Binary Relations

Part II

Outline for Today

- ***Recap from Last Time***
  - Where are we, again?
- ***Properties of Equivalence Relations***
  - What's so special about those three rules?
- ***Strict Orders***
  - A different type of mathematical structure
- ***Hasse Diagrams***
  - How to visualize rankings

Recap from Last Time

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 $\boldsymbol{a \not R b}$.

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

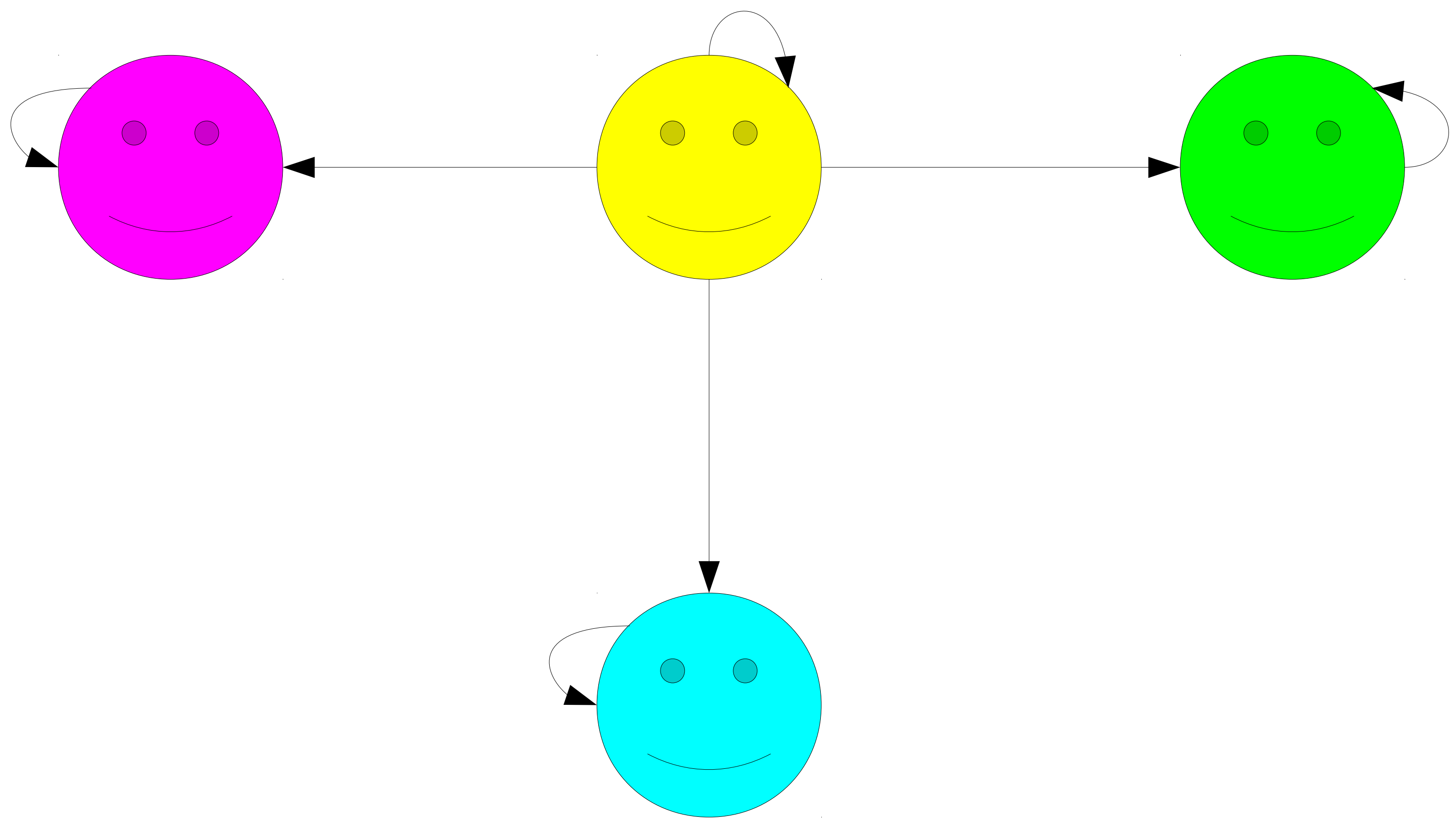
Reflexivity

- 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 logic statement is true about R:

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

("Every element is related to itself.")

Reflexivity Visualized



$\forall a \in A.\ aRa$

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

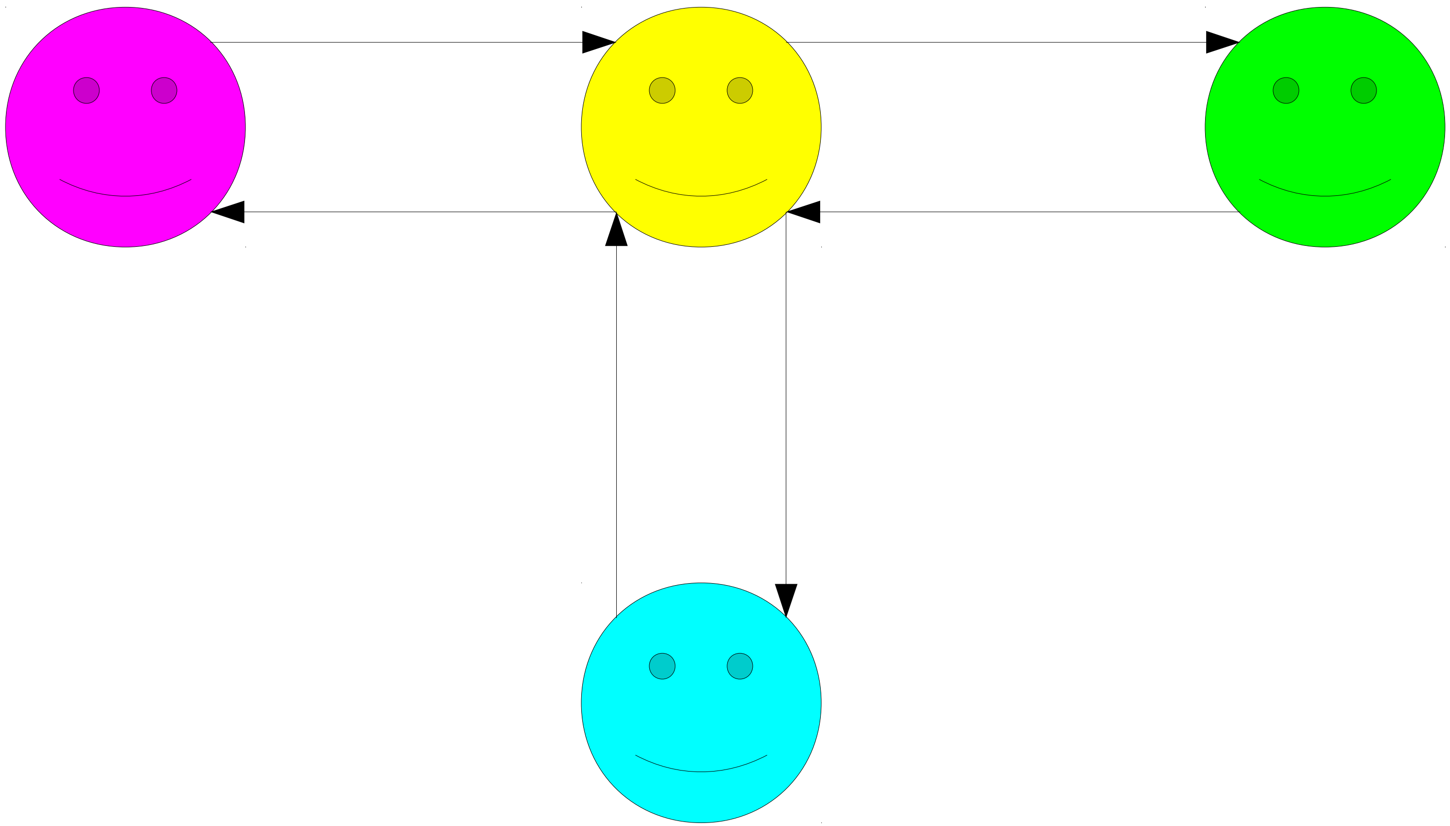
Symmetry

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

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

(*"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.”*)

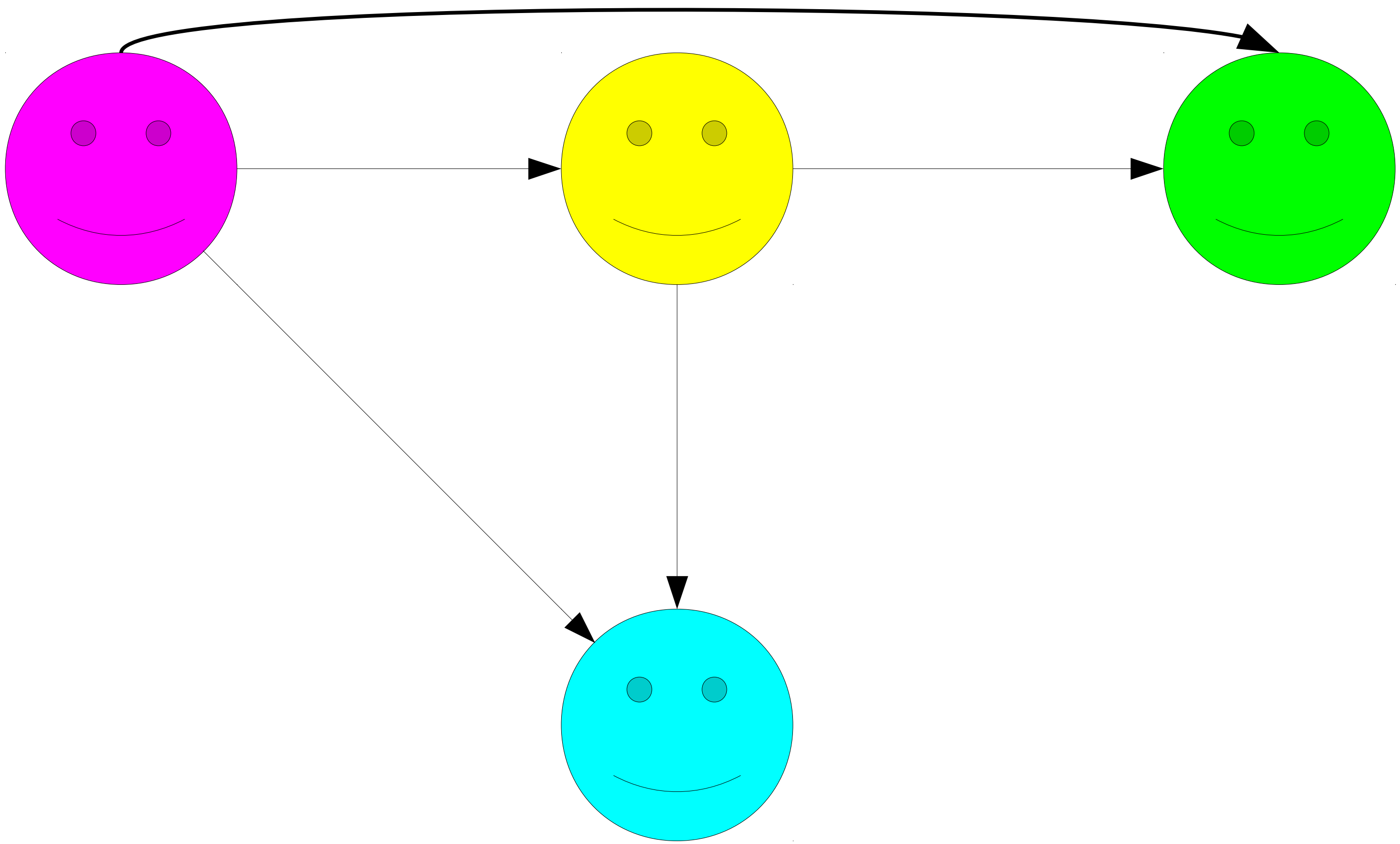
Transitivity

- 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 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

