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 $\boldsymbol{aRb}$.
  - For example: $3 = 3$, $5 < 7$, and $\emptyset \subseteq \mathbb{N}$.
- If R is a binary relation over A and it does not hold for the pair (a, b), we write $\boldsymbol{a \not R b}$.
  - For example: $4 \neq 3$, $4 \not< 3$, and $\mathbb{N} \not\subseteq \emptyset$.

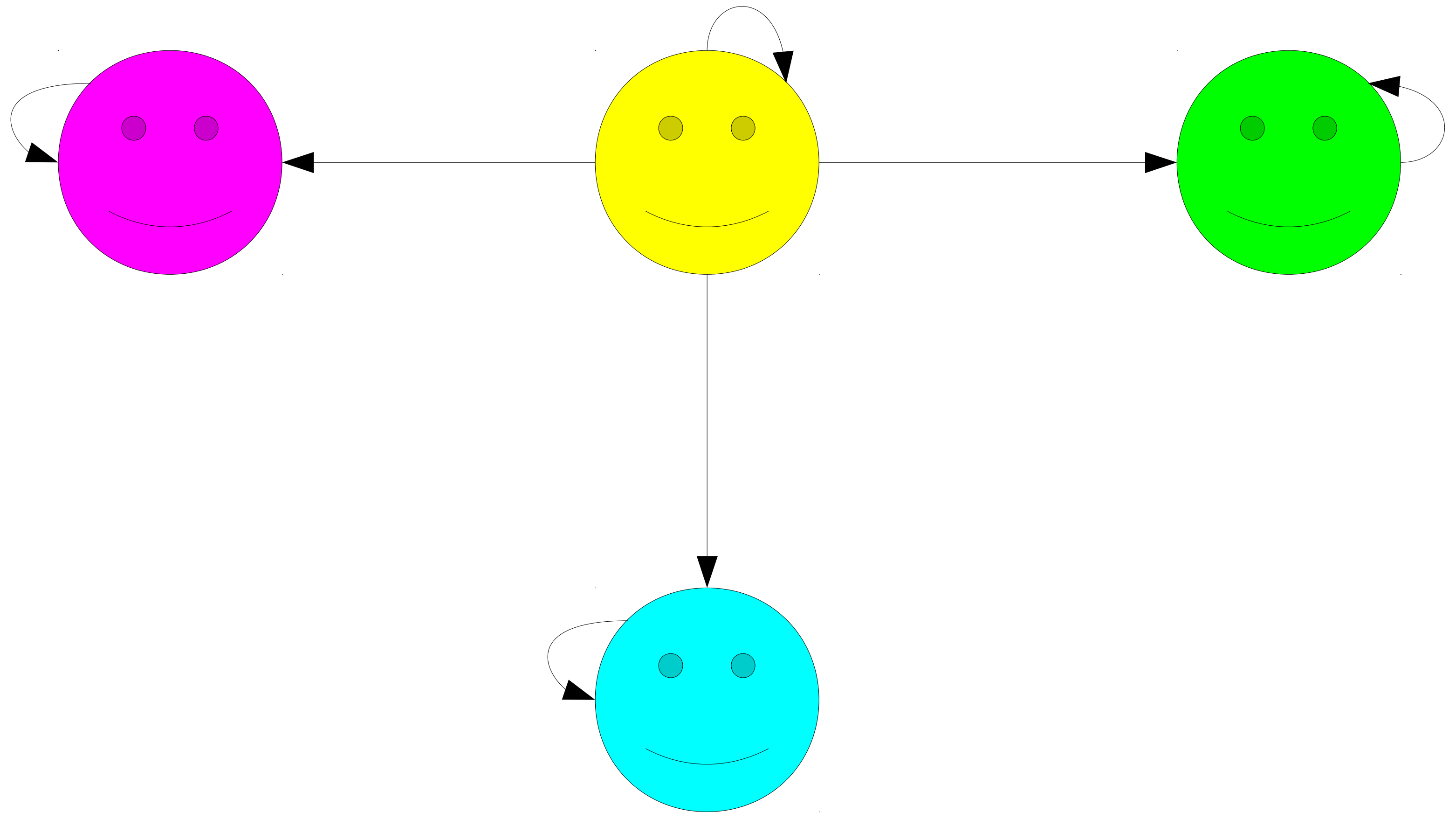
Reflexivity

- 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."*)

