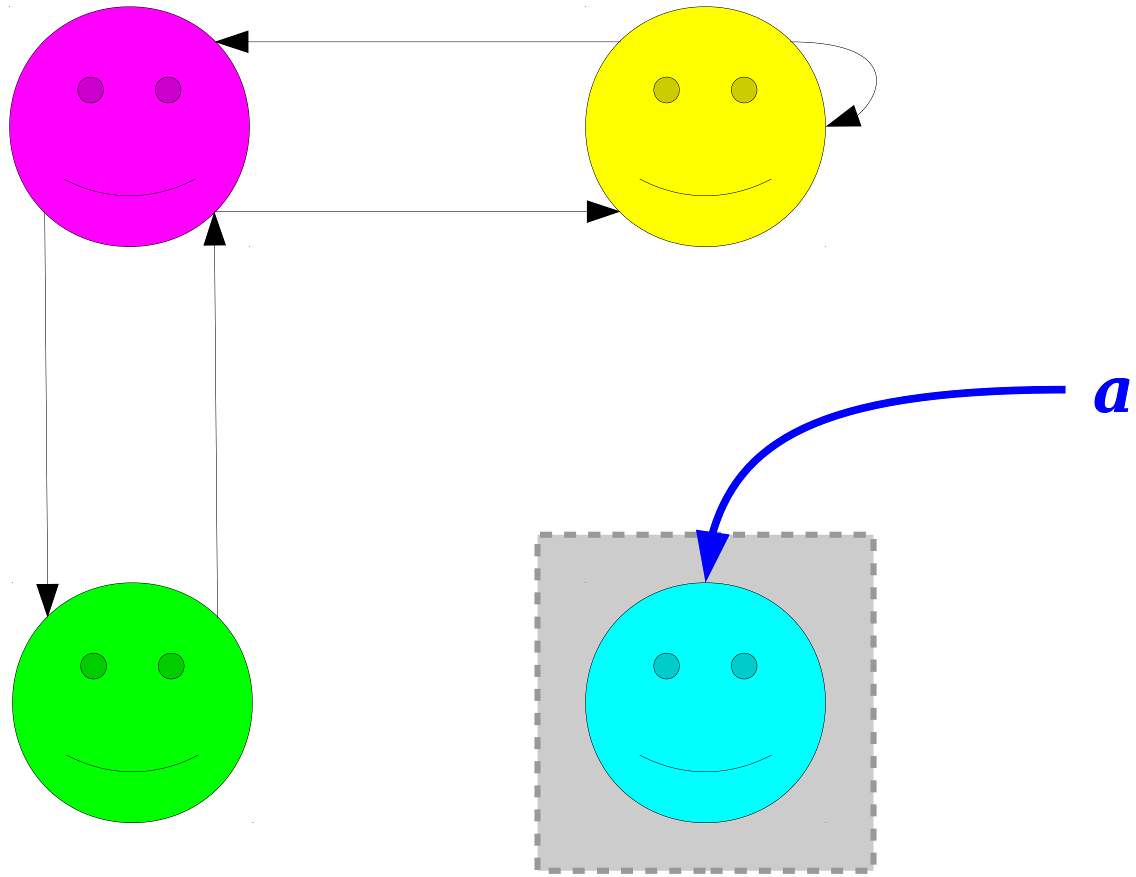
Is This Relation Symmetric?