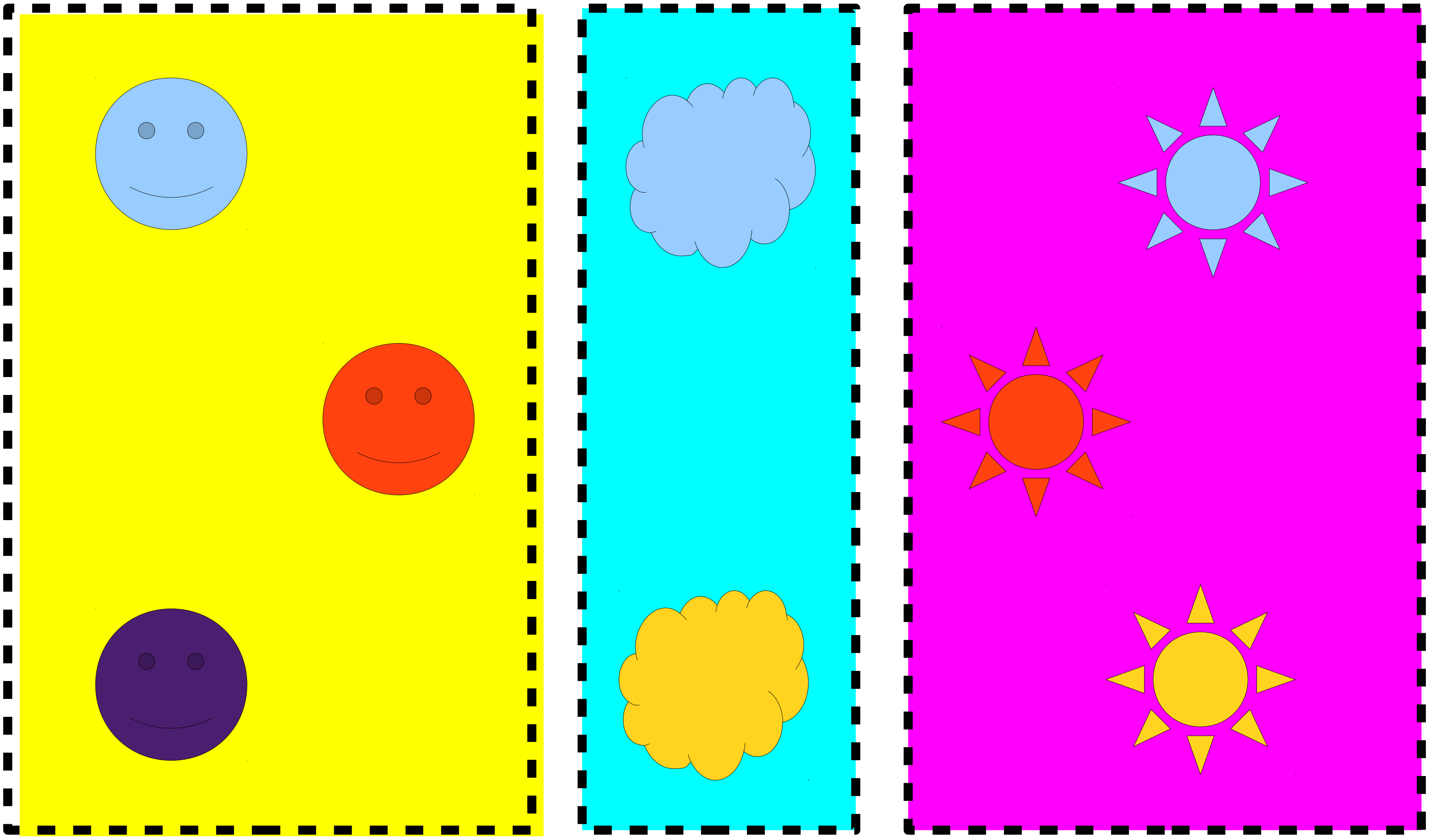


$$\forall a \in A. \forall b \in A. \forall c \in A. (aRb \land bRc \to aRc)$$

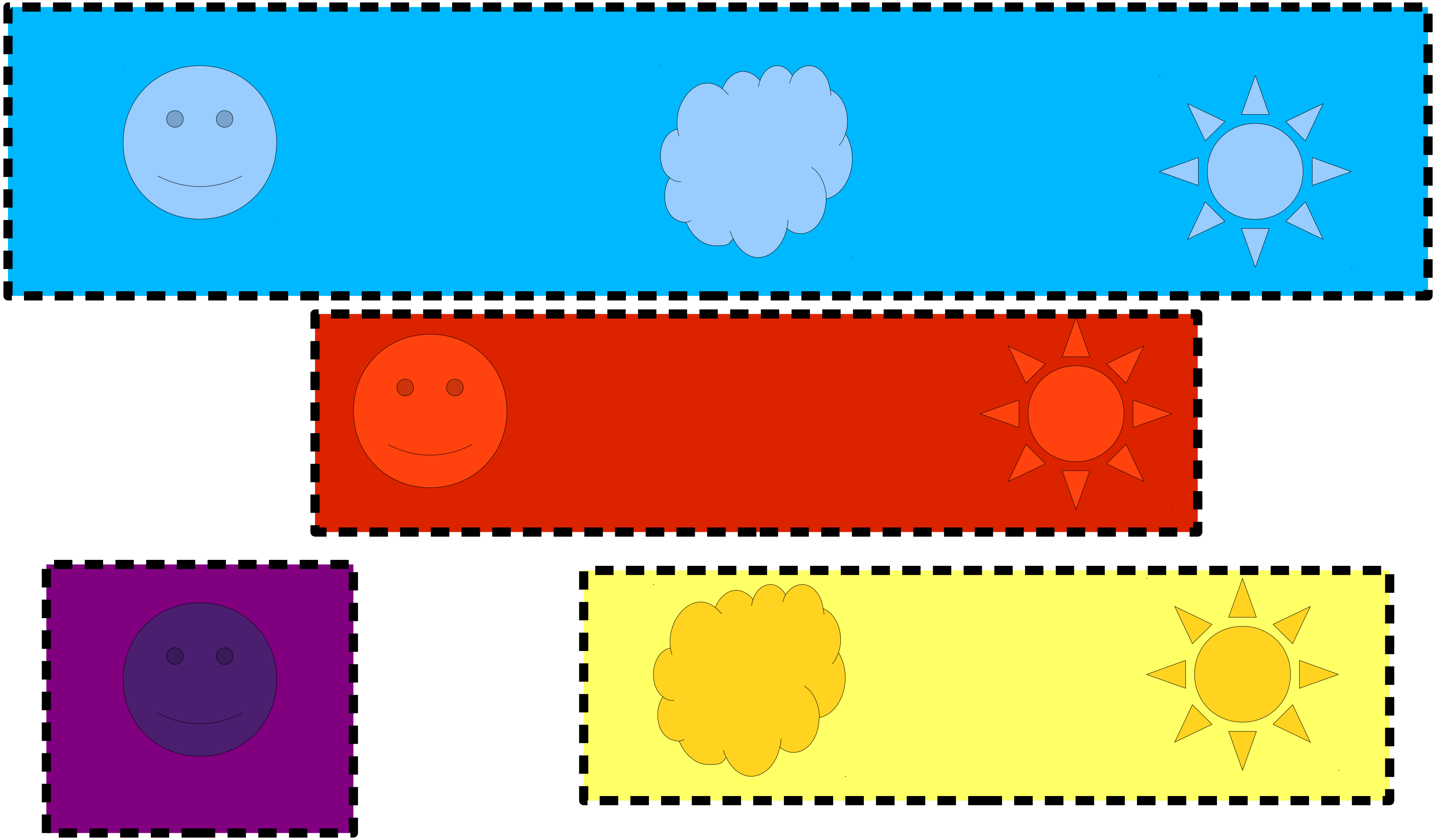
("Whenever a is related to b and b is related to c, we know a is related to c.)

New Stuff!

Properties of Equivalence Relations



xRy if x and y have the same shape



xTy if x is the same **color** as y

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.
- For example, consider the $\equiv_3$ relation over $\mathbb{N}$. Then
  - $[0]_{\equiv_3} = \{0, 3, 6, 9, 12, 15, 18, \ldots\}$
  - $[1]_{\equiv_3} = \{1, 4, 7, 10, 13, 16, 19, \ldots\}$
  - $[2]_{\equiv_3} = \{2, 5, 8, 11, 14, 17, 20, \ldots\}$
  - $[3]_{\equiv_3} = \{0, 3, 6, 9, 12, 15, 18, \ldots\}$

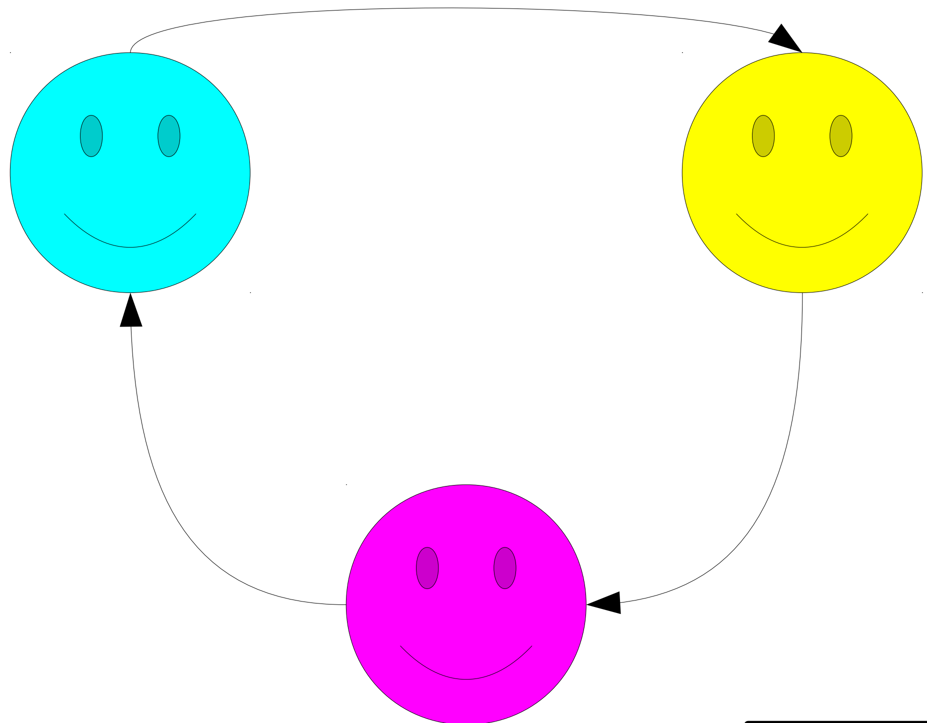
Notice that $[0]_{\equiv_3} = [3]_{\equiv_3}$. These are *literally* the same set, so they're just different names for the same thing.

The Fundamental Theorem of Equivalence Relations: Let R be an equivalence relation over a set A. Then every element $a \in A$ belongs to exactly one equivalence class of R.

How’d We Get Here?

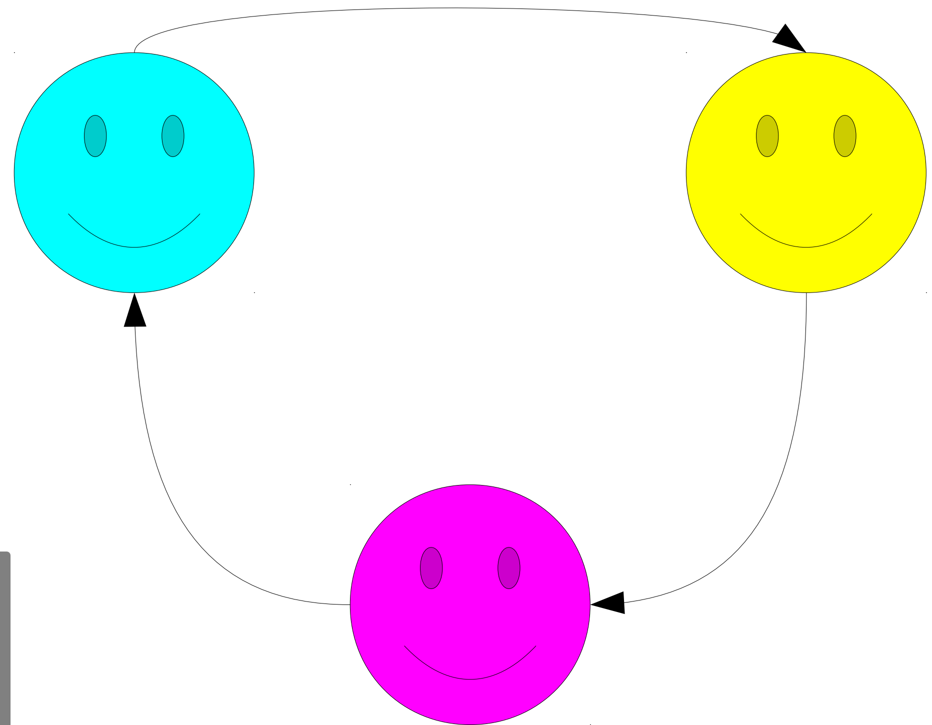
- We discovered equivalence relations by thinking about ***partitions*** of a set of elements.
- We saw that if we had a binary relation that tells us whether two elements are in the same group, it had to be reflexive, symmetric, and transitive.
- The FToER says that, in some sense, these rules precisely capture what it means to be a partition.
- ***Question:*** What’s so special about these three rules?

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



A binary relation with this property is called ***cyclic***.

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



Is this an equivalence relation?

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

Theorem: A binary relation R over a set A is an equivalence relation if and only if it is reflexive and cyclic.

Lemma 1: If R is an equivalence relation over a set A, then R is reflexive and cyclic.

Lemma 2: If R is a binary relation over a set A that is reflexive and cyclic, then R is an equivalence relation.

Lemma 1: If R is an equivalence relation over a set A, then R is reflexive and cyclic.

What We're Assuming

- R is an equivalence relation.
  - R is reflexive.
  - R is symmetric.
  - R is transitive.

What We Need To Show

- R is reflexive.
- R is cyclic.

Lemma 1: If R is an equivalence relation over a set A, then R is reflexive and cyclic.

What We're Assuming

R is an equivalence relation.

- R is reflexive.

  R is symmetric.

  R is transitive.

What We Need To Show

- R is reflexive.

  R is cyclic.

Lemma 1: If R is an equivalence relation over a set A, then R is reflexive and cyclic.

What We're Assuming

- R is an equivalence relation.
  - R is reflexive.
  - R is symmetric.
  - R is transitive.

What We Need To Show

R is reflexive.

R is cyclic.

- If aRb and bRc, then cRa.

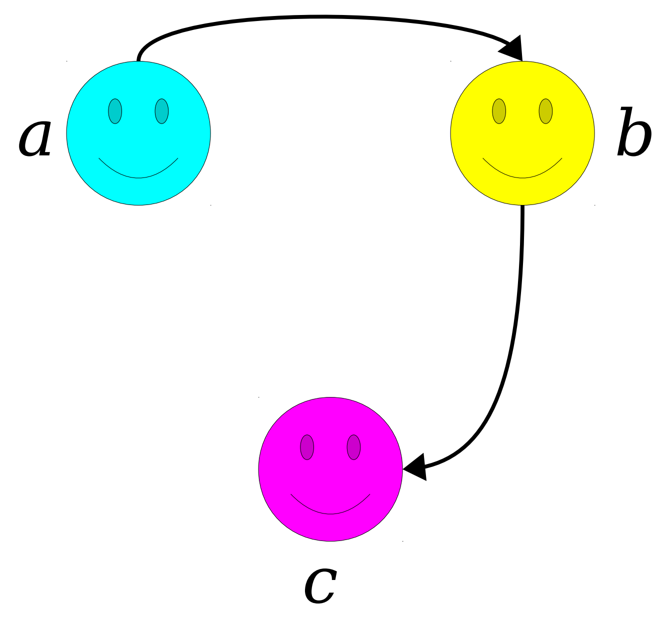
Lemma 1: If R is an equivalence relation over a set A, then R is reflexive and cyclic.

What We're Assuming

- R is an equivalence relation.
  - R is reflexive.
  - R is symmetric.
  - R is transitive.

What We Need To Show

- If aRb and bRc, then cRa.



Lemma 1: If R is an equivalence relation over a set A, then R is reflexive and cyclic.

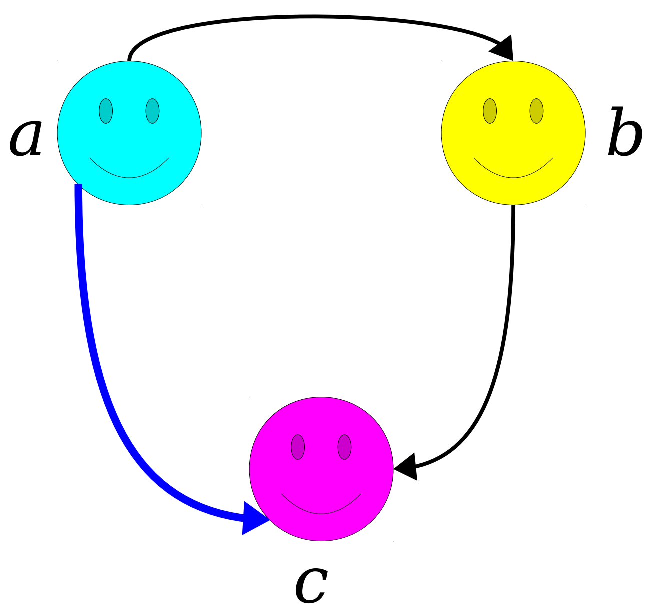
What We're Assuming

R is an equivalence relation.

- R is reflexive.
- R is symmetric.
- R is transitive.

What We Need To Show

- If aRb and bRc, then cRa.



Lemma 1: If R is an equivalence relation over a set A, then R is reflexive and cyclic.

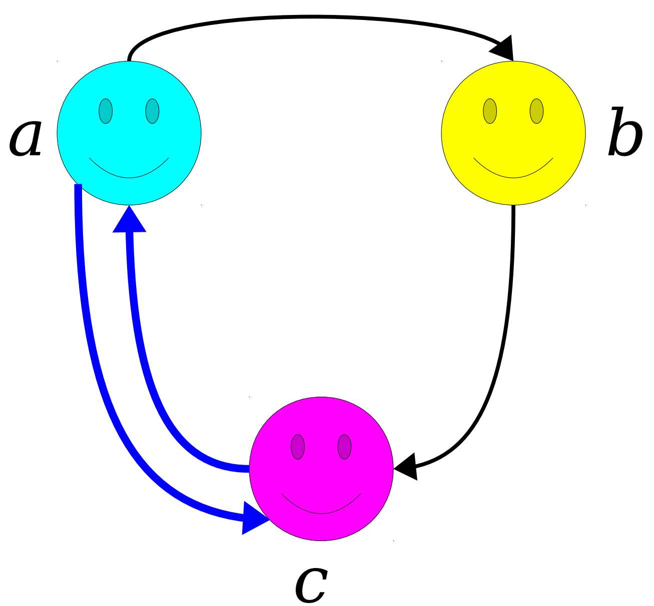
What We're Assuming

R is an equivalence relation.

- R is reflexive.
- R is symmetric.
- R is transitive.

What We Need To Show

- If aRb and bRc, then cRa.



Lemma 1: If R is an equivalence relation over a set A, then R is reflexive and cyclic.

Proof: Let R be an arbitrary equivalence relation over some set A. We need to prove that R is reflexive and cyclic.

Since R is an equivalence relation, we know that R is reflexive, symmetric, and transitive. Consequently, we already know that R is reflexive, so we only need to show that R is cyclic.

To prove that R is cyclic, consider any arbitrary $a, b, c \in A$ where aRb and bRc. We need to prove that cRa holds. Since R is transitive, from aRb and bRc we see that aRc. Then, since R is symmetric, from aRc we see that cRa, which is what we needed to prove. ■

Lemma 1: If R is an equivalence relation over a set A, then R is reflexive and cyclic.

Proof: Let R be an arbitrary equivalence relation over some set A. We need to prove that R is reflexive and cyclic.

Since R is an equivalence relation, we know that R is reflexive, symmet
already know that
that R is cyclic.

Notice how the first few sentences of this proof mirror the structure of what needs to be proved. We're just following the templates from the first week of class!

To prove that R is … A
where aRb and bR
Since R is transitive, from aRb and bRc we see that aRc. Then, since R is symmetric, from aRc we see that cRa, which is what we needed to prove. ■

Notice how this setup mirrors the first-order definition of cyclicity:

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

When writing proofs about terms with first-order definitions, it's critical to call back to those definitions!

Lemma … en R is refl…

Proof: L… me set A.

Since … reflexi… alread… ow that R is cyclic.

To prove that R is cyclic, consider any arbitrary a, b, $c \in A$ where aRb and bRc. We need to prove that cRa holds.

Since R is transitive, from aRb and bRc we see that aRc. Then, since R is symmetric, from aRc we see that cRa, which is what we needed to prove. ■

Although this proof is deeply informed by the first-order definitions, notice that there is no first-order logic notation anywhere in the proof. That's normal – it's actually quite rare to see first-order logic in written proofs.

Lem ...en R is ...

Proof: Let R be an arbitrary equivalence relation over some set A. We need to prove that R is reflexive and cyclic.

Since R is an equivalence relation, we know that R is reflexive, symmetric, and transitive. Consequently, we already know that R is reflexive, so we only need to show that R is cyclic.