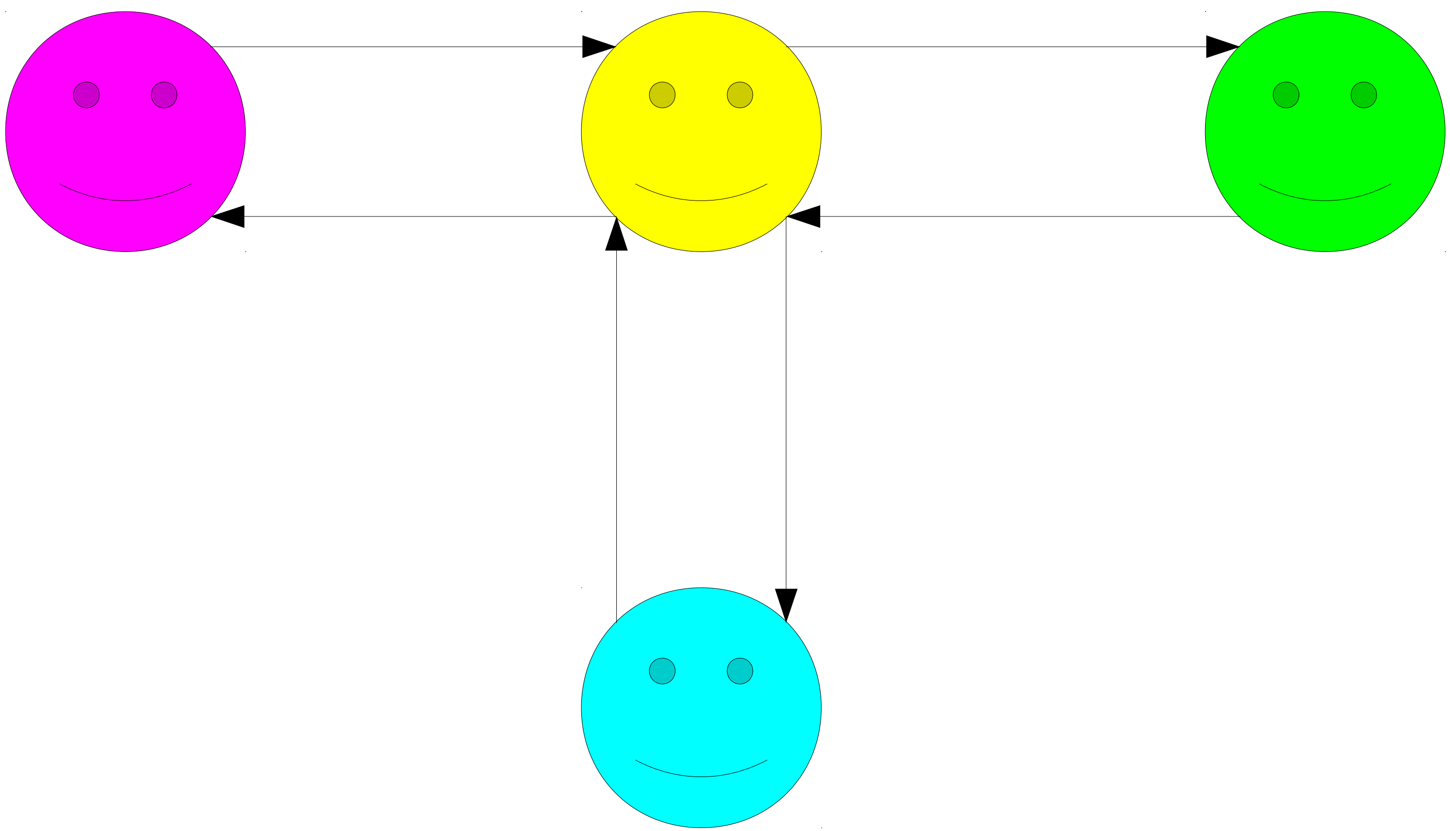
Symmetry

- 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



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

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

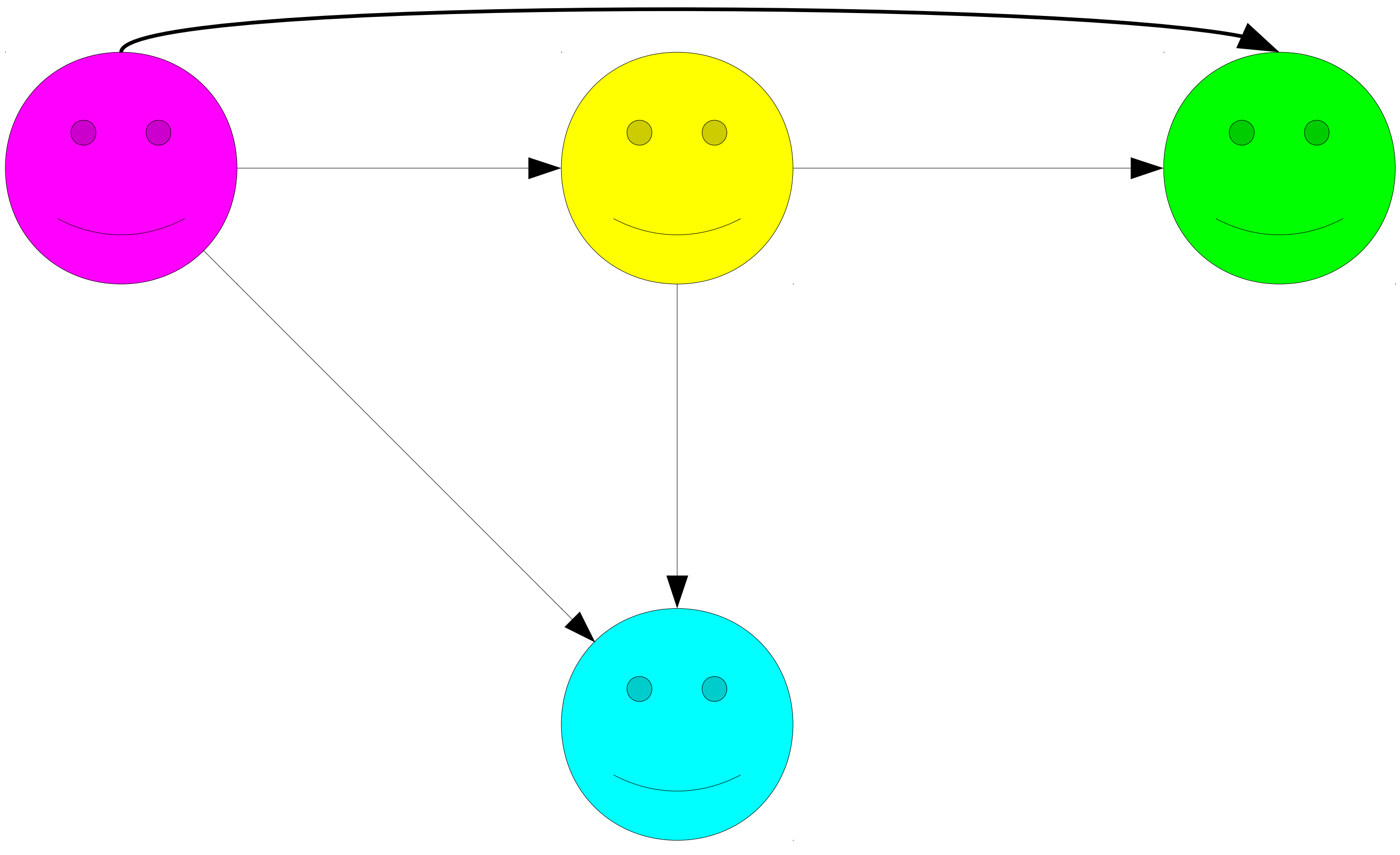
Transitivity

- 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:

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

(*"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.