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

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

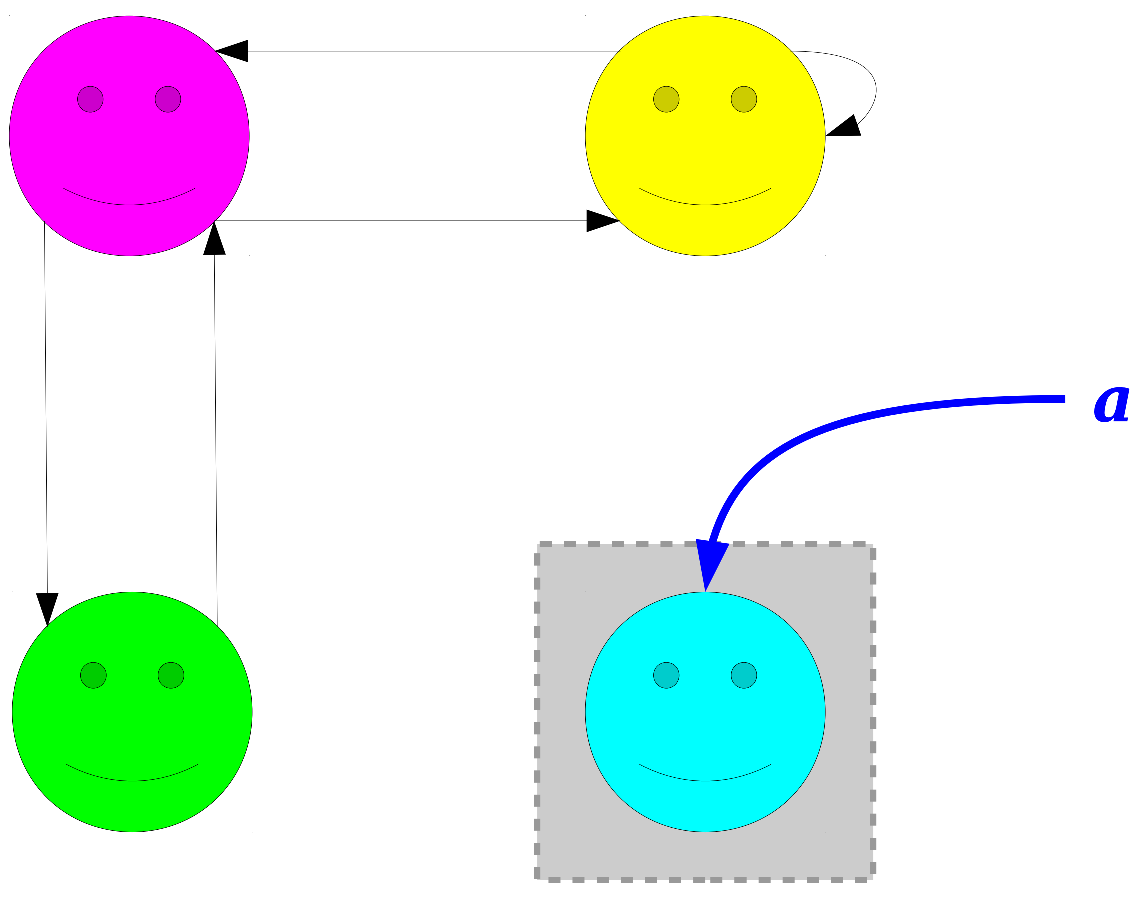
Is This Relation Symmetric?



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

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

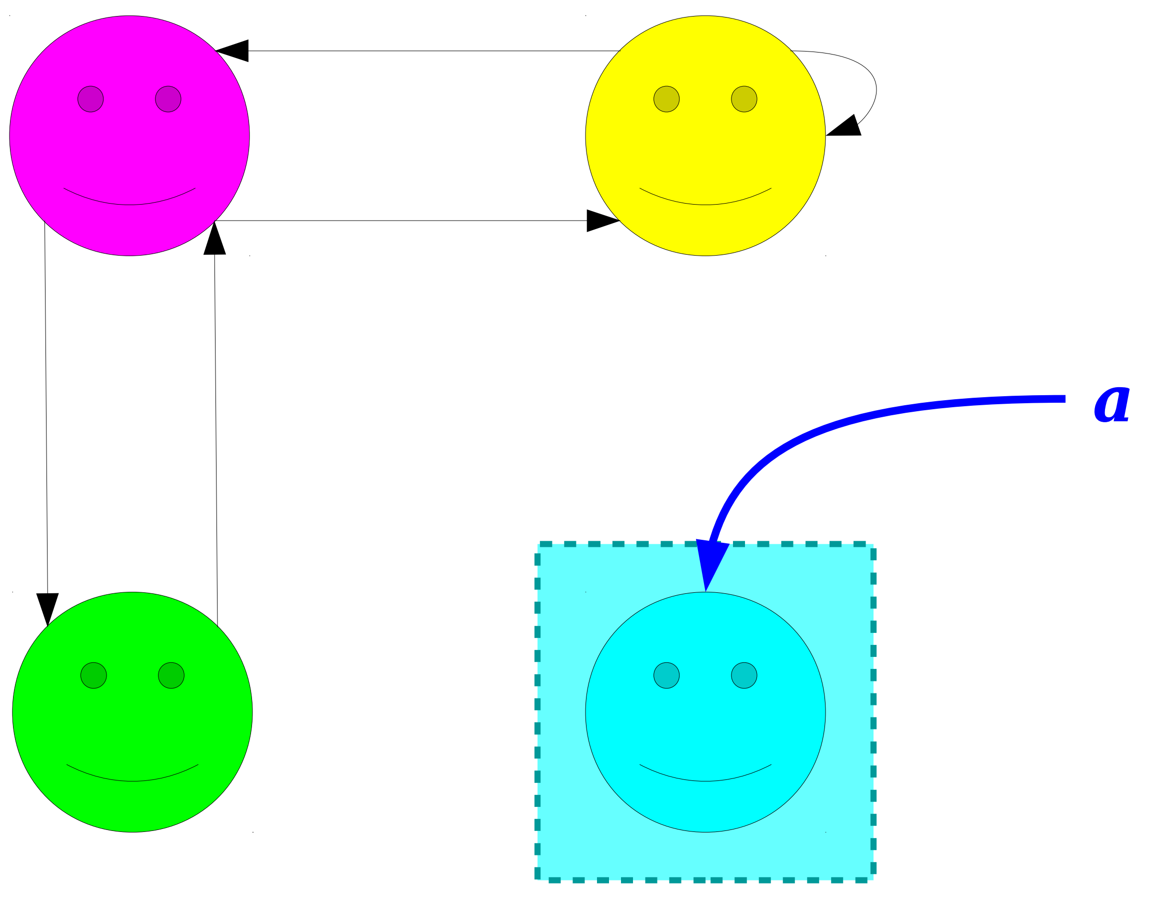
Is This Relation Symmetric?