To prove that R is cyclic, consider any arbitrary $a, b, c \in A$ where aRb and bRc. We need to prove that cRa holds. Since R is transitive, from aRb and bRc we see that aRc. Then, since R is symmetric, from aRc we see that cRa, which is what we needed to prove. ■

Lemma 2: If R is a binary relation over a set A that is reflexive and cyclic, then R is an equivalence relation.

What We're Assuming

- R is reflexive.
- R is cyclic.

What We Need To Show

- R is an equivalence relation.
  - R is reflexive.
  - R is symmetric.
  - R is transitive.

Lemma 2: If R is a binary relation over a set A that is reflexive and cyclic, then R is an equivalence relation.

What We're Assuming

- R is reflexive.

R is cyclic.

What We Need To Show

R is an equivalence relation.

- R is reflexive.

R is symmetric.

R is transitive.

Lemma 2: If R is a binary relation over a set A that is reflexive and cyclic, then R is an equivalence relation.

What We're Assuming

- R is reflexive.
- R is cyclic.

What We Need To Show

- R is an equivalence relation.
  - R is reflexive.
  - R is symmetric.
  - R is transitive.

Lemma 2: If R is a binary relation over a set A that is reflexive and cyclic, then R is an equivalence relation.

What We're Assuming

- R is reflexive.
- R is cyclic.

What We Need To Show

R is an equivalence relation.

- R is reflexive.
- R is symmetric.
- R is transitive.

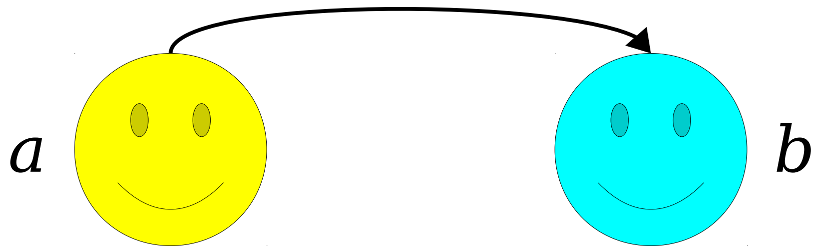
Lemma 2: If R is a binary relation over a set A that is reflexive and cyclic, then R is an equivalence relation.

What We're Assuming

- R is reflexive.
- R is cyclic.

What We Need To Show

- R is symmetric.
  - If aRb, then bRa.



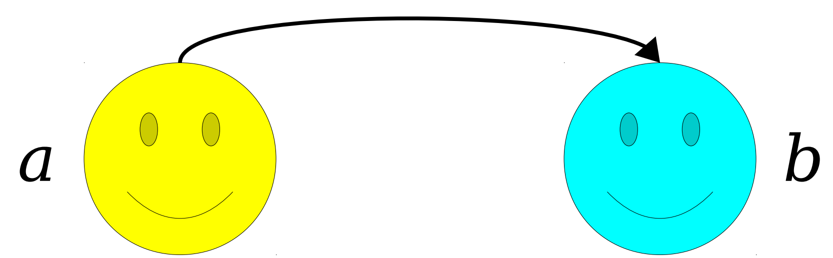
Lemma 2: If R is a binary relation over a set A that is reflexive and cyclic, then R is an equivalence relation.

What We're Assuming

- R is reflexive.
  - $\forall x \in A.\ xRx$
- R is cyclic.
  - $xRy \land yRz \to zRx$

What We Need To Show

- R is symmetric.
  - If aRb, then bRa.



Lemma 2: If R is a binary relation over a set A that is reflexive and cyclic, then R is an equivalence relation.

What We're Assuming

R is reflexive.

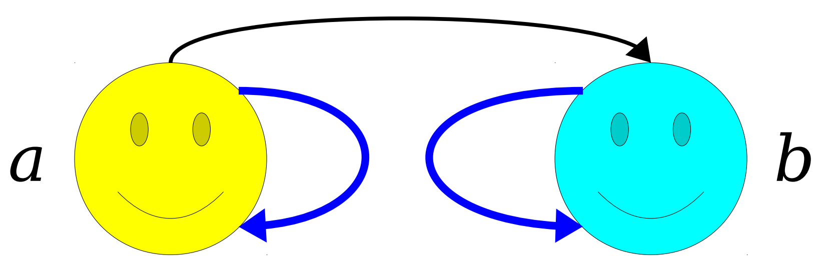
- $\forall x \in A.\ xRx$

R is cyclic.

$xRy \land yRz \rightarrow zRx$

What We Need To Show

- R is symmetric.
  - If aRb, then bRa.



Lemma 2: If R is a binary relation over a set A that is reflexive and cyclic, then R is an equivalence relation.

What We're Assuming

R is reflexive.

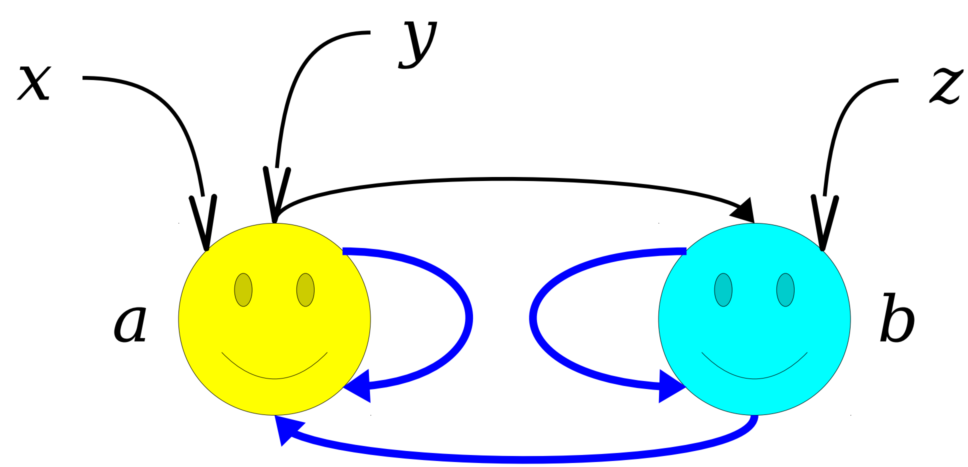
$\forall x \in A.\ xRx$

R is cyclic.

- $xRy \land yRz \to zRx$

What We Need To Show

- R is symmetric.
  - If aRb, then bRa.



Lemma 2: If R is a binary relation over a set A that is reflexive and cyclic, then R is an equivalence relation.

What We're Assuming

- R is reflexive.
  - $\forall x \in A.\ xRx$
- R is cyclic.
  - $xRy \land yRz \to zRx$

What We Need To Show

R is an equivalence relation.

R is reflexive.

R is symmetric.

- R is transitive.

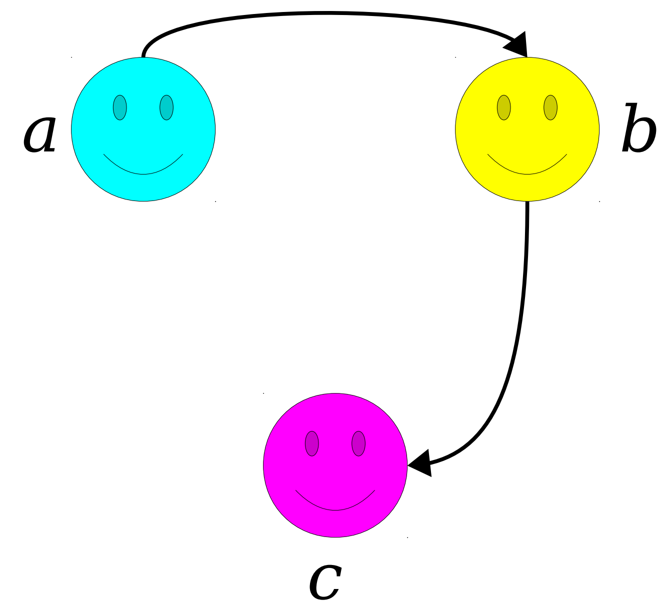
Lemma 2: If R is a binary relation over a set A that is reflexive and cyclic, then R is an equivalence relation.

What We're Assuming

- R is reflexive.
  - $\forall x \in A.\ xRx$
- R is cyclic.
  - $xRy \land yRz \to zRx$

What We Need To Show

- R is transitive.
  - If aRb and bRc, then aRc.



Lemma 2: If R is a binary relation over a set A that is reflexive and cyclic, then R is an equivalence relation.

What We're Assuming

R is reflexive.

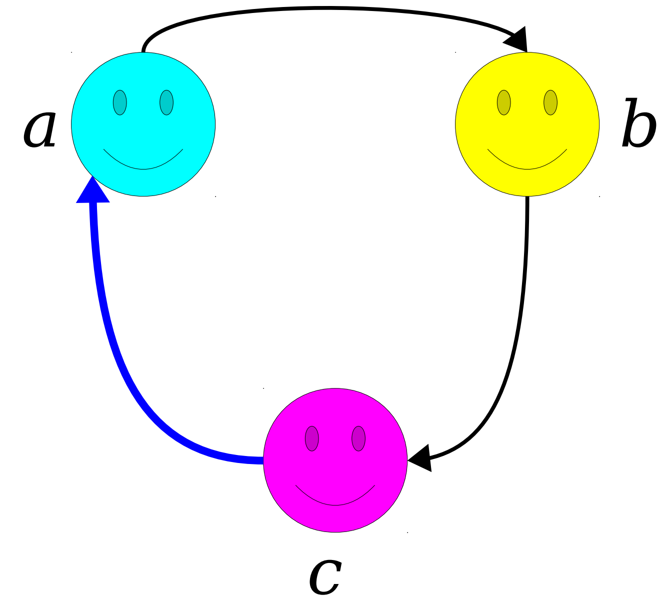
$\forall x \in A.\ xRx$

R is cyclic.

$xRy \land yRz \to zRx$

What We Need To Show

- R is transitive.
  - If aRb and bRc, then aRc.



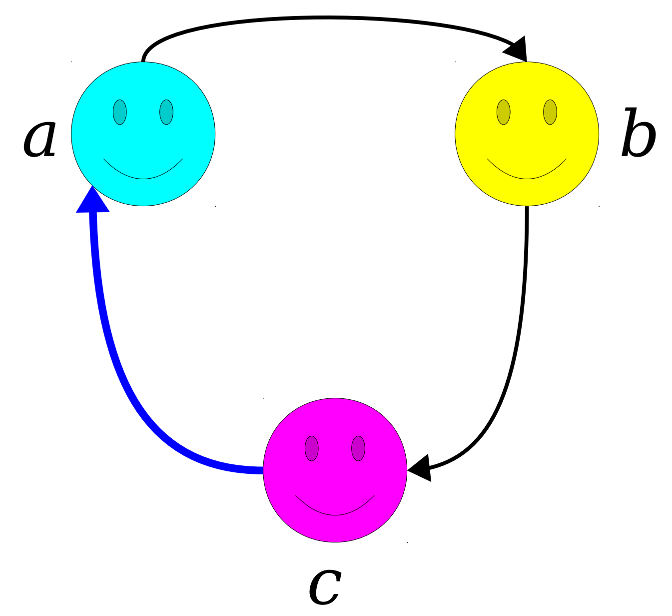
Lemma 2: If R is a binary relation over a set A that is reflexive and cyclic, then R is an equivalence relation.

What We're Assuming

- R is reflexive.
  - $\forall x \in A.\ xRx$
- R is cyclic.
  - $xRy \land yRz \to zRx$

What We Need To Show

- R is transitive.
  - If aRb and bRc, then aRc.



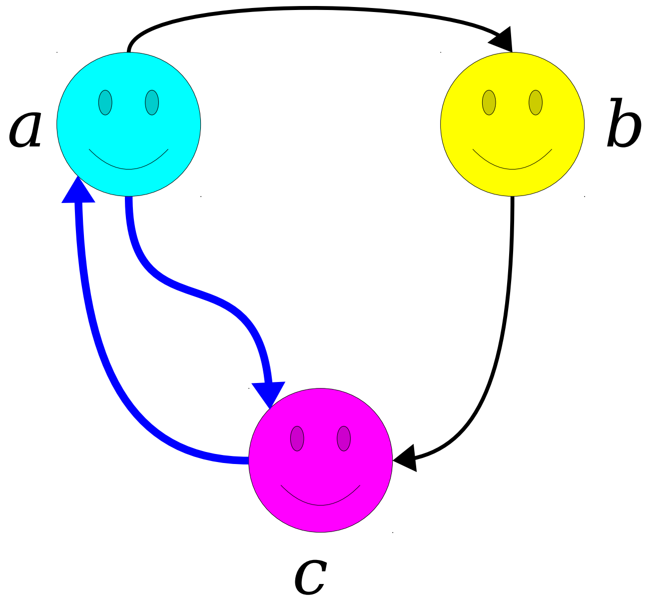
Lemma 2: If R is a binary relation over a set A that is reflexive and cyclic, then R is an equivalence relation.

What We're Assuming

- R is reflexive.
  - $\forall x \in A.\ xRx$
- R is cyclic.
  - $xRy \land yRz \to zRx$
- R is symmetric
  - $xRy \to yRx$

What We Need To Show

- R is transitive.
  - If aRb and bRc, then aRc.



Lemma 2: If R is a binary relation over a set A that is cyclic and reflexive, then R is an equivalence relation.

Proof: Let R be an arbitrary binary relation over a set A that is cyclic and reflexive. We need to prove that R is an equivalence relation. To do so, we need to show that R is reflexive, symmetric, and transitive. Since we already know by assumption that R is reflexive, we just need to show that R is symmetric and transitive.

First, we'll prove that R is symmetric. To do so, pick any arbitrary $a, b \in A$ where aRb holds. We need to prove that bRa is true. Since R is reflexive, we know that aRa holds. Therefore, by cyclicity, since aRa and aRb, we learn that bRa, as required.

Next, we'll prove that R is transitive. Let a, b, and c be any elements of A where aRb and bRc. We need to prove that aRc. Since R is cyclic, from aRb and bRc we see that cRa. Earlier, we showed that R is symmetric. Therefore, from cRa we see that aRc is true, as required. ■

Lemm clic
and
Proof that
is c
equ R is
ref
kno o
sho

Notice how this setup mirrors the first-order definition of symmetry:

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

When writing proofs about terms with first-order definitions, it's critical to call back to those definitions!

First, we'll prove that R is symmetric. To do so, pick any arbitrary $a, b \in A$ where aRb holds. We need to prove that bRa is true. Since R is reflexive, we know that aRa holds. Therefore, by cyclicity, since aRa and aRb, we learn that bRa, as required.

Next, we'll prove that R is transitive. Let a, b, and c be any elements of A where aRb and bRc. We need to prove that aRc. Since R is cyclic, from aRb and bRc we see that cRa. Earlier, we showed that R is symmetric. Therefore, from cRa we see that aRc is true, as required. ■

Lemma 2: If R is a binary relation over a set A that is cyclic and reflexive, then R is an equivalence relation.

Proof: Let R be an arbitrary binary relation over a set A that is cyclic and reflexive. We need to prove that R is an equi… is refle… know… show…

Notice how this setup mirrors the first-order definition of transitivity:

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

When writing proofs about terms with first-order definitions, it's critical to call back to those definitions!

First… y arbit… at bRa … s. Ther… t bRa, as required.

Next, we'll prove that R is transitive. Let a, b, and c be any elements of A where aRb and bRc. We need to prove that aRc. Since R is cyclic, from aRb and bRc we see that cRa. Earlier, we showed that R is symmetric. Therefore, from cRa we see that aRc is true, as required. ■

Lemma 2. ...
and refl...

Although this proof is deeply informed by the first-order definitions, notice that there is no first-order logic notation anywhere in the proof. That's normal – it's actually quite rare to see first-order logic in written proofs.

Proof: Let ...
is cyclic ...
equivalence relation. To do so, we need to show that R is reflexive, symmetric, and transitive. Since we already know by assumption that R is reflexive, we just need to show that R is symmetric and transitive.

First, we'll prove that R is symmetric. To do so, pick any arbitrary $a, b \in A$ where aRb holds. We need to prove that bRa is true. Since R is reflexive, we know that aRa holds. Therefore, by cyclicity, since aRa and aRb, we learn that bRa, as required.