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
  - $[0]_{\equiv_3} = \{0, 3, 6, 9, 12, 15, 18, \ldots\}$
  - $[1]_{\equiv_3} = \{1, 4, 7, 10, 13, 16, 19, \ldots\}$
  - $[2]_{\equiv_3} = \{2, 5, 8, 11, 14, 17, 20, \ldots\}$
  - $[3]_{\equiv_3} = \{0, 3, 6, 9, 12, 15, 18, \ldots\}$

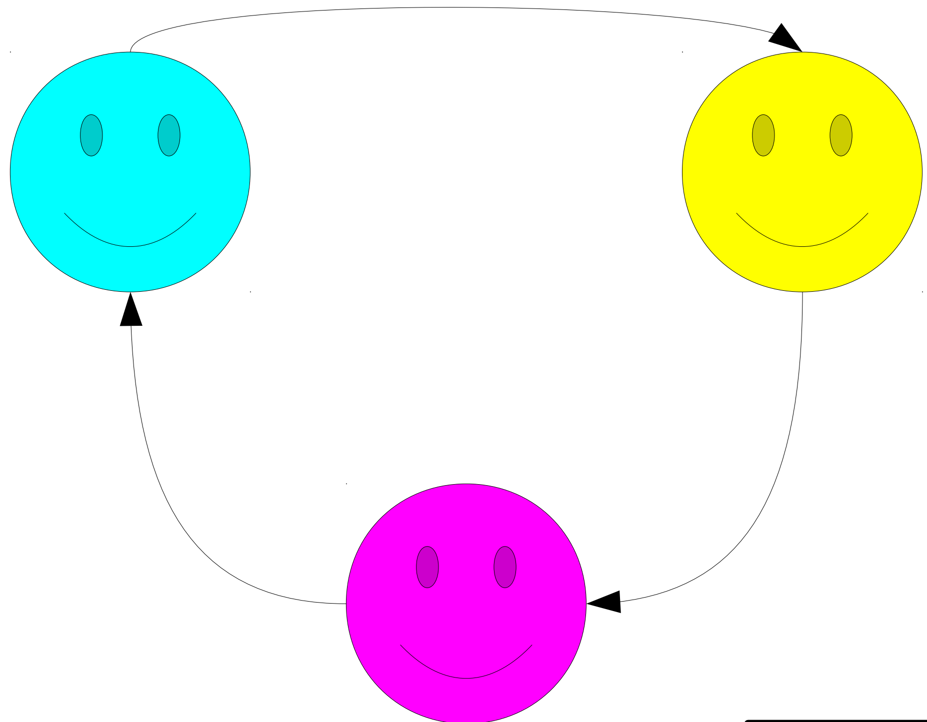
Notice that $[0]_{\equiv_3} = [3]_{\equiv_3}$. These are *literally* the same set, so they're just different names for the same thing.

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.

New Stuff!

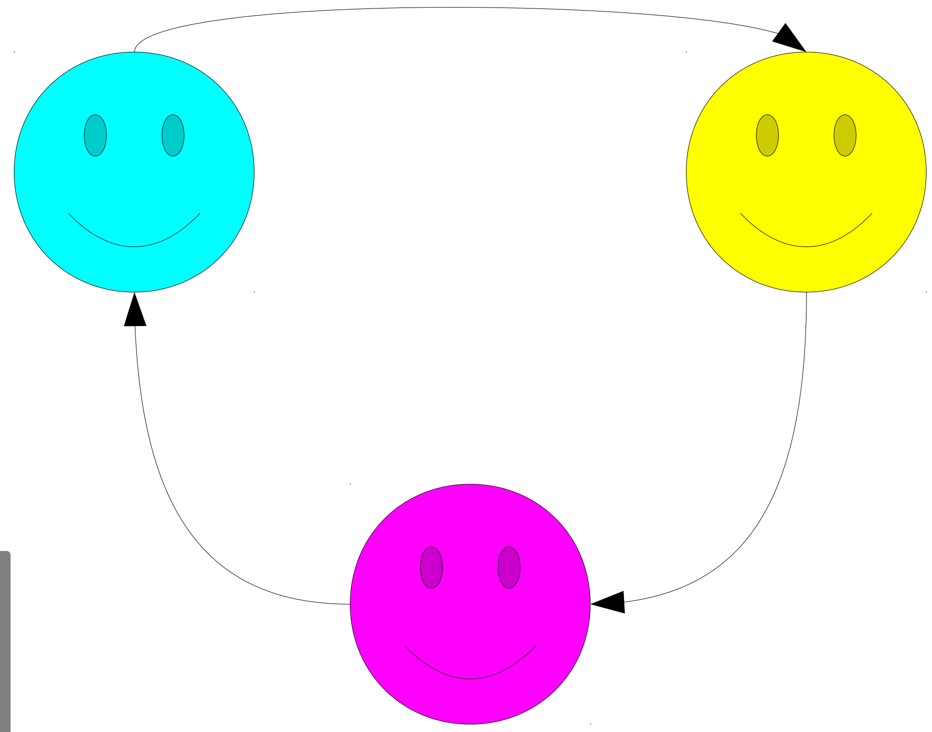
How'd We Get Here?

- 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?



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

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



Is this an equivalence relation?

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

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

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.