∀*a* ∈ *A*. ∀*b* ∈ *A*. (*aRb* → *bRa*)

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

Is This Relation Symmetric?



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

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

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

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

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

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

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

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

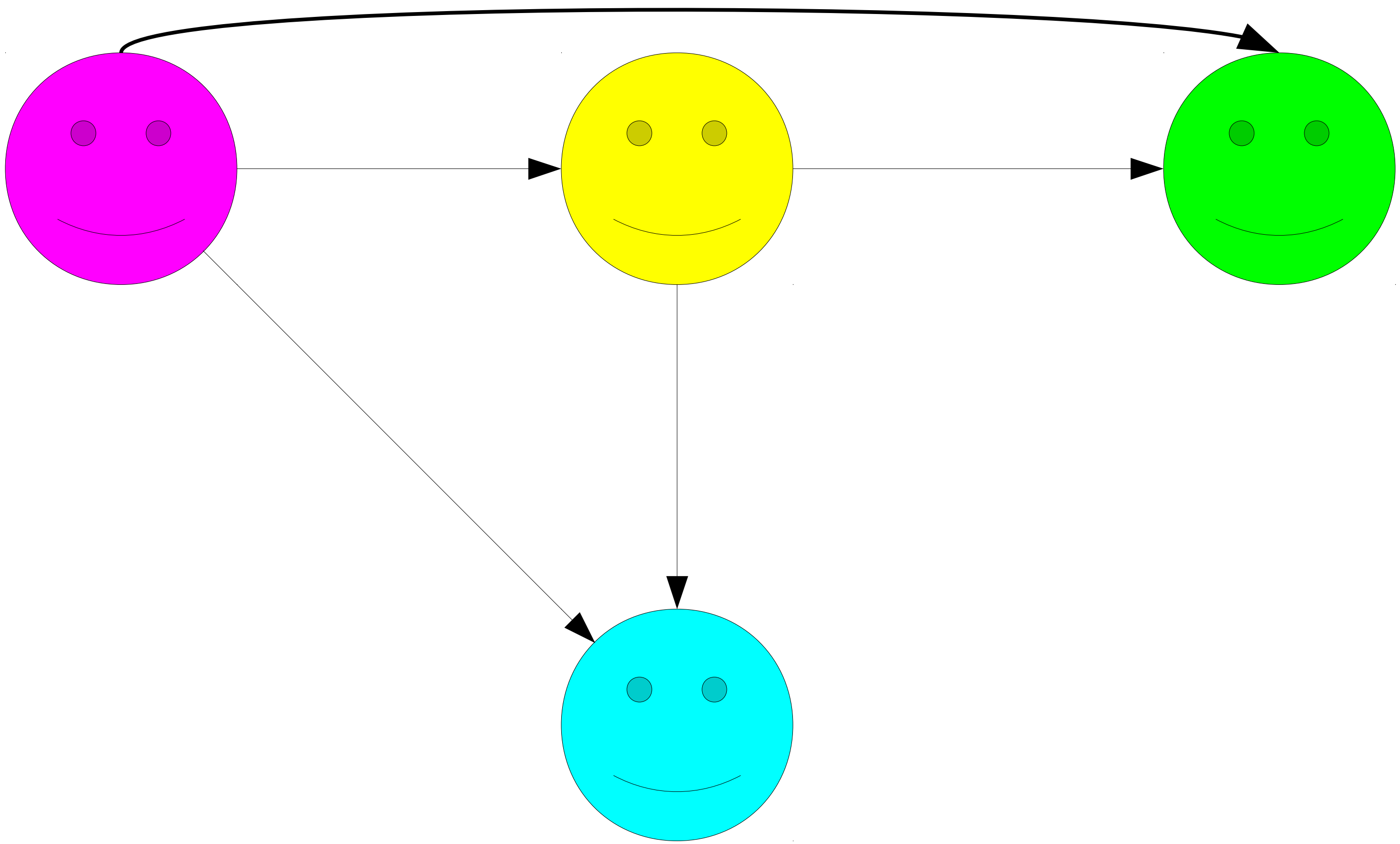
Transitivity

- 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 the following first-order statement is true about R:

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

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

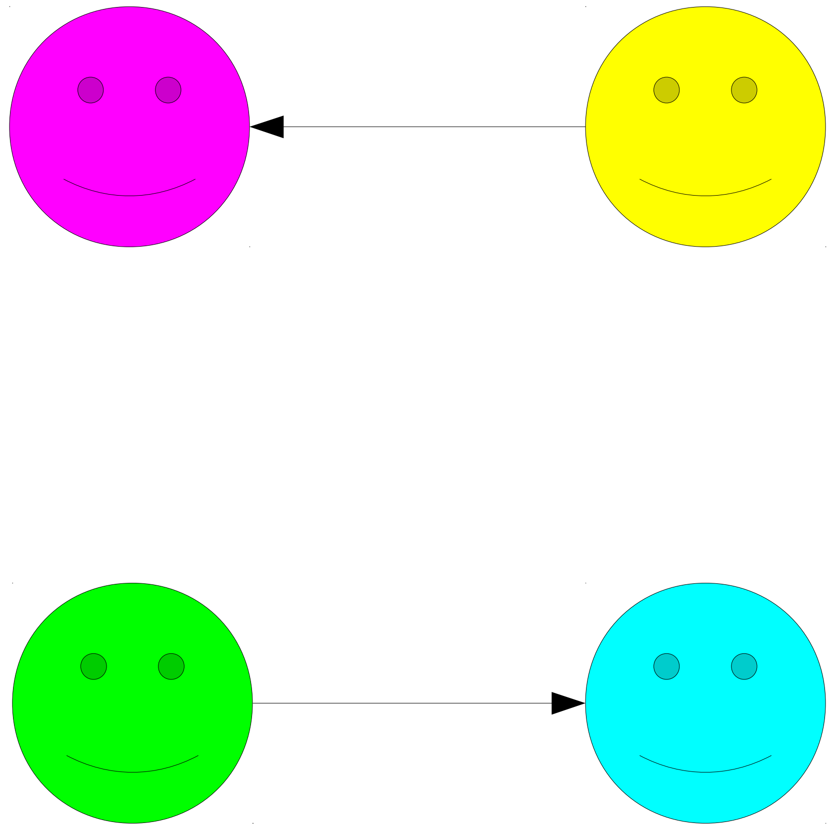
Transitivity Visualized



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

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

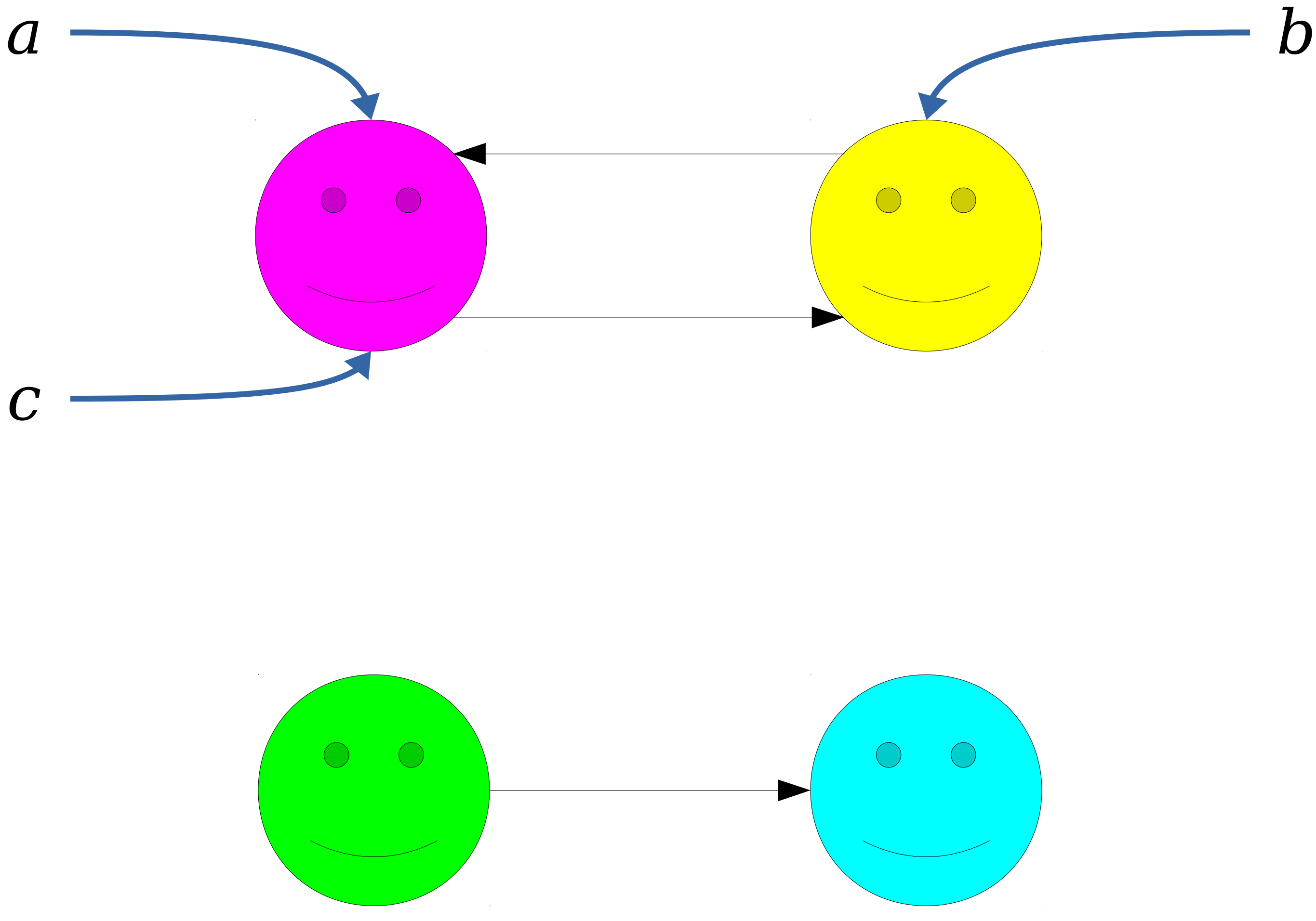
Is This Relation Transitive?