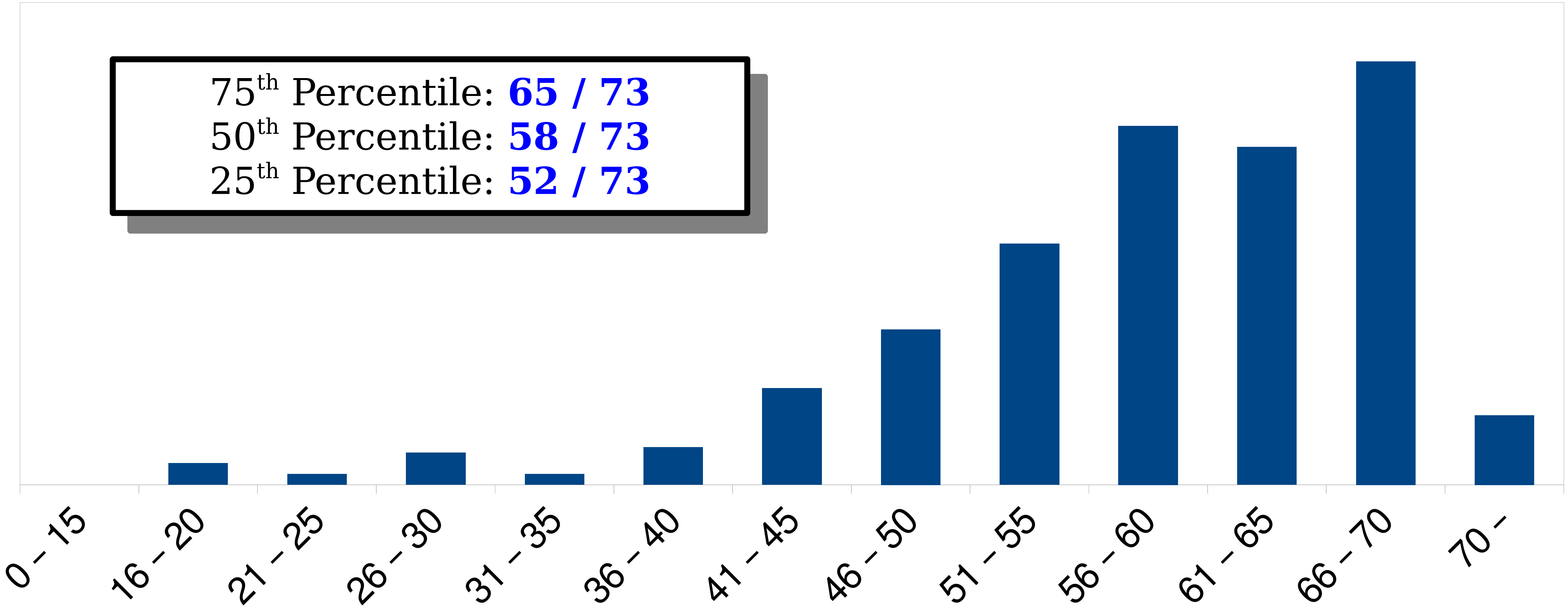
Next, we'll prove that R is transitive. Let a, b, and c be any elements of A where aRb and bRc. We need to prove that aRc. Since R is cyclic, from aRb and bRc we see that cRa. Earlier, we showed that R is symmetric. Therefore, from cRa we see that aRc is true, as required. ■

Refining Your Proofwriting

- When writing proofs about terms with formal definitions, you ***must*** call back to those definitions.
  - Use the first-order definition to see what you'll assume and what you'll need to prove.
- When writing proofs about terms with formal definitions, you ***must not*** include any first-order logic in your proofs.
  - Although you won't use any FOL *notation* in your proofs, your proof implicitly calls back to the FOL definitions.
- You'll get a lot of practice with this on Problem Set Three. If you have any questions about how to do this properly, please feel free to ask on Piazza or stop by office hours!

Time-Out for Announcements!

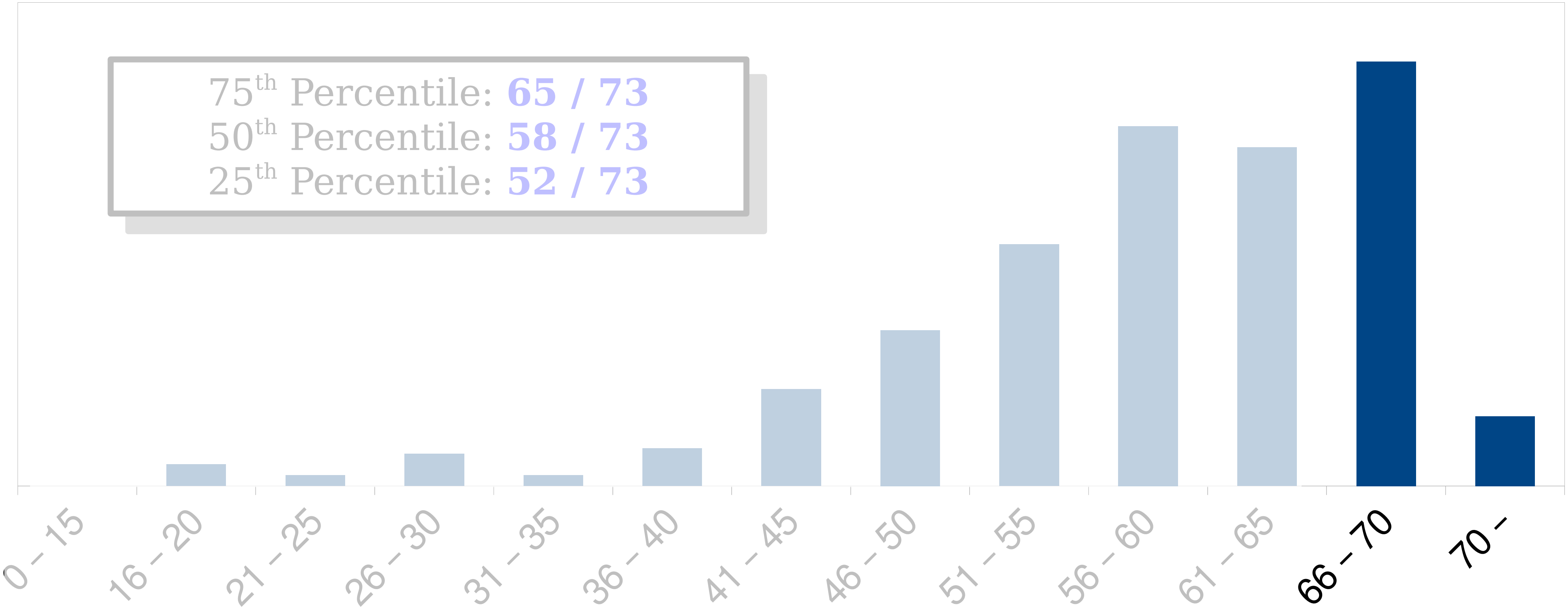
Problem Set One Graded



Pro tips when seeing a grading curve:

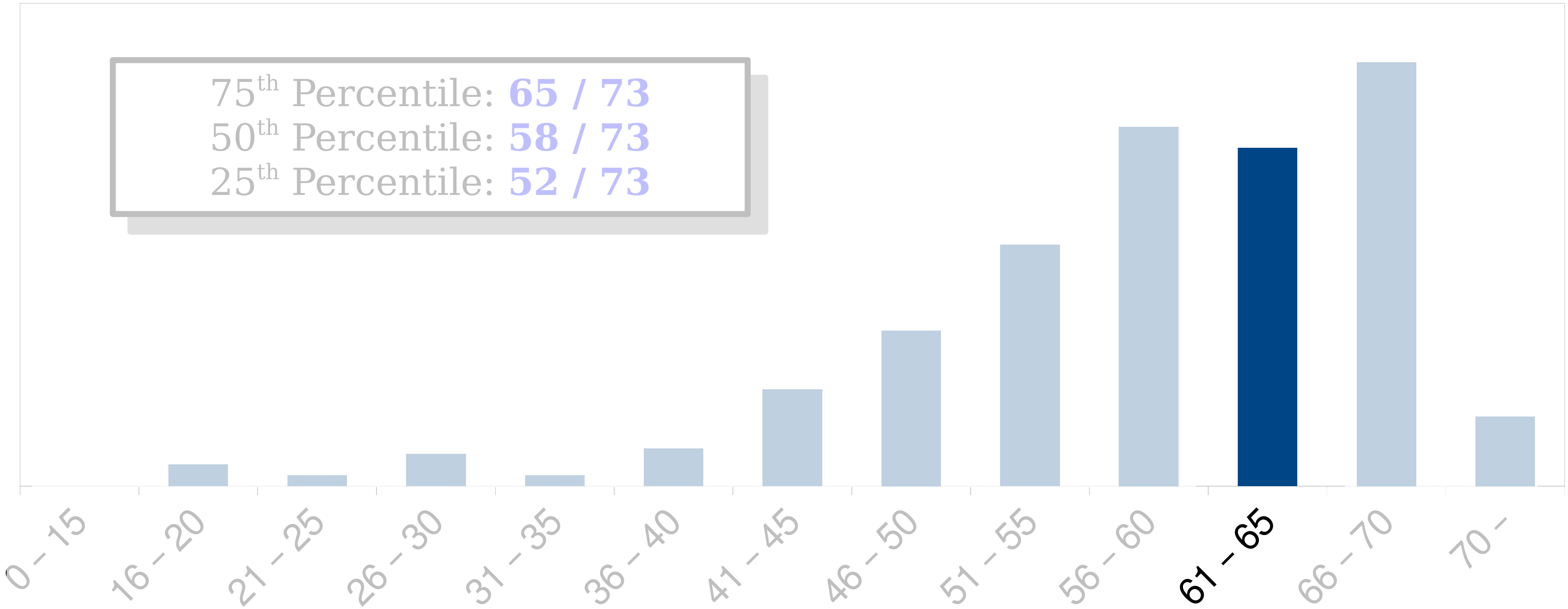
1. Standard deviations are ***malicious lies***. Ignore them.
2. The average score is a ***malicious lie***. Ignore it.
3. Raw scores are ***malicious lies***. Ignore them.

Problem Set One Graded



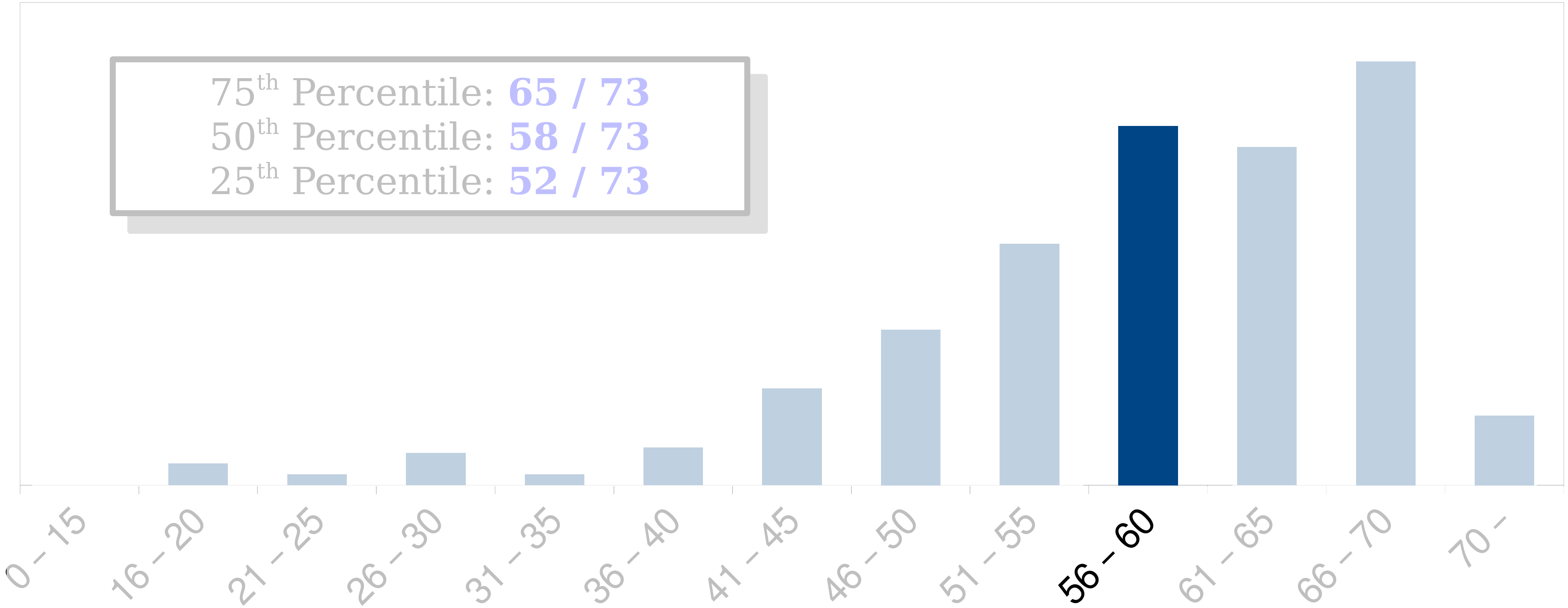
"Great job! Look over your feedback for some tips on how to tweak things for next time."

Problem Set One Graded



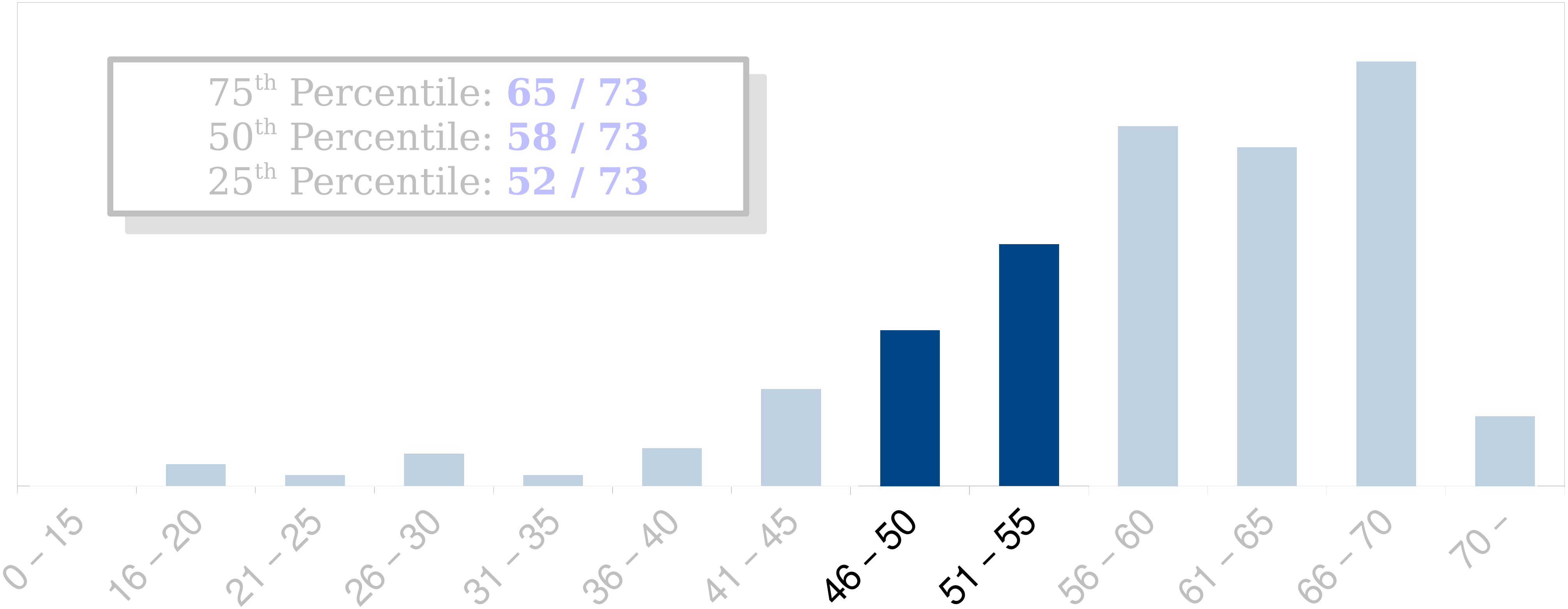
"You're almost there! Review the feedback on your submission and see if there's anything to focus on for next time."