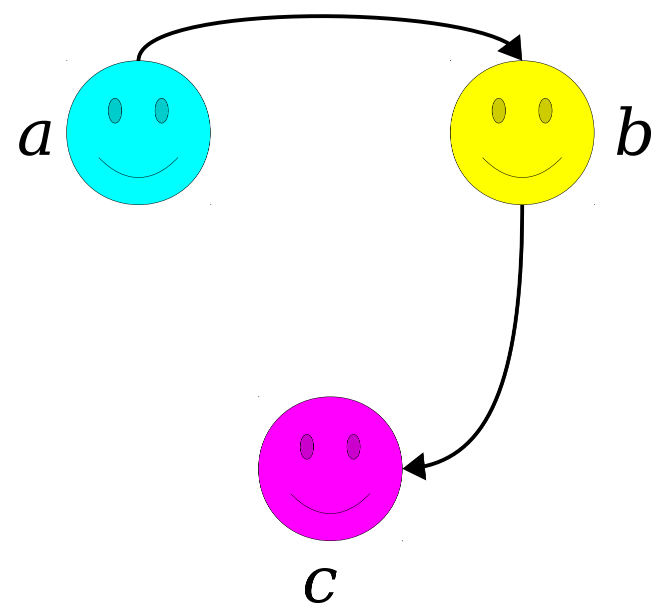
Lemma 1: If R is an equivalence relation over a set A, then R is reflexive and cyclic.

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.

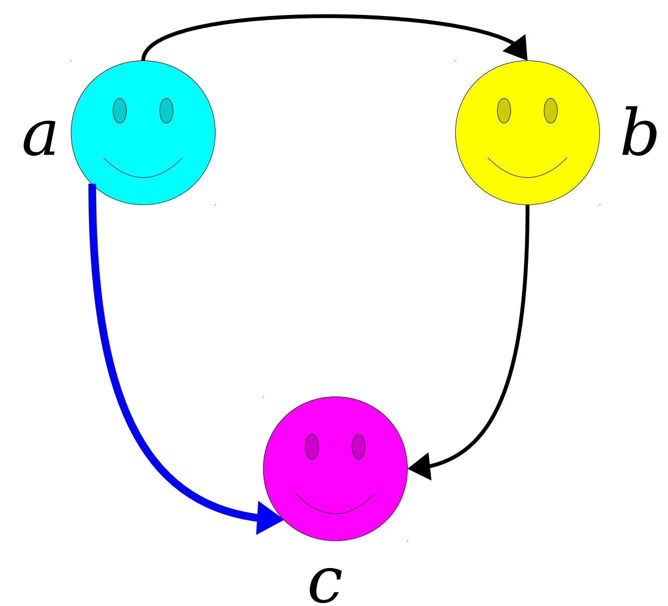
What We're Assuming

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.

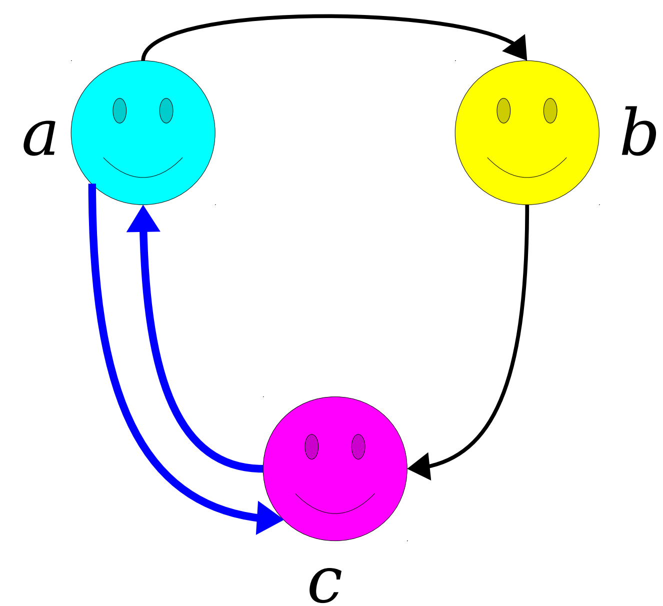
R is reflexive.

- 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, symmetr
already know that
that R is cyclic.

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 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!

Lemma ... en R is refl...

Proof: L... me set A. ...

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

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!

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	*What We Need To Show*
• R is reflexive. • R is cyclic.	• 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	*What We Need To Show*
• R is reflexive. • R is cyclic.	R is an equivalence relation. R is reflexive. • R is symmetric. R is transitive.

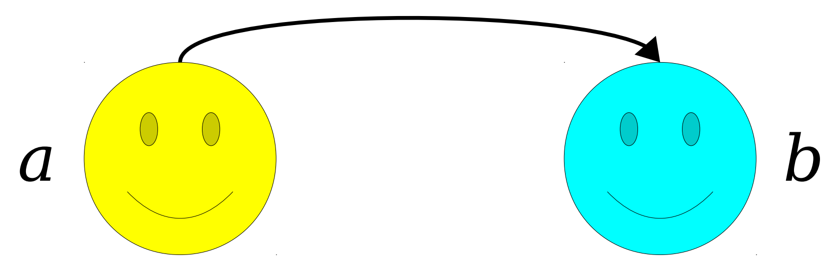
Lemma 2: If R is a binary relation over a set A that is reflexive and cyclic, then R is an equivalence relation.

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.

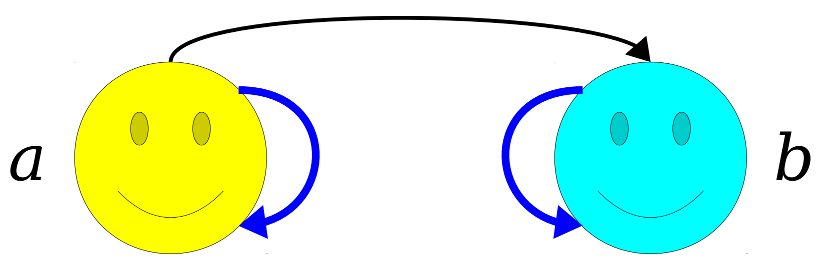
- ∀x ∈ A. xRx

R is cyclic.

xRy ∧ yRz → 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.