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

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

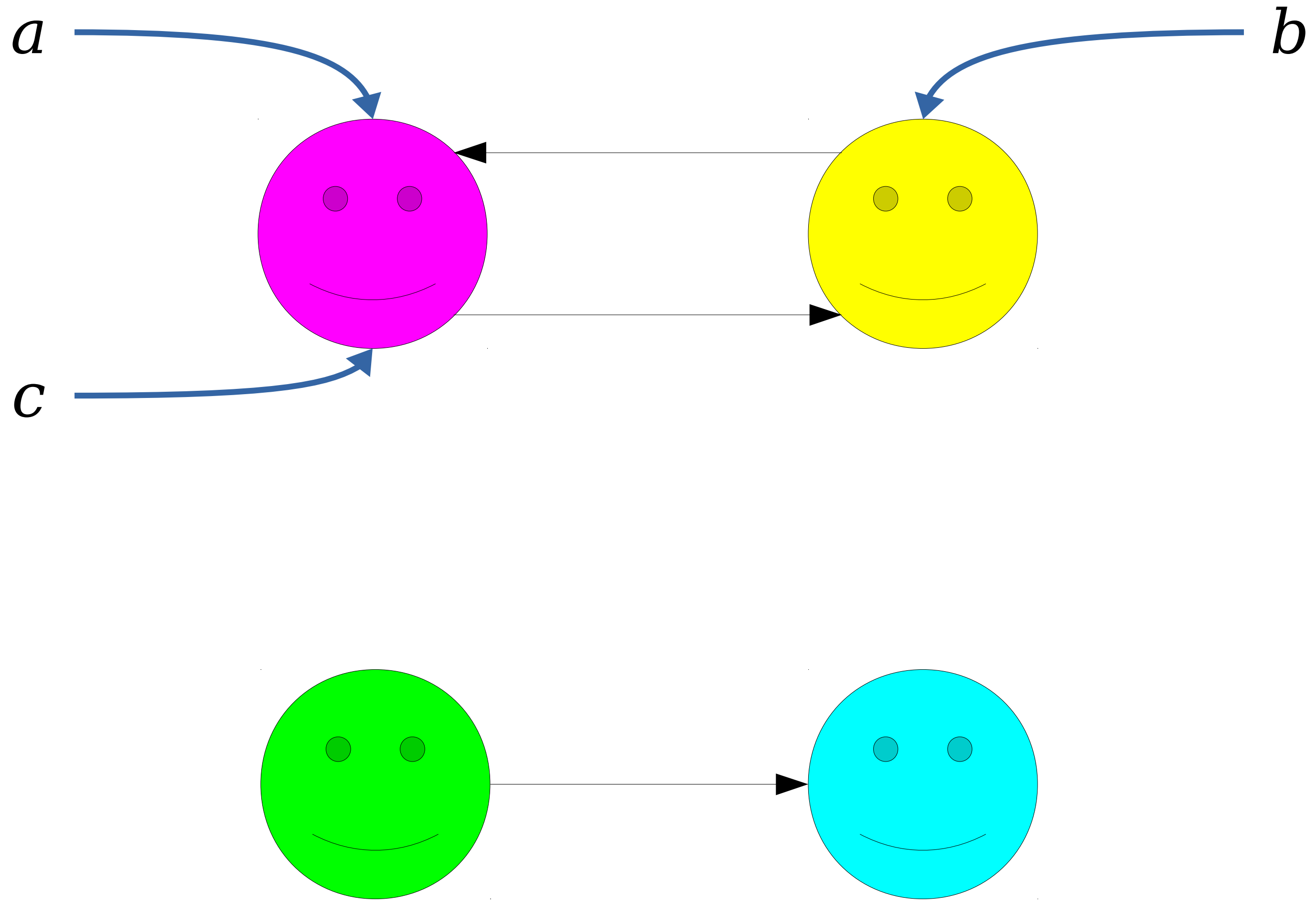
Is This Relation Transitive?