Problem Set One Graded



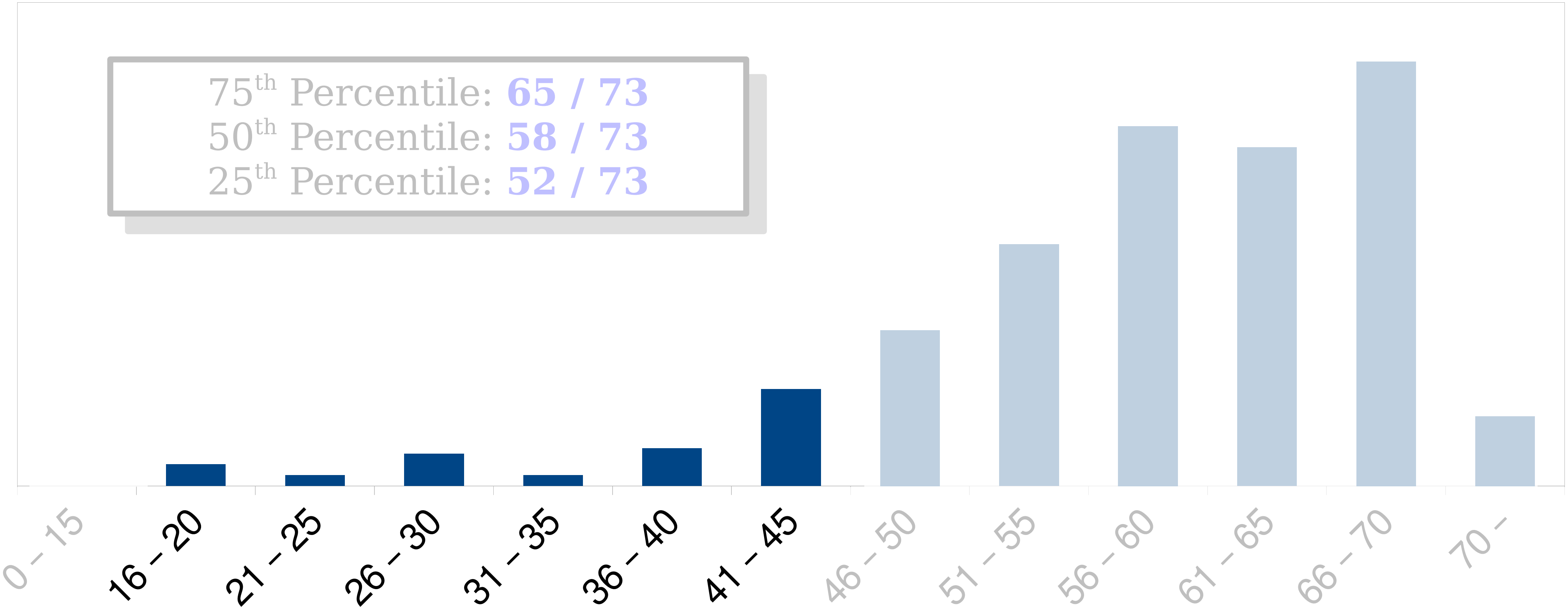
"You're on the right track, but there are some areas where you need to improve. Review your feedback and ask us questions about how to improve."

Problem Set One Graded



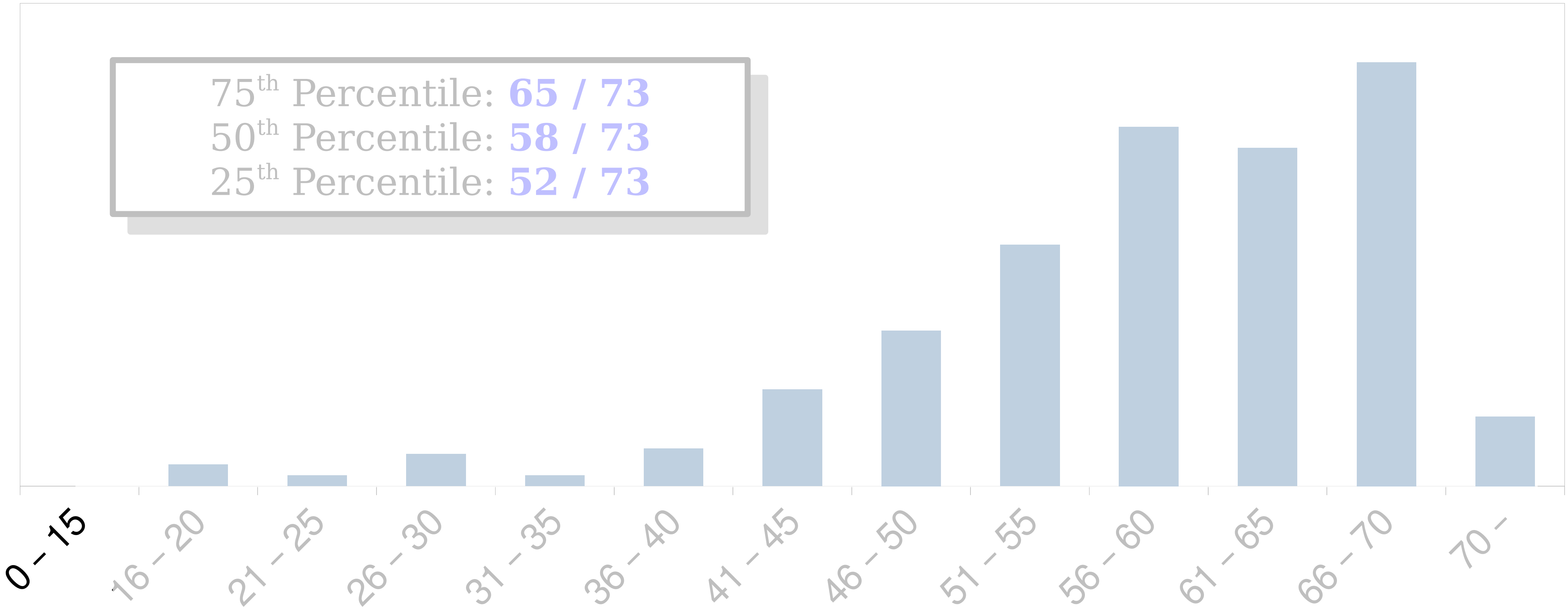
"You're not quite there yet, but don't worry! Review your feedback in depth and find some concrete areas where you can improve. Ask us questions, focus on your weak spots, and you'll be in great shape."

Problem Set One Graded



"Looks like something hasn't quite clicked yet. Get in touch with us and stop by office hours to get some extra feedback and advice. Don't get discouraged – you can do this!"

Problem Set One Graded



"Oops, you forgot to submit."

What Not to Think

- "Well, I guess I'm just not good at math."
  - For most of you, this is your first time doing any rigorous proof-based math.
  - Don't judge your future performance based on a single data point.
  - Life advice: avoid "***minivanning***."
  - Life advice: have a growth mindset!
- "Hey, I did above the median. That's good enough."
  - Unless you literally earned every single point on this problem set – and even in that case – there's some area where the course staff thinks you need to improve. ***Take the time to see what that is***.

THE ROAD TO WISDOM

The road to wisdom?—Well, it's plain
and simple to express:
Err
and err
and err again,
but less
and less
and less.

— Piet Hein

And this guy is interesting. You should look him up.

CS legend Don Knuth has this poem on the wall of his house.

Problem Set Two

- Problem Set Two is due on Friday at 2:30.
    - Have questions? Stop by office hours or ask on Piazza!
- Reminder: check your first-order logic translations using our handy checklist! It's up on the course website.

Your Questions

"What are the pros and cons for a masters in CS vs. a B.S. in CS?"

I think there are two main contexts in which you could ask this question. First, should you do a CS major (BS), or do something else and get a coterm in CS (MS)? Second, if you already have a CS undergrad (BS), should you then go on to do a master's in it (MS)?

For that first case: both a CS BS and a CS MS will teach you a ton about the field. The CS BS gives you more of an opportunity to explore the field, since the tracks are more flexible, and the CS MS goes into way more depth into a single area but has a bit less exploration built in. The biggest question, though, would be what other major you're looking at. If you have multiple interests, a major outside of CS and a coterm in CS can be a great option. Just don't do that because you're nervous about majoring in CS; if you want to do CS but feel a bit intimidated, please come talk to me!

For that second case: there's a slight salary difference between BS and MS grads, but the opportunity cost of giving up a year's salary usually will eat it. The main reason to do an MS is if you're really liking what you're learning and want to dive deeper into it.

“What are your thoughts on AI?”

It's really exciting, there's a ton of new cool technologies coming out now, and we're making a lot of progress in areas like vision and translation that have historically been real sticking points.

I also think that there's a bit too much hype and that while we are making huge steps forward, there's still a lot to figure out and in most domains simple heuristics are “good enough” for our purposes.

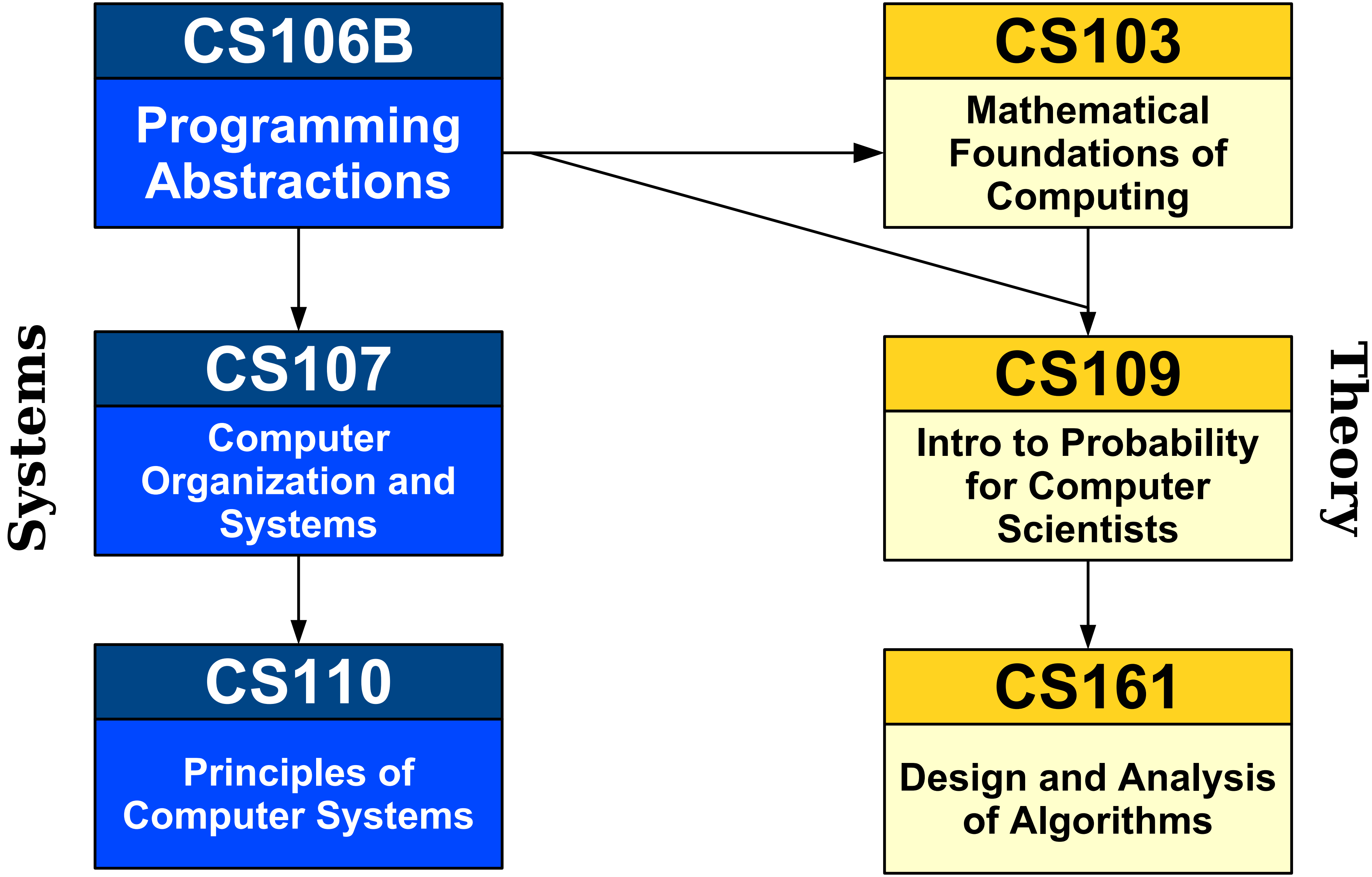
“What motivates you to wake up in the morning?”

Circadian rhythms and my alarm clock. ☺

Back to CS103!

Prerequisite Structures

The CS Core



CS106B
Programming Abstractions
CS103
Mathematical Foundations of Computing
Systems
CS107
Computer Organization and Systems
CS109
Intro to Probability for Computer Scientists
Theory
CS110
Principles of Computer Systems
CS161
Design and Analysis of Algorithms

Pancakes

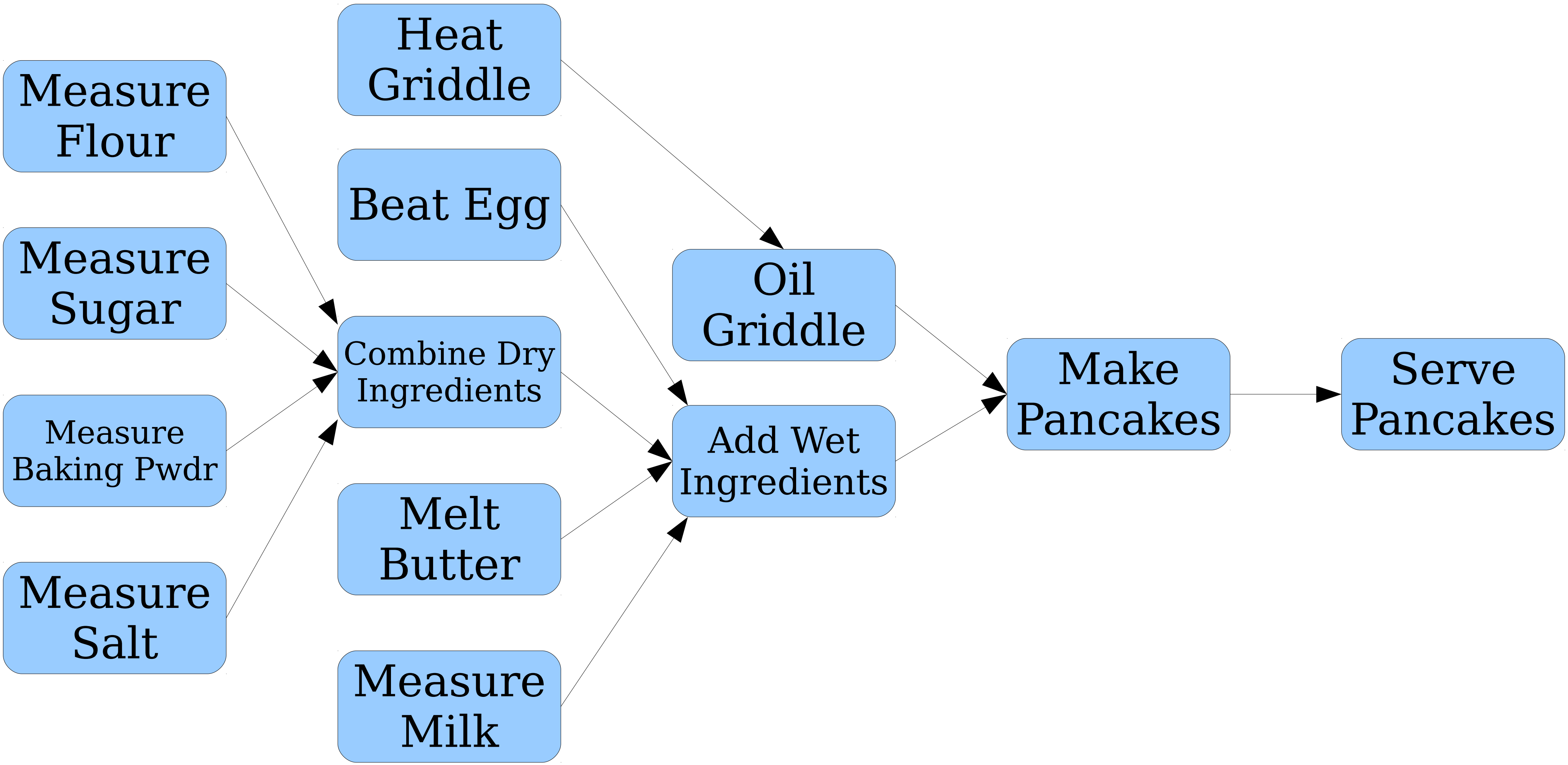
Everyone's got a pancake recipe. This one comes from Food Wishes (http://foodwishes.blogspot.com/2011/08/grandma-kellys-good-old-fashioned.html).