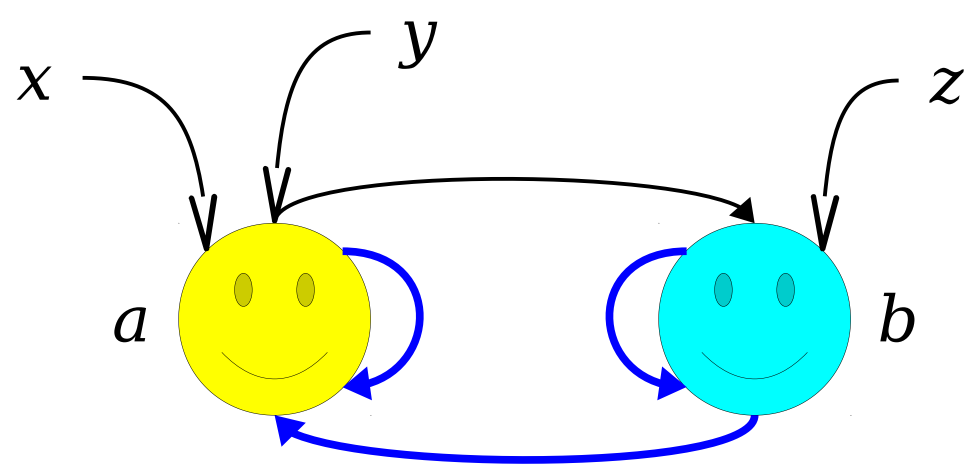
$\forall x \in A.\ xRx$

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 an equivalence relation.

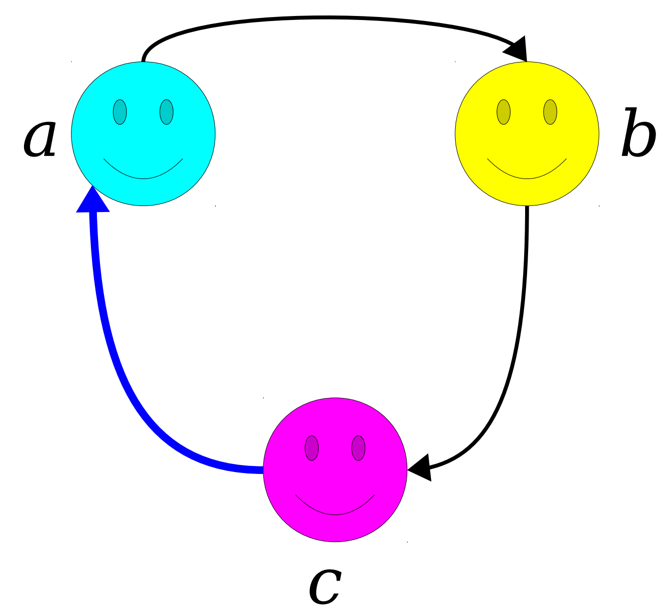
R is reflexive.

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	*What We Need To Show*
R is reflexive.	• R is transitive.
$\forall x \in A.\ xRx$	• If aRb and bRc, then aRc.
R is cyclic.	
$xRy \land yRz \to zRx$	



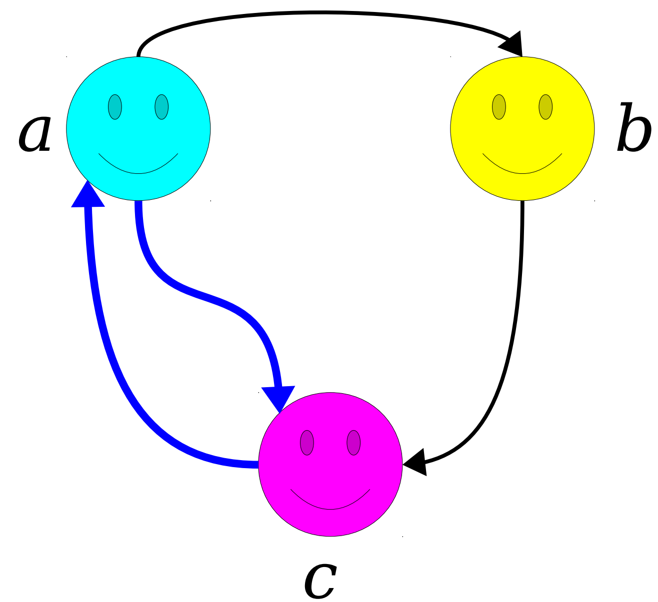
Lemma 2: If R is a binary relation over a set A that is reflexive and cyclic, then R is an equivalence relation.

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
anc

Proof that
is c
equ R is
ref
knc 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…

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

> 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!

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

Proof: Let

is cyclic

equivale

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.

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.

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 ***should 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!

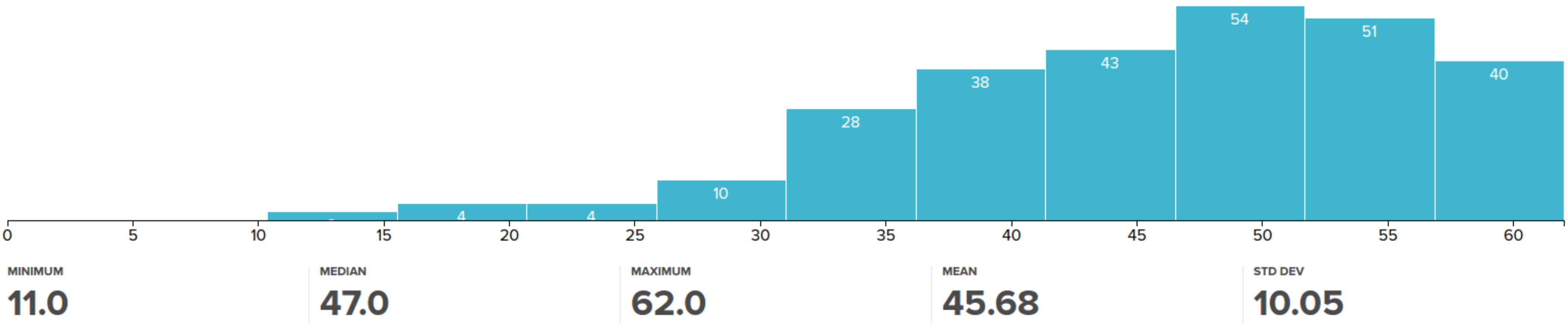
Time-Out for Announcements!