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

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

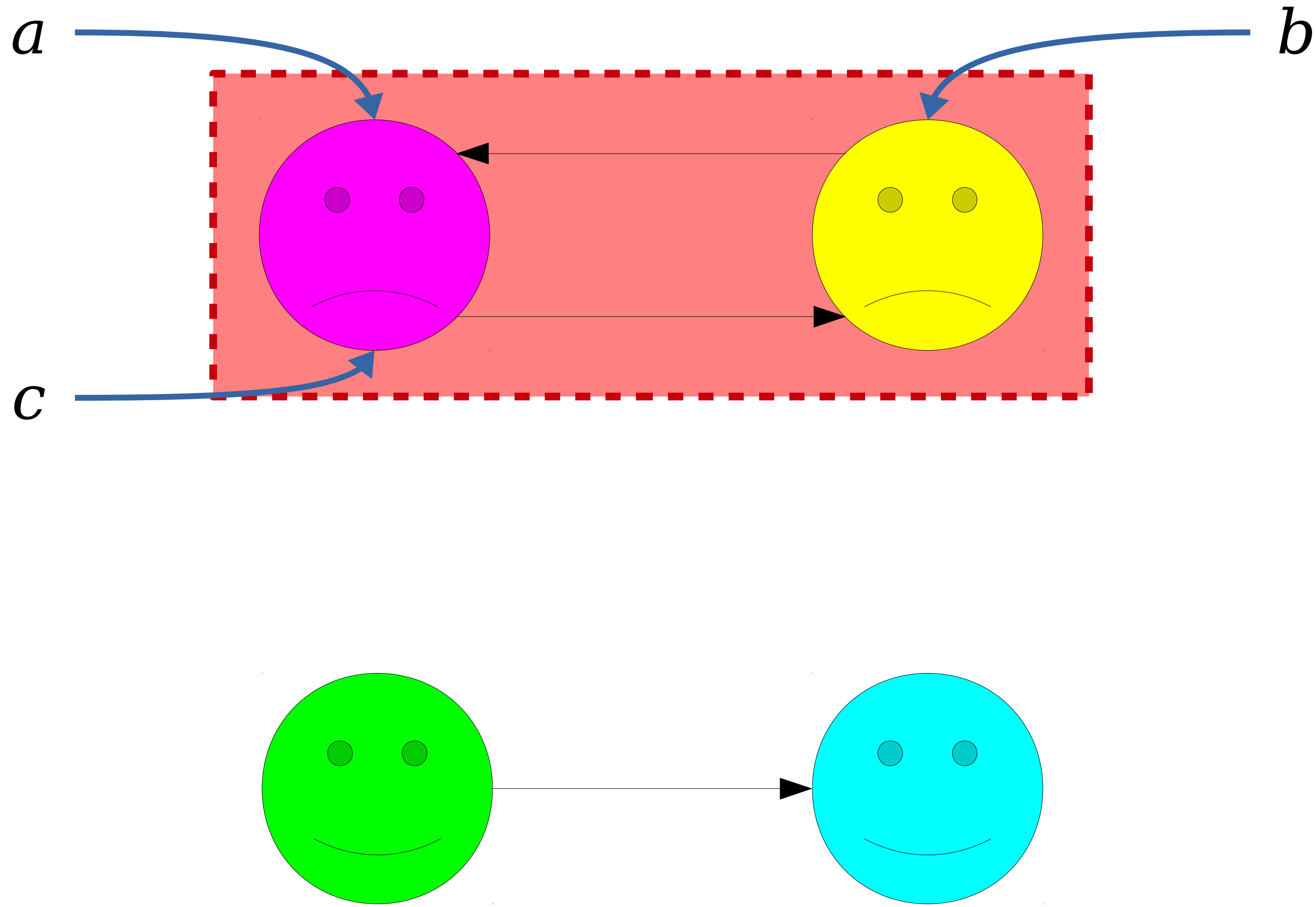
Is This Relation Transitive?



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

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

Is This Relation Transitive?



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

(*"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.

Binary relations give us a ***common language*** to describe ***common structures***.

Equivalence Relations

- Most modern programming languages include some sort of hash table data structure.
  - Java: `HashMap`
  - C++: `std::unordered_map`
  - Python: `dict`
- If you insert a key/value pair and then try to look up a key, the implementation has to be able to tell whether two keys are equal.
- Although each language has a different mechanism for specifying this, many languages describe them in similar ways…

Equivalence Relations

“The `equals` method implements an equivalence relation on non-null object references:

- It is *reflexive*: for any non-null reference value `x`, `x.equals(x)` should return true.
- It is *symmetric*: for any non-null reference values `x` and `y`, `x.equals(y)` should return true if and only if `y.equals(x)` returns true.
- It is *transitive*: for any non-null reference values `x`, `y`, and `z`, if `x.equals(y)` returns `true` and `y.equals(z)` returns true, then `x.equals(z)` should return `true`.”

Java 8 Documentation

Equivalence Relations

"The `equals` method implements an equivalence relation on non-null object references:

- It is *reflexive*: for any non-null reference value `x`, `x.equals(x)` should return true.
- It is *symmetric*: for any non-null reference values `x` and `y`, `x.equals(y)` should return true if and only if `y.equals(x)` returns true.
- It is *transitive*: for any non-null reference values `x`, `y`, and `z`, if `x.equals(y)` returns `true` and `y.equals(z)` returns true, then `x.equals(z)` should return `true`."