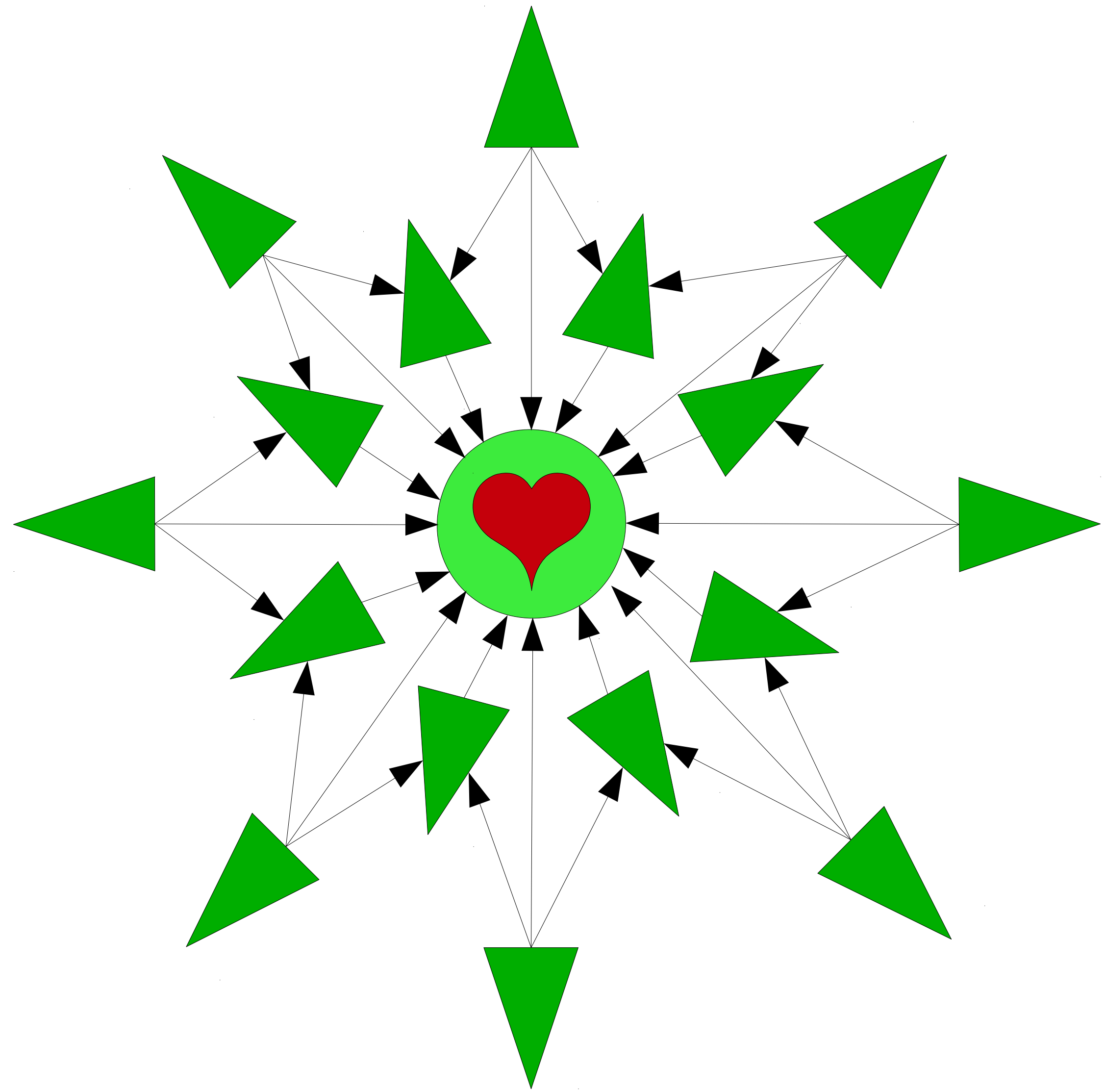
Ingredients

- 1 1/2 cups all-purpose flour
- 3 1/2 tsp baking powder
- 1 tsp salt
- 1 tbsp sugar
- 1 1/4 cup milk
- 1 egg
- 3 tbsp butter, melted

Directions

1. Sift the dry ingredients together.
2. Stir in the butter, egg, and milk. Whisk together to form the batter.
3. Heat a large pan or griddle on medium-high heat. Add some oil.
4. Make pancakes one at a time using 1/4 cup batter each. They're ready to flip when the centers of the pancakes start to bubble.


Measure Flour
Measure Sugar
Measure Baking Pwdr
Measure Salt
Heat Griddle
Beat Egg
Combine Dry Ingredients
Melt Butter
Measure Milk
Oil Griddle
Add Wet Ingredients
Make Pancakes
Serve Pancakes

Relations and Prerequisites

- Let's imagine that we have a prerequisite structure with no circular dependencies.
- We can think about a binary relation R where aRb means

“***a* must happen before *b***”

- What properties of R could we deduce just from this?

$$\forall a \in A.\ a \not R a$$

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

$$\forall a \in A.\ \forall b \in A.\ (aRb \to b \not R a)$$

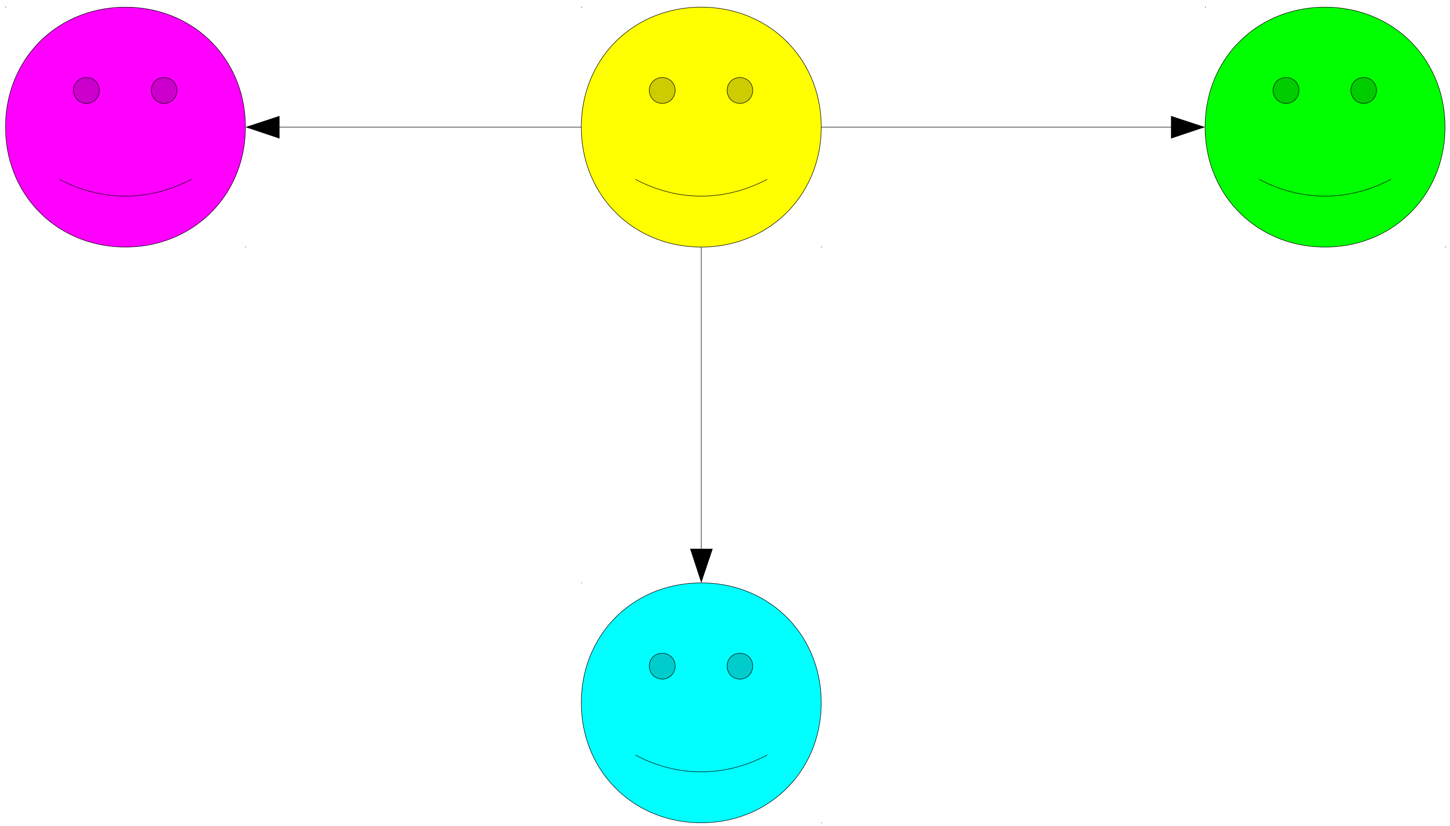
Irreflexivity

- Some relations *never* hold from any element to itself.
- As an example, $x \not< x$ for any x.
- Relations of this sort are called ***irreflexive***.
- Formally speaking, a binary relation R over a set A is irreflexive if the following first-order logic statement is true about R:

$$\forall a \in A.\ a \not R a$$

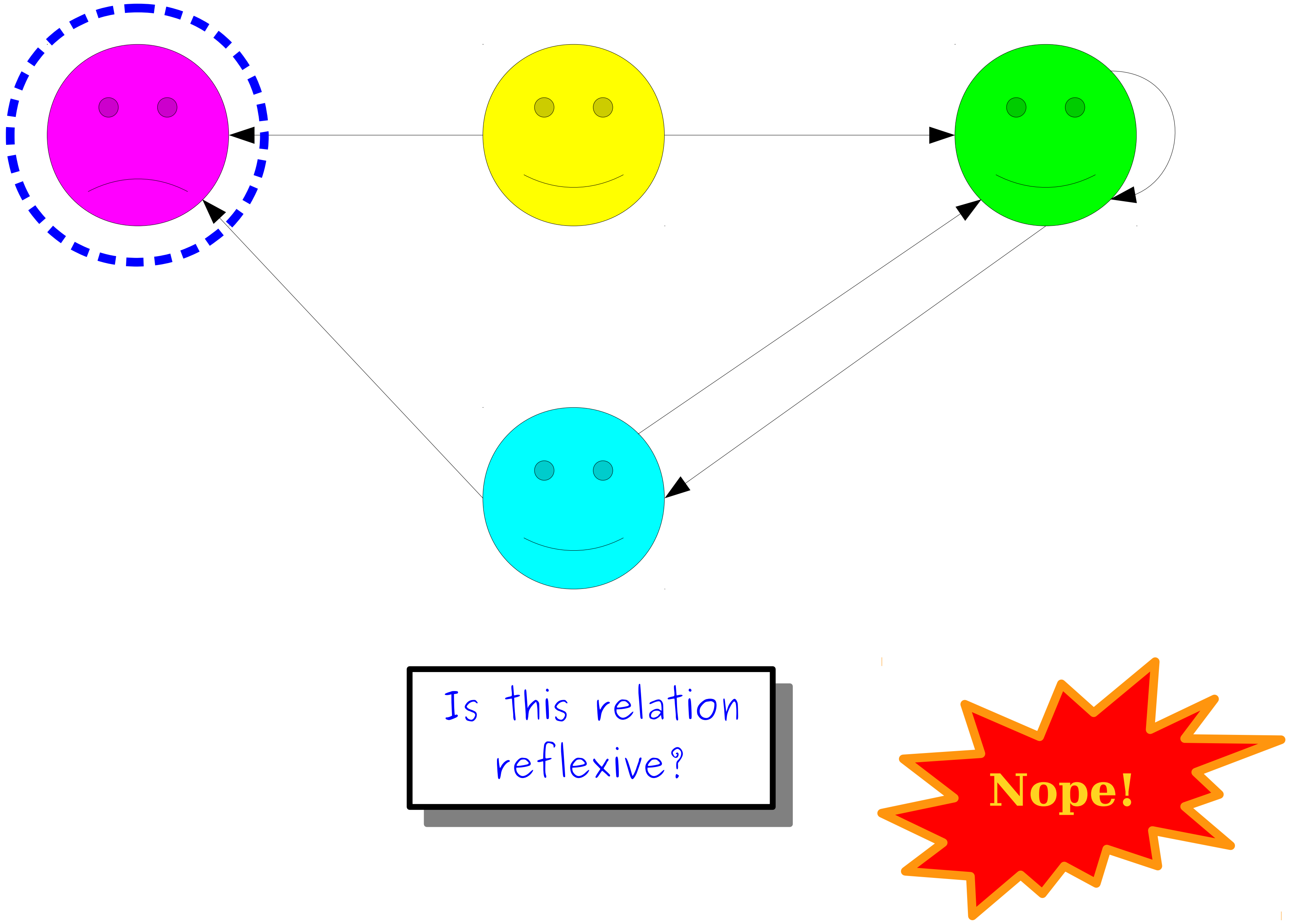
("No element is related to itself.")

Irreflexivity Visualized



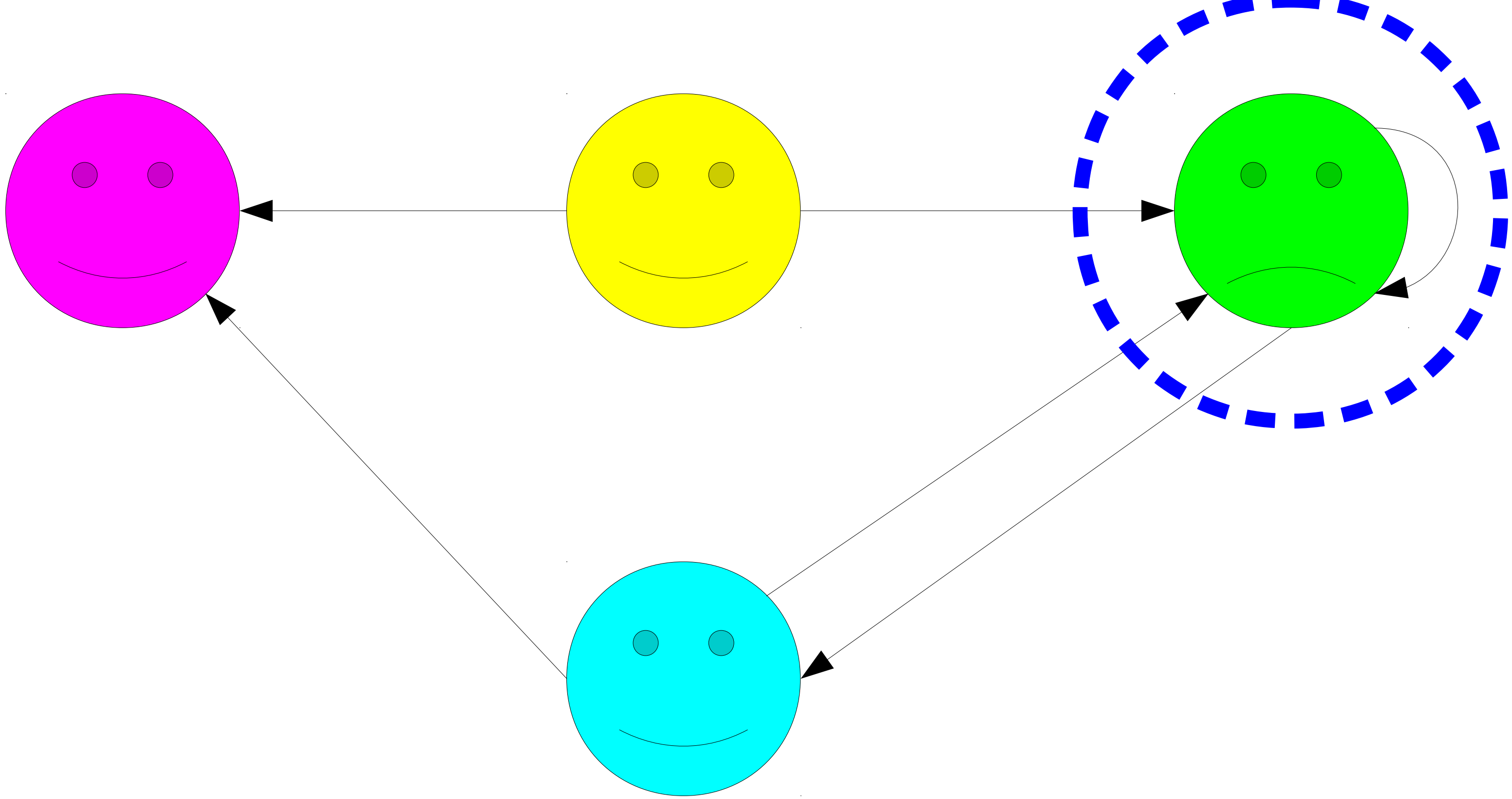
$$\forall a \in A.\ a \not R a$$

(*"No element is related to itself."*)



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

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



Is this relation irreflexive?

$$\forall a \in A.\ a \not R a$$

(*"No element is related to itself."*)

Reflexivity and Irreflexivity

- Reflexivity and irreflexivity are ***not*** opposites!
- Here's the definition of reflexivity:

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

- What is the negation of the above statement?

$$\exists a \in A.\ a\not{R}a$$

- What is the definition of irreflexivity?