My Office Hours

- Oops! I forgot to put my office hours into the OH timetable.
- They're Thursdays, 2:15PM – 4:15PM, in Gates 167.
- Feel free to stop on by!

Problem Set One Graded

- We’ve finished grading Problem Set One. Feedback is available on GradeScope.
- Here’s the distribution:



What To Do Next

- ***Review the grader's feedback***. We try to leave detailed feedback on each problem. Look over our notes and see if you can find some concrete, tangible ways to improve going forward.
- ***Don't get discouraged***. This problem set is downweighted relative to the other problem sets this quarter. For the overwhelming majority of you, this is your first time writing proofs. Don't extrapolate from just one data point – figure out where to focus your efforts, and try to make new mistakes each time.

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 the start of class.
  - Have questions? Stop by office hours or ask on Piazza!
- We've released a handout containing a first-order logic translation checklist. We *highly* recommend reviewing your translations using that checklist before submitting!

Your Questions

“Why is biocomp rare? Will it be big in the future?”

Biocomputation is a really big field right now! The number of CS majors graduating with the biocomputation track has exploded over the past few years (much faster than the general CS major), which is really exciting!

I've seen some amazing talks by Gill Bejerano and Serafim Batzoglu about the work they're doing working out how genes control one another and how to sequence cancer genomes, and it really feels like science fiction. This is a very cool area to explore!

“Do you think Silicon Valley has a good moral compass?”

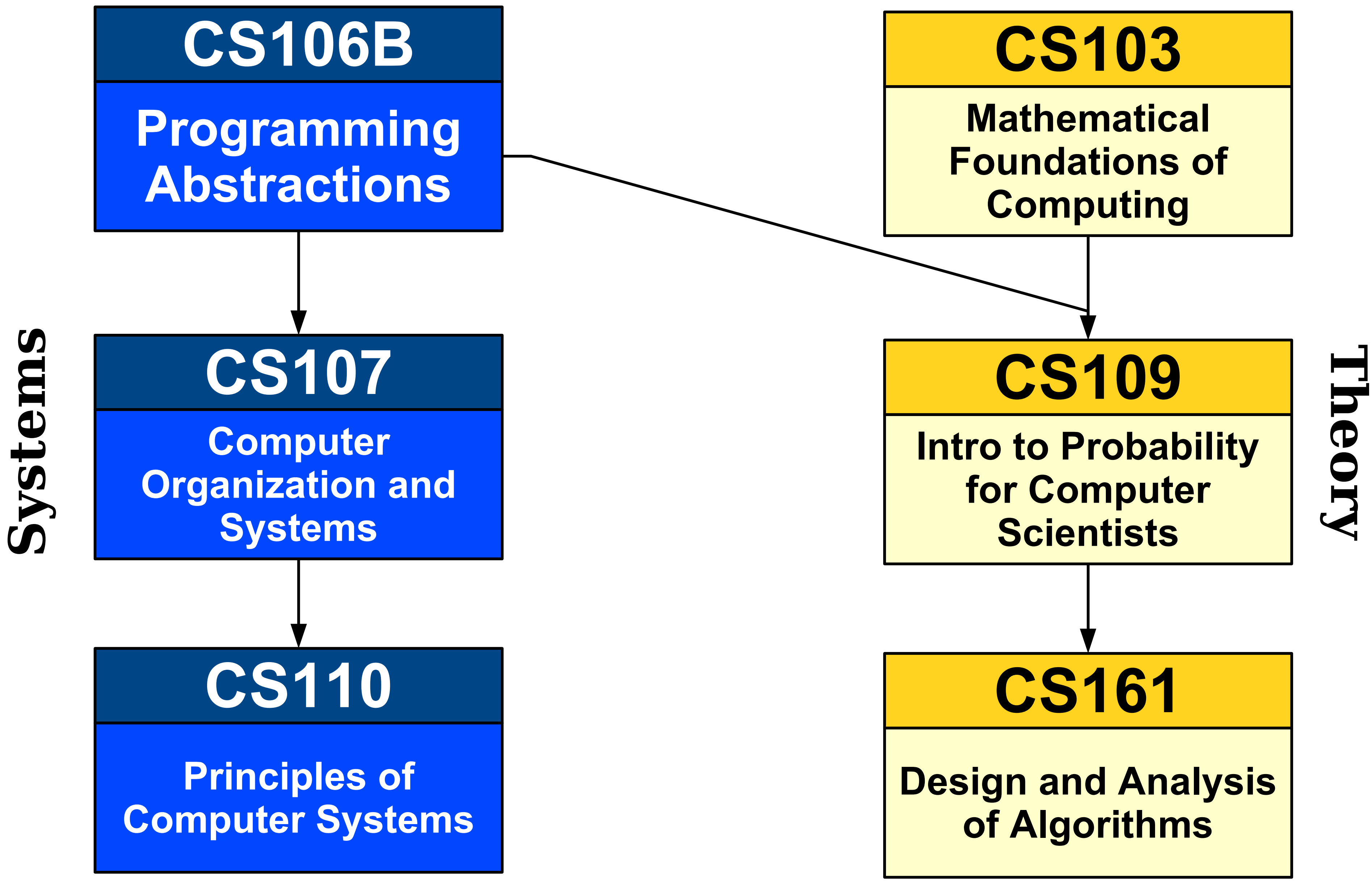
I think it's useful to think about things like this from a few perspectives. First, what does the leadership at a company value? Second, what are the incentive structures? Third, what are the broader values of the community?

There are many areas where I think the tech industry has things right. There are many areas where I think the tech industry has things wrong. But I wouldn't necessarily attribute it to a “moral compass.” I (personally) think aggregate behavior is best explained by the three above factors.

Back to CS103!