Java 8 Documentation

Equivalence Relations

"Each unordered associative container is parameterized by `Key`, by a function object type `Hash` that meets the Hash requirements (17.6.3.4) and acts as a hash function for argument values of type `Key`, and by a binary predicate `Pred` that induces an equivalence relation on values of type `Key`. Additionally, `unordered_map` and `unordered_multimap` associate an arbitrary mapped type `T` with the `Key`."

C++14 ISO Spec, §23.2.5/3

Equivalence Relations

"Each unordered associative container is parameterized by `Key`, by a function object type `Hash` that meets the Hash requirements (17.6.3.4) and acts as a hash function for argument values of type `Key`, and by a binary predicate `Pred` that induces an equivalence relation on values of type `Key`. Additionally, `unordered_map` and `unordered_multimap` associate an arbitrary mapped type `T` with the `Key`."

C++14 ISO Spec, §23.2.5/3

Time-Out for Announcements!

Problem Set Two

- The Problem Set Two checkpoint problem was due at 2:30PM today.
  - Have any questions about the problems? Let us know!
- The remaining problems are due on Friday at 2:30PM.
  - Please feel free to stop by office hours with questions!

Problem Set One Solutions

- Problem Set One solutions are now available on the course website.
  - ***Please read over the solutions***. Each problem was chosen for a reason, and it's important to both see one possible solution and the motivation behind the problem.
  - ***Make sure you understand the solutions***. If you don't understand the solutions, please come talk to us and ask us questions. That's how you learn!
- We'll try to get graded problem sets back by Wednesday.

LaTeX Tutorial

- LaTeX is a mathematical typesetting system that's widely used throughout computer science.
  - It's based on TeX, which was invented by our very own Donald Knuth!
- The TAs will be running a tutorial session on how to use LaTeX tomorrow, ***Tuesday October 10th***, from ***4:30PM – 5:30PM*** in ***Gates B03***. Everyone is welcome!
  - This session will also be available on the SCPD website.
- Recommended, but not required. You can type up your solutions however you'd like.

Your Questions

“What’s your favorite book?”

I don’t have a clear favorite, but I do have a ton of recommendations:

“Guns, Germs, and Steel” by Jared Diamond
“The Omnivore’s Dilemma” by Michael Pollan
“Whistling Vivaldi” by Claude Steele
“Stories of Your Life and Others” by Ted Chiang
“Radetsky March” by Joseph Roth
“The Source” by James Michener
“Command and Control” by Eric Schlosser
“Longitude” by Dana Sobel
“The Trial” by Franz Kafka
“Suddenly, a Knock on the Door” by Etgar Keret
“The Prize” by Dale Russakoff”

“Where should freshmen/sophomores look for CS internships/research looking to get their feet wet?”

There's a huge career fair on Wednesday at the Alumni Center from 11:00AM – 4:00PM. It's a great way to get to look around at internships. I'm sorry this is so early in the year! It's really, really dumb that so much internship recruiting goes on then.

For freshmen, look at Facebook U, Explore Microsoft, Google Summer Engineering Practicum, or Pinterest Engage. Those programs are targeted at you!

For research, check out CURIS and look at 500-level CS classes as a way of seeing what the field looks like. And let me know if you want to chat one-on-one! I'm happy to help out.

“What are 3 things on your bucket list?”

1. Visit every US National Park.

2. Go to the moon.

3. Start an airline of all-glass, hypersonic planes that take off when the sun sets and follow the sunset all the way around the earth.

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 binary relation ~ defined over the set ℤ:

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

- Some examples:

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

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

We can binary relations by giving a rule, like this:

a~b* if *some property of a and b holds

This is the general template for defining a relation. Although we're using “if” rather than “iff” here, the two above statements are definitionally equivalent. For a variety of reasons, definitions are often introduced with “if” rather than “iff.” Check the “Mathematical Vocabulary” handout for details.

What properties must $\sim$ 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 $\sim$ is reflexive.

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

Lemma 1: The binary relation ~ is reflexive.

Proof:

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

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

Proof:

What is the formal definition of reflexivity?

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

Lemma 1: The binary relation ~ is reflexive.

Proof:

What is the formal definition of reflexivity?

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

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

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 ~ 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 ~ 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 $a \sim a$.

$a \sim b$ 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.

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

Lemma 1: The binary relation ~ is reflexive.

Proof: Consider an arbitrary $a \in \mathbb{Z}$. We need to prove that $a \sim a$. From the definition of the ~ 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.

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

Lemma 1: The binary relation ~ is reflexive.

Proof: Consider an arbitrary $a \in \mathbb{Z}$. We need to prove that $a \sim a$. From the definition of the ~ 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 1: The binary relation ~ is reflexive.