$$\forall a \in A.\ a\not{R}a$$

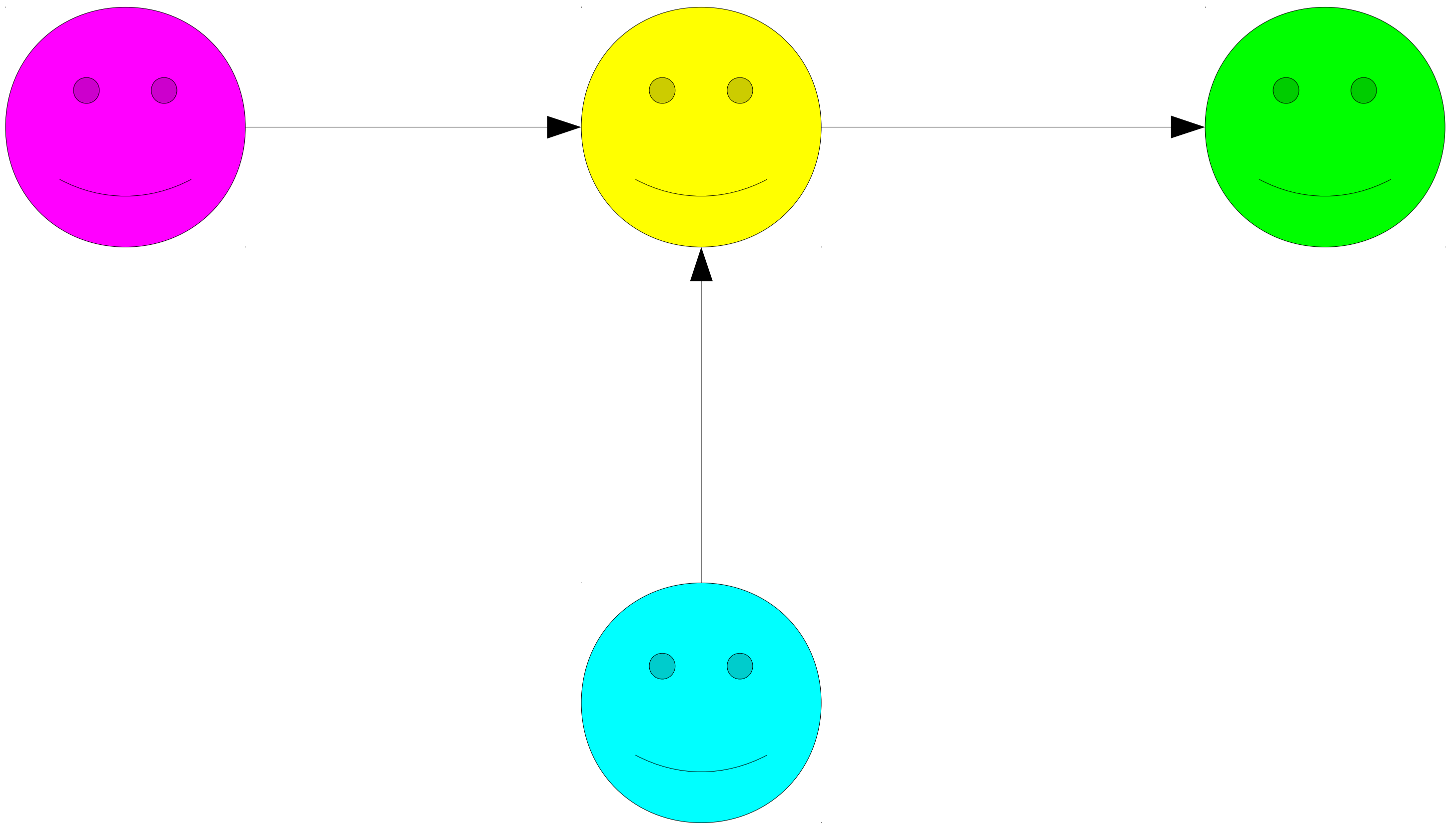
Asymmetry

- In some relations, the relative order of the objects can never be reversed.
- As an example, if $x < y$, then $y \not< x$.
- These relations are called ***asymmetric***.
- Formally: a binary relation R over a set A is called *asymmetric* if the following first-order logic statement is true about R:

$$\forall a \in A.\ \forall b \in A.\ (aRb \rightarrow b\not{R}a)$$

(*"If a relates to b, then b does not relate to a."*)

Asymmetry Visualized



$$\forall a \in A.\ \forall b \in A.\ (aRb \to b\not R a)$$

("If a relates to b, then b does not relate to a.")

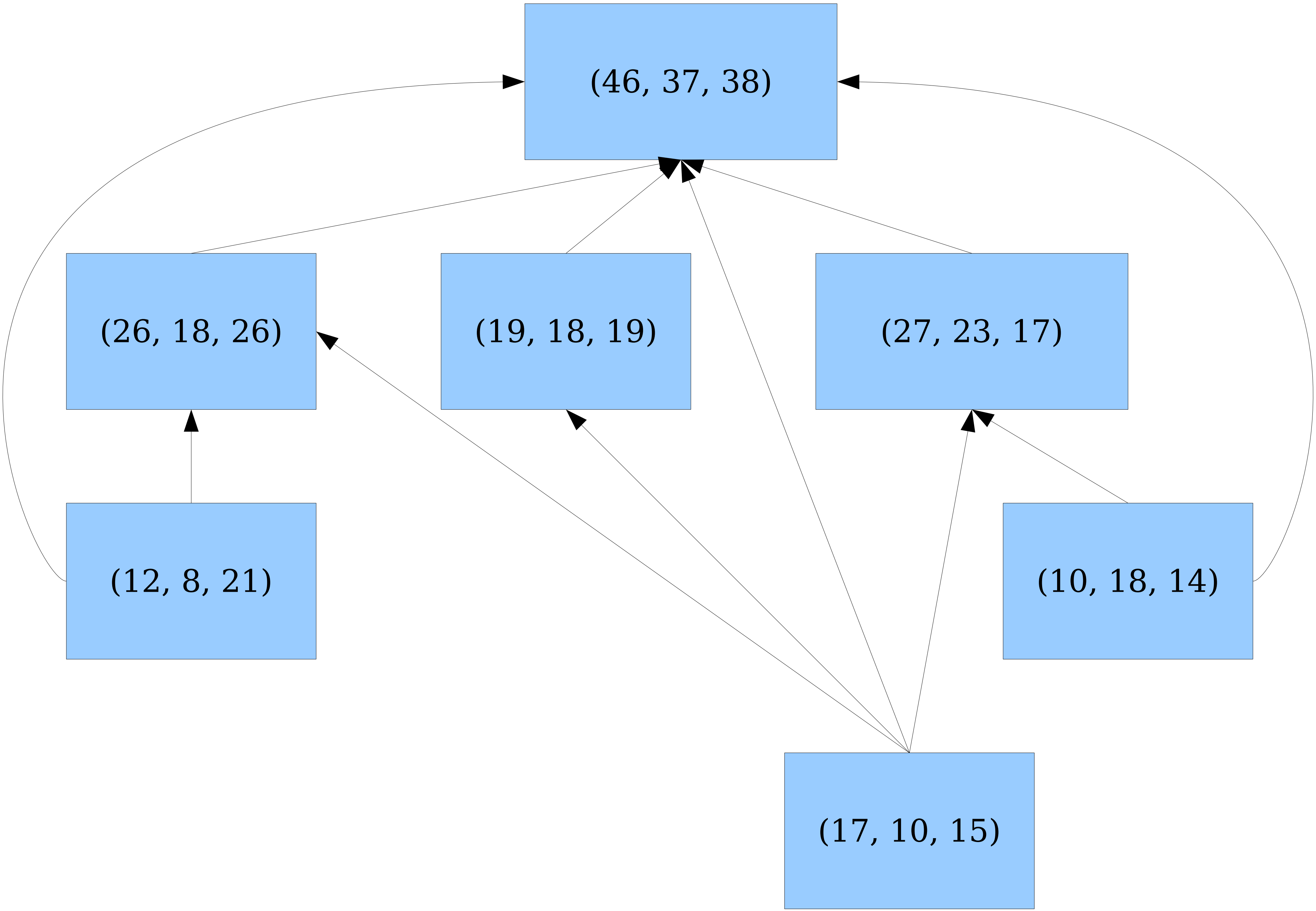
Question to Ponder: Are symmetry and asymmetry opposites of one another?

Strict Orders

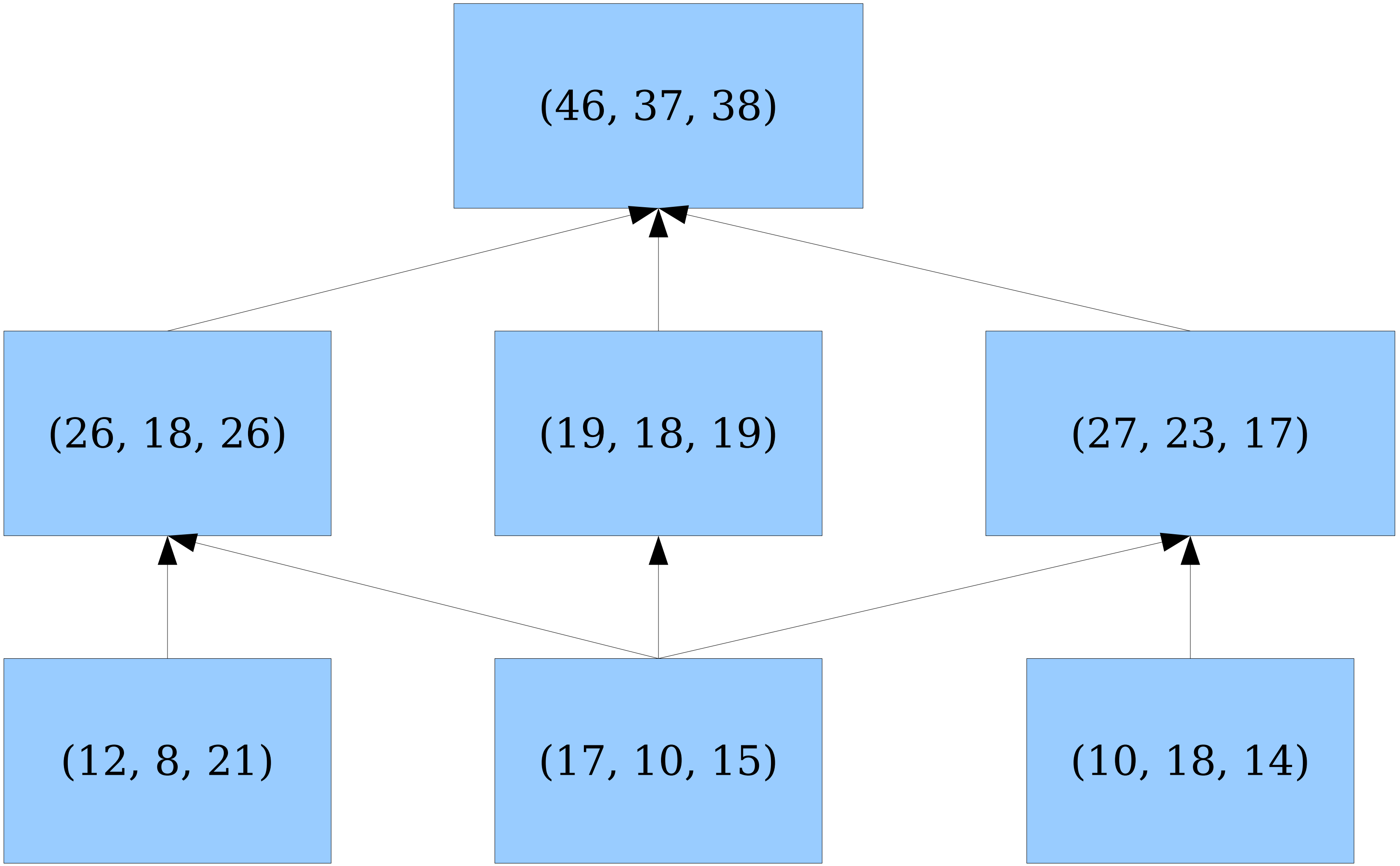
- A ***strict order*** is a relation that is irreflexive, asymmetric and transitive.
- Some examples:
  - $x < y$.
  - a can run faster than b.
  - $A \subsetneq B$ (that is, $A \subseteq B$ and $A \neq B$).
- Strict orders are useful for representing prerequisite structures and have applications in complexity theory (measuring notions of relative hardness) and algorithms (searching and sorting).

Drawing Strict Orders

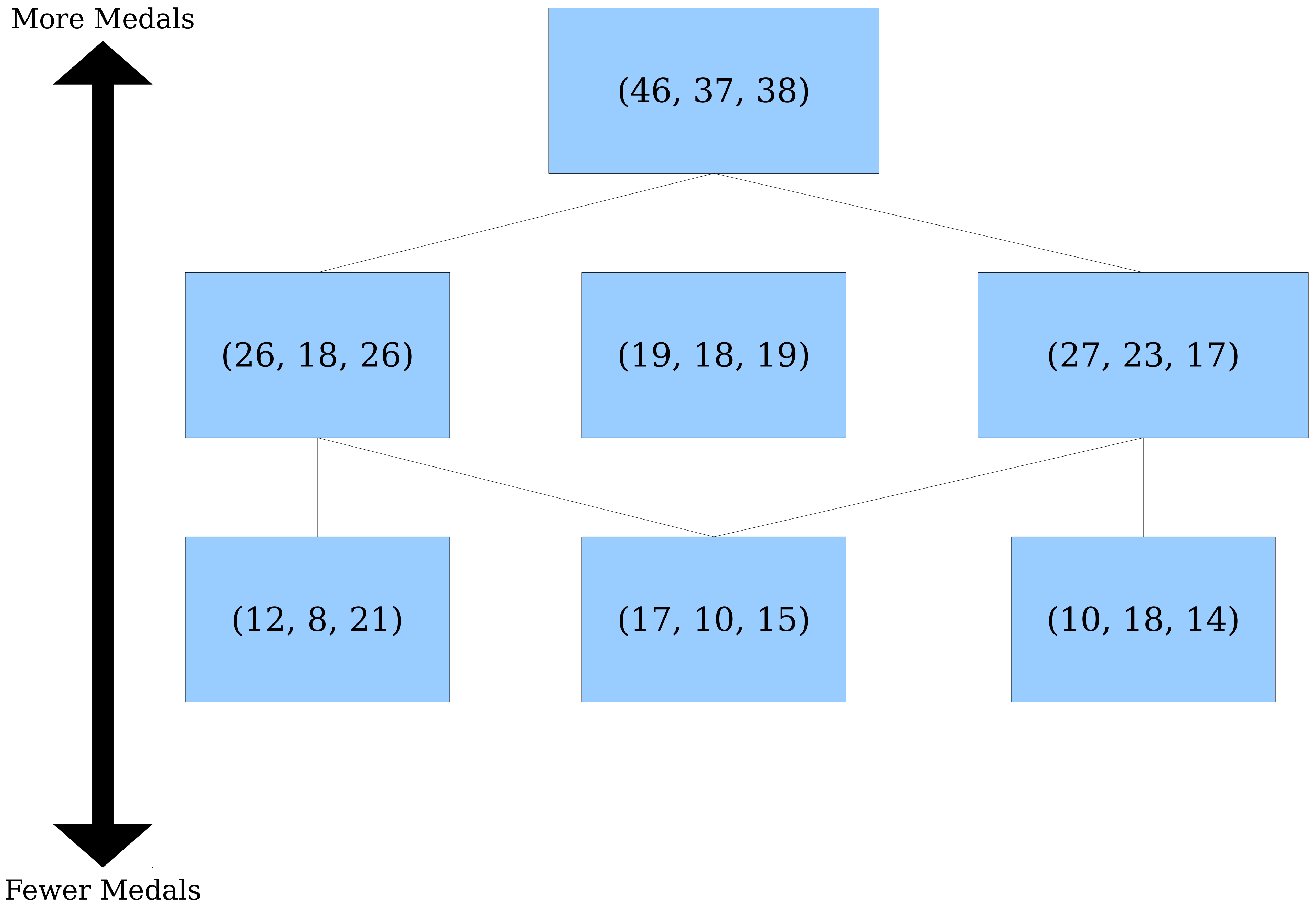
Gold	Silver	Bronze
46	37	38
27	23	17
26	18	26
19	18	19
17	10	15
12	8	21
10	18	14
9	3	9
8	12	8
8	11	10
8	7	4
8	3	4
7	6	6
7	4	6
6	6	1
6	3	2



$$(g_1, s_1, b_1)\ R\ (g_2, s_2, b_2) \qquad \text{if} \qquad g_1 < g_2 \ \wedge\ s_1 < s_2 \ \wedge\ b_1 < b_2$$