Prerequisite Structures

The CS Core



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

Pancakes

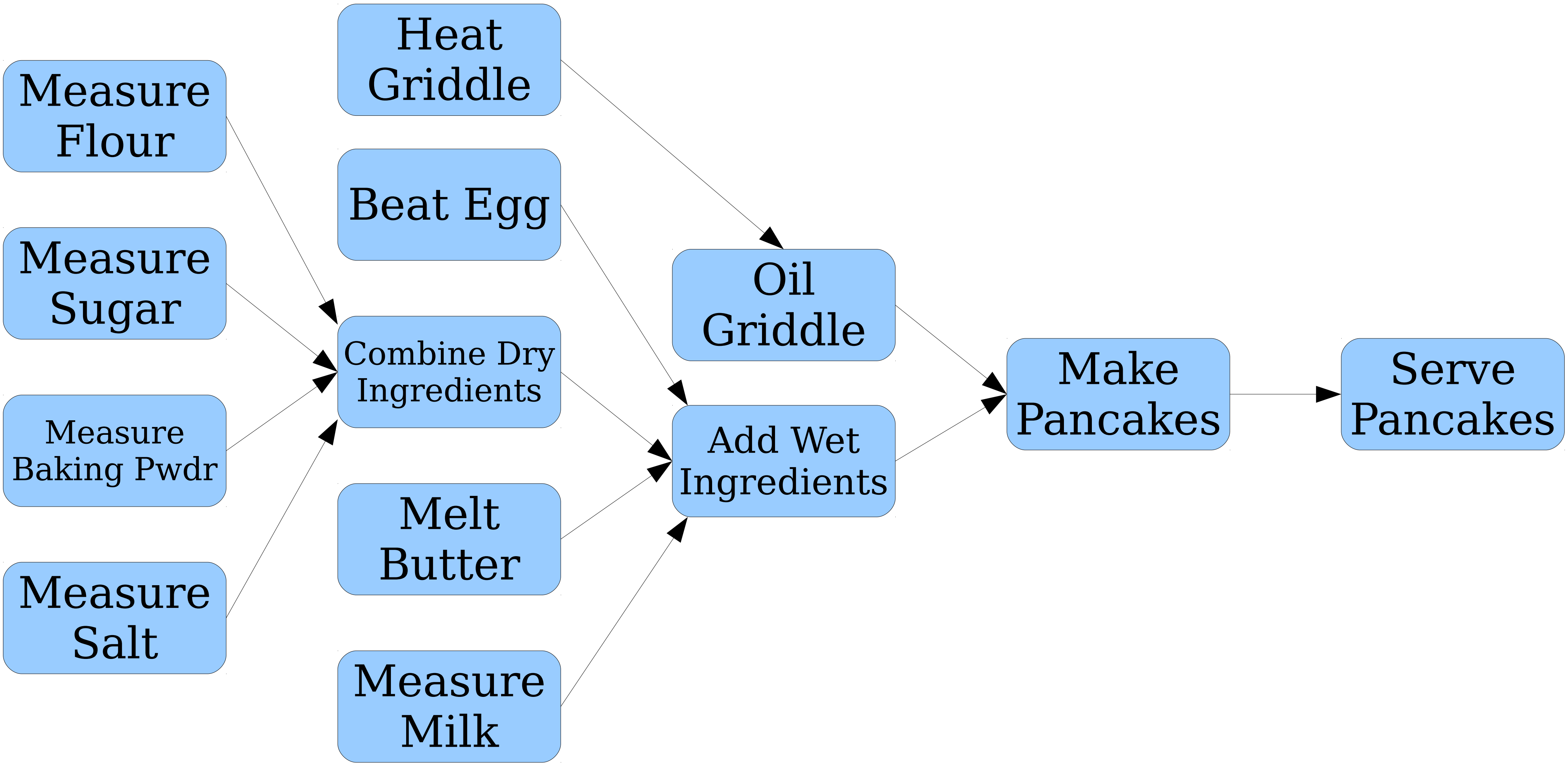
Everyone's got a pancake recipe. This one comes from Food Wishes (http://foodwishes.blogspot.com/2011/08/grandma-kellys-good-old-fashioned.html).