Proof: Consider an arbitrary $a \in \mathbb{Z}$. We need to prove that $a \sim a$. From the definition of the ~ 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*~*b* if *a*+*b* is even**

Lemma 2: The binary relation ~ is symmetric.

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

Lemma 2: The binary relation $\sim$ is symmetric.

Proof:

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

Lemma 2: The binary relation ~ is symmetric.

Proof:

What is the formal definition of symmetry?

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

Lemma 2: The binary relation ~ is symmetric.

Proof:

What is the formal definition of symmetry?

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

$a \sim b$ **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 $b \sim a$.

$a \sim b$ 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 $b \sim a$.

$a \sim b$ 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 $b \sim a$.

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

Lemma 2: The binary relation ~ is symmetric.

Proof: Consider any integers a and b where $a \sim b$. We need to show that $b \sim a$.

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

Lemma 2: The binary relation $\sim$ is symmetric.

Proof: Consider any integers a and b where $a \sim b$.
We need to show that $b \sim a$.

Since $a \sim b$, we know that $a+b$ is even.

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

Lemma 2: The binary relation $\sim$ is symmetric.

Proof: Consider any integers a and b where $a \sim b$. We need to show that $b \sim a$.

Since $a \sim b$, we know that $a+b$ is even. Because $a+b = b+a$, this means that $b+a$ is even.

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

Lemma 2: The binary relation $\sim$ is symmetric.

Proof: Consider any integers a and b where $a \sim b$. We need to show that $b \sim a$.

Since $a \sim b$, 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 $b \sim a$, as required.

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

Lemma 2: The binary relation ~ is symmetric.

Proof: Consider any integers a and b where $a \sim b$. We need to show that $b \sim a$.

Since $a \sim b$, 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 $b \sim a$, as required. ■

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

Lemma 3: The binary relation $\sim$ is transitive.

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

Lemma 3: The binary relation ~ is transitive.

Proof:

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

Lemma 3: The binary relation ~ is transitive.

Proof:

What is the formal definition of transitivity?

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

Lemma 3: The binary relation ~ is transitive.

Proof:

What is the formal definition of transitivity?

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

$\boldsymbol{a \sim b}$ **if** $\boldsymbol{a+b}$ **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 $a \sim c$.

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

Lemma 3: The binary relation $\sim$ is transitive.

Proof: Consider arbitrary integers a, b and c where $a \sim b$ and $b \sim c$.

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

Lemma 3: The binary relation $\sim$ 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.

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

Lemma 3: The binary relation $\sim$ 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 $a+b$ and $b+c$ are even.

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

Lemma 3: The binary relation $\sim$ 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 $a+b$ and $b+c$ are even. This means there are integers k and m where $a+b = 2k$ and $b+c = 2m$.

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

Lemma 3: The binary relation $\sim$ 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 $a+b$ and $b+c$ are even. This means there are integers k and m where $a+b = 2k$ and $b+c = 2m$. Notice that

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

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

Lemma 3: The binary relation $\sim$ 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 $a+b$ and $b+c$ 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

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

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

Lemma 3: The binary relation $\sim$ 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 $a+b$ and $b+c$ 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

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

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

Lemma 3: The binary relation $\sim$ 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 $a+b$ and $b+c$ 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.

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

Lemma 3: The binary relation $\sim$ 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 $a+b$ and $b+c$ 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. ■

$a \sim b$ 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 $a+b$ and $b+c$ 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. ■

Notice that these are grammatically complete sentences. In your own proofs, make sure to write in complete sentences and use appropriate punctuation. It looks really classy and makes your proofs easier to read.

Try the "mugga mugga test." If you read a proof and replace mathematical symbols with "mugga mugga," it should still be grammatically correct.

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

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. ■

An Observation

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

Lemma 1: The binary relation ~ is reflexive.

Proof: Consider an arbitrary $a \in \mathbb{Z}$. We need to prove that $a \sim a$. From the definition of the ~ 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. ■

The formal definition of reflexivity is given in first-order logic, but this proof does not contain any first-order logic symbols!

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

Lemma 2: The binary relation ~ is symmetric.

Proof: Consider any integers a and b where $a \sim b$. We need to show that $b \sim a$.

Since $a \sim b$, 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 $b \sim a$, as required. ■

The formal definition of symmetry is given in first-order logic, but this proof does not contain any first-order logic symbols!

$a \sim b$ 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 $a+b$ and $b+c$ 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

$$a+c = 2k + 2$$

So there is an integer r, n
$a+c = 2r$. Thus $a+c$ is eve

The formal definition of transitivity is given in first-order logic, but this proof does not contain any first-order logic symbols!

First-Order Logic and Proofs

- First-order logic is an excellent tool for giving formal definitions to key terms.
- While first-order logic *guides* the structure of proofs, it is *exceedingly rare* to see first-order logic in written proofs.
- Follow the example of these proofs:
  - Use the FOL definitions to determine what to assume and what to prove.
  - Write the proof in plain English using the conventions we set up in the first week of the class.
- ***Please, please, please, please, please internalize the contents of this slide!***