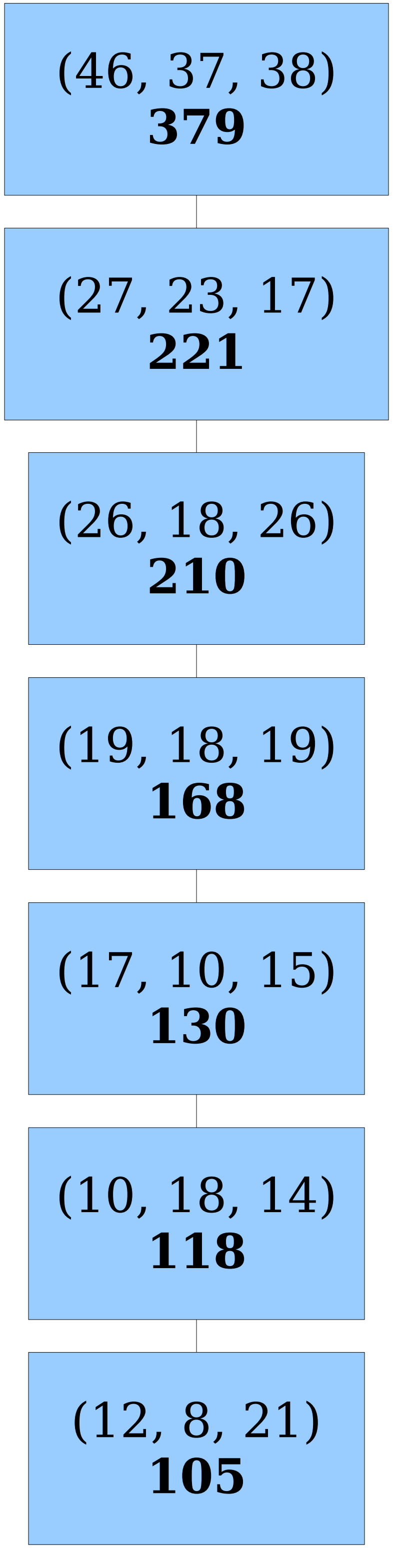
$$(g_1, s_1, b_1)\ R\ (g_2, s_2, b_2) \quad \text{if} \quad g_1 < g_2 \ \wedge\ s_1 < s_2 \ \wedge\ b_1 < b_2$$



$$(g_1, s_1, b_1)\ R\ (g_2, s_2, b_2) \qquad \text{if} \qquad g_1 < g_2 \ \wedge\ s_1 < s_2 \ \wedge\ b_1 < b_2$$

Hasse Diagrams

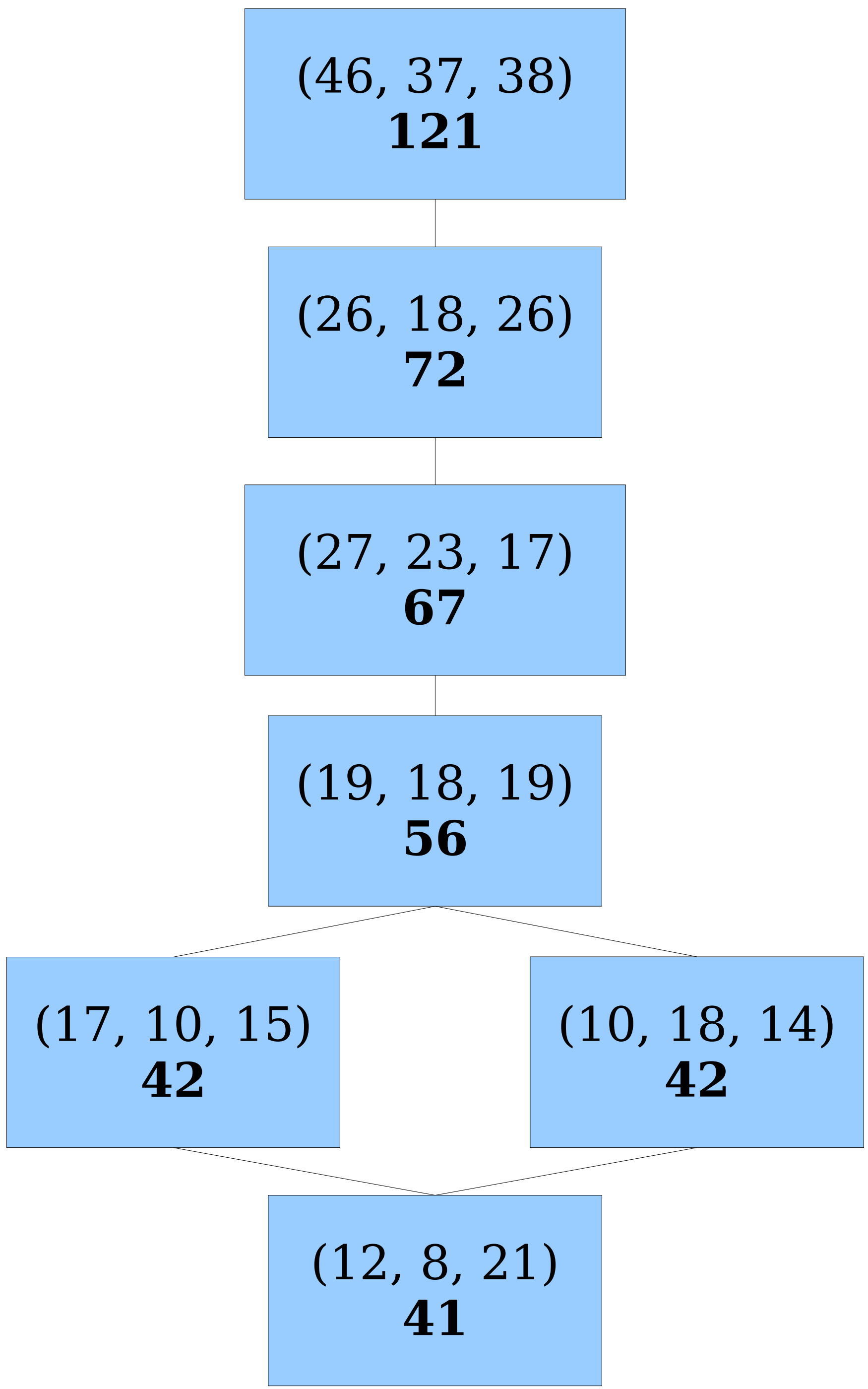
- A ***Hasse diagram*** is a graphical representation of a strict order.
- Elements are drawn from bottom-to-top.
- No self loops are drawn, and none are needed! By ***irreflexivity*** we know they shouldn't be there.
- Higher elements are bigger than lower elements: by ***asymmetry***, the edges can only go in one direction.
- No redundant edges: by ***transitivity***, we can infer the missing edges.



$(g_1, s_1, b_1)\ T\ (g_2, s_2, b_2)$

if

$5g_1 + 3s_1 + b_1 < 5g_2 + 3s_2 + b_2$

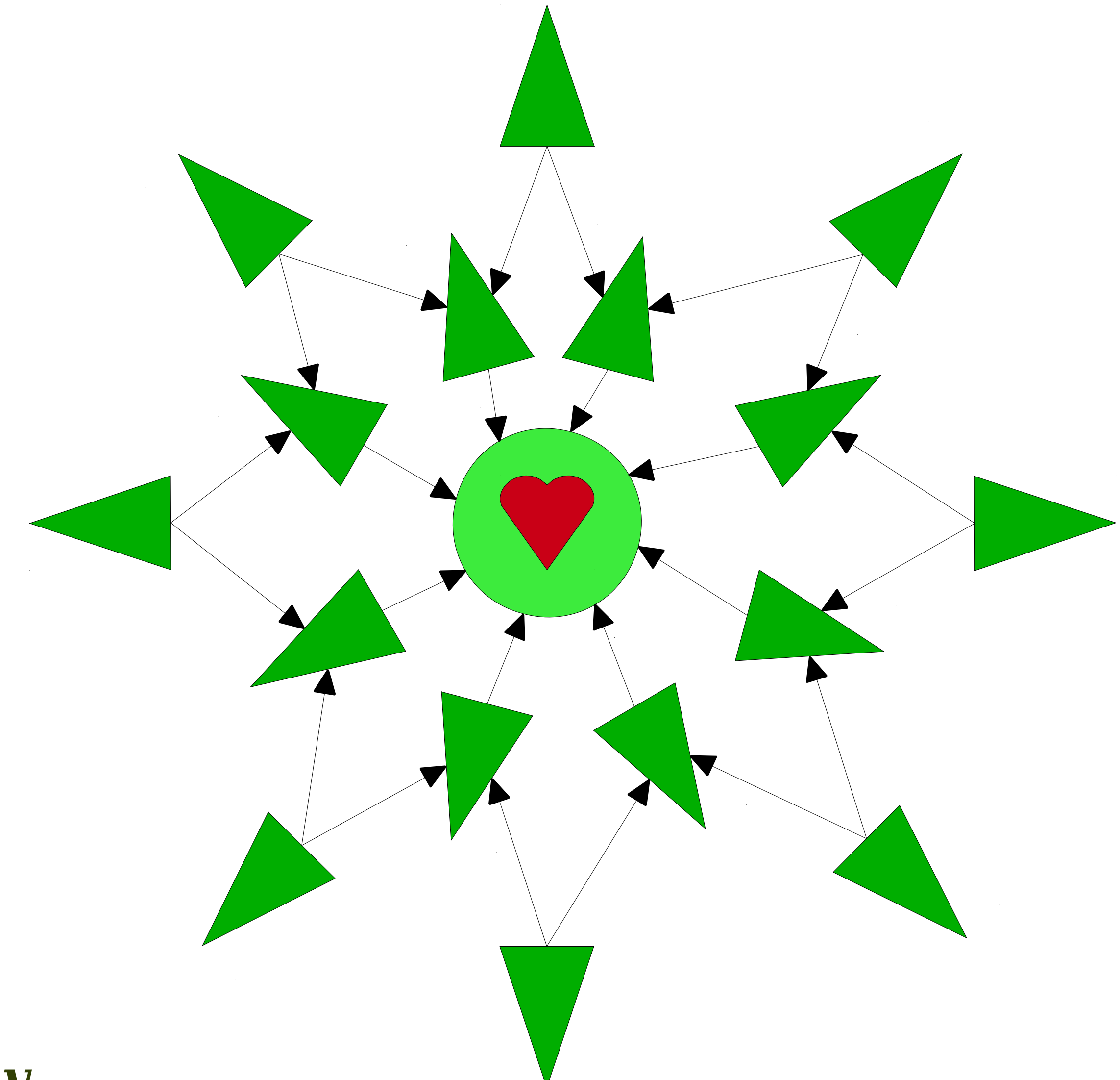


$$(g_1, s_1, b_1)\ U\ (g_2, s_2, b_2)$$

if

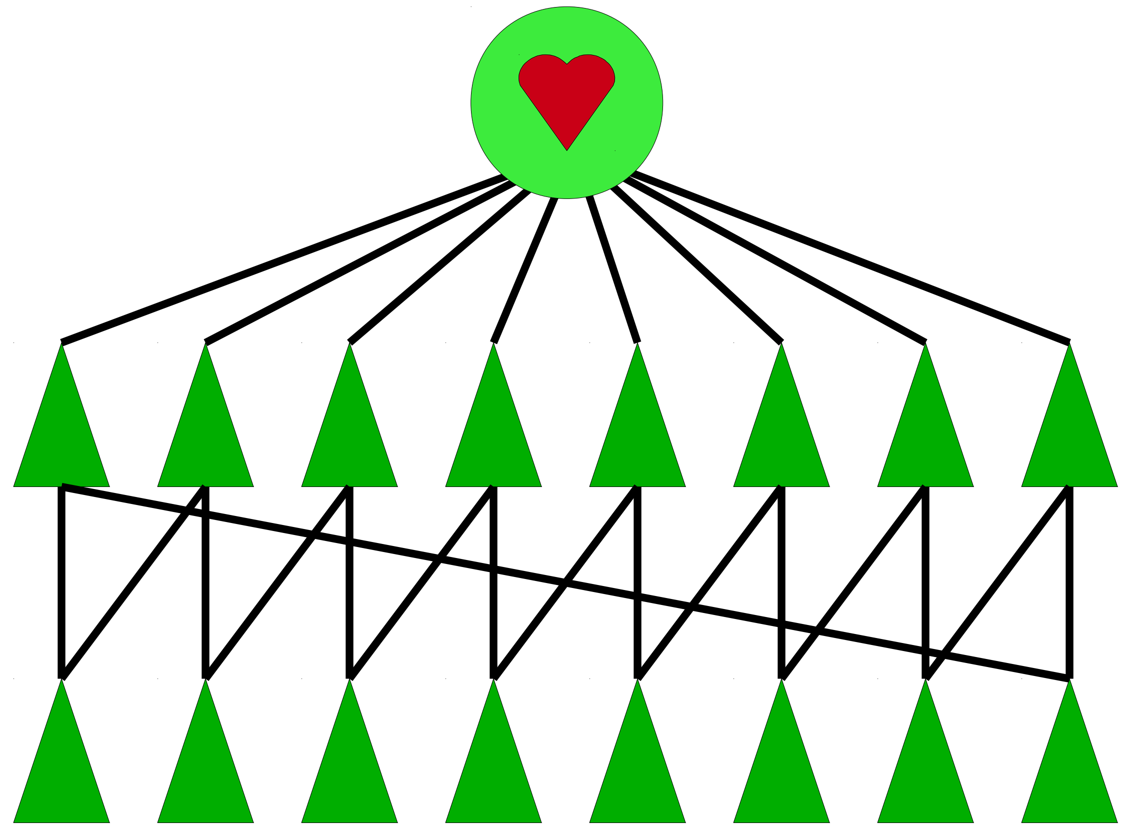
$$g_1 + s_1 + b_1 < g_2 + s_2 + b_2$$

Hasse Artichokes



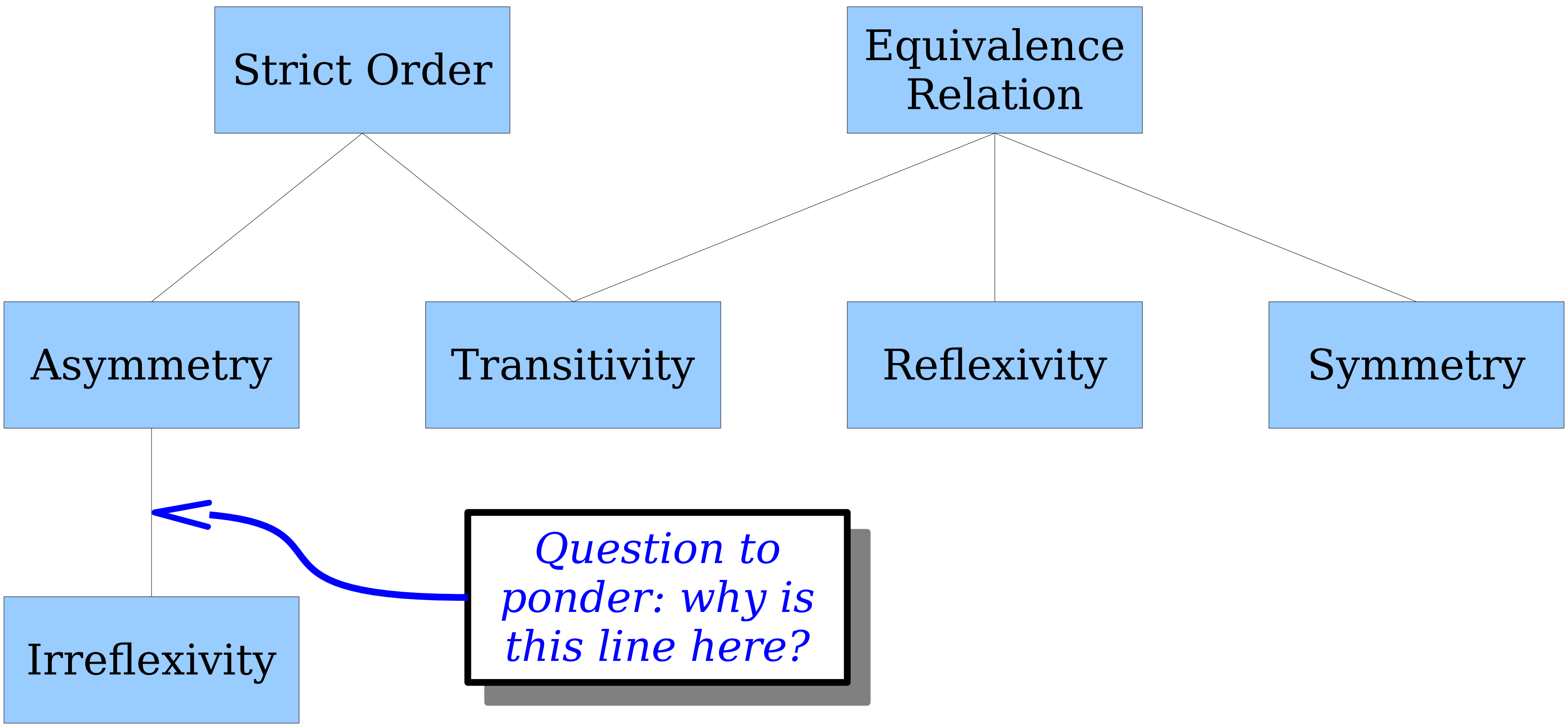
xRy if x must be eaten before y

Hasse Artichokes



xRy if x must be eaten before y

The Meta Strict Order



aRb **if** ***a*** **is less specific than** ***b***

Next Time

- ***Functions***
  - How do we model transformations in a mathematical sense?
- ***Domains and Codomains***
  - Type theory meets mathematics!
- ***Injections, Surjections, and Bijections***
  - Three special classes of functions.