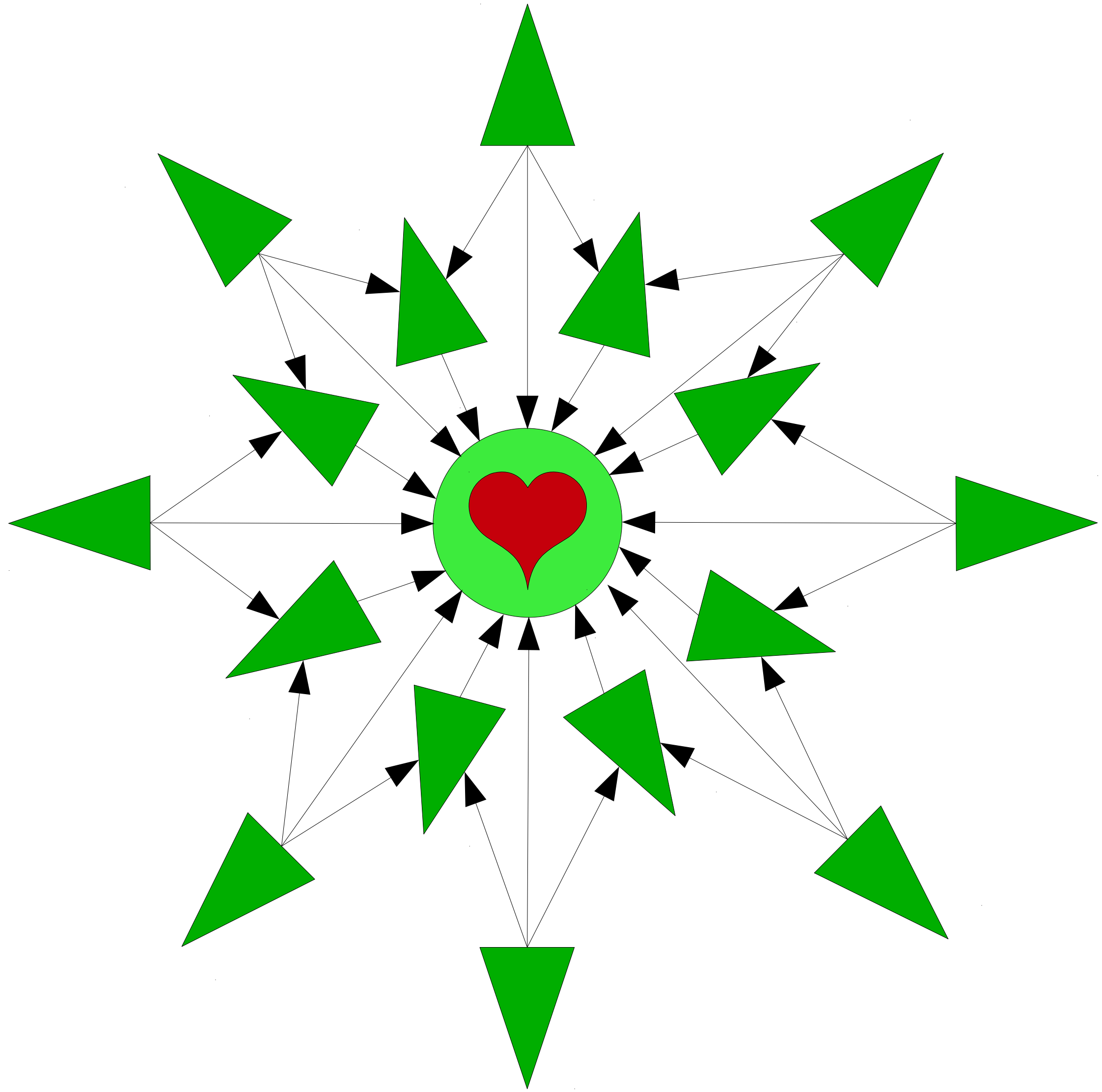
Ingredients

- 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

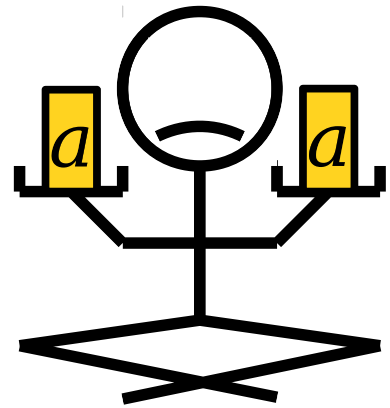


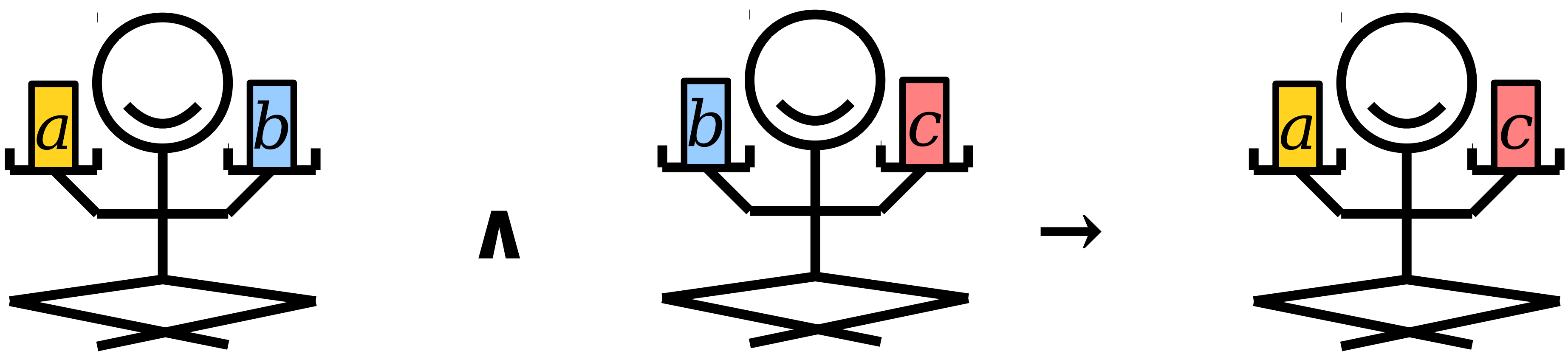
Relations and Prerequisites

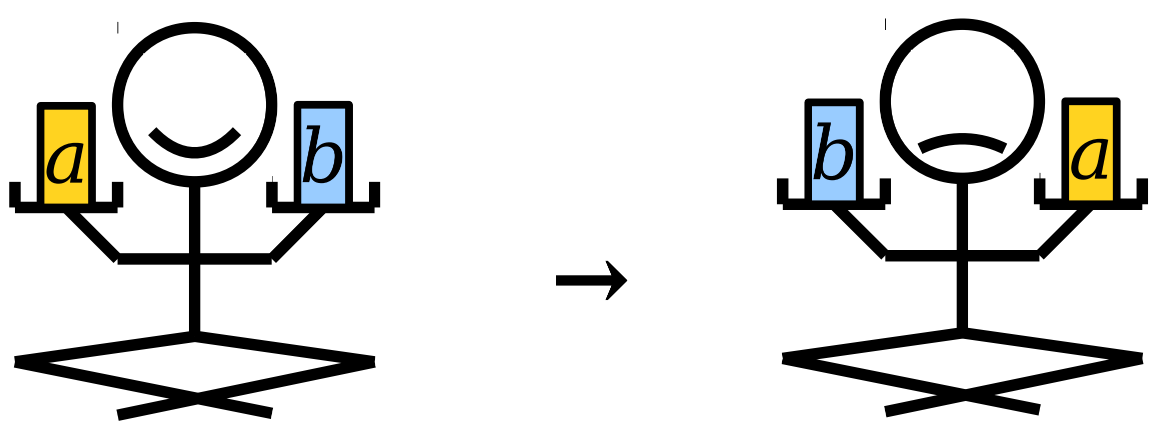
- Let's imagine that we have a prerequisite structure with no circular dependencies.
- We can think about a binary relation R where aRb means

“***a* must happen before *b***”

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


a
a


a
b
∧
b
c
→
a
c


a
b
→
b
a

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

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

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

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

Transitivity

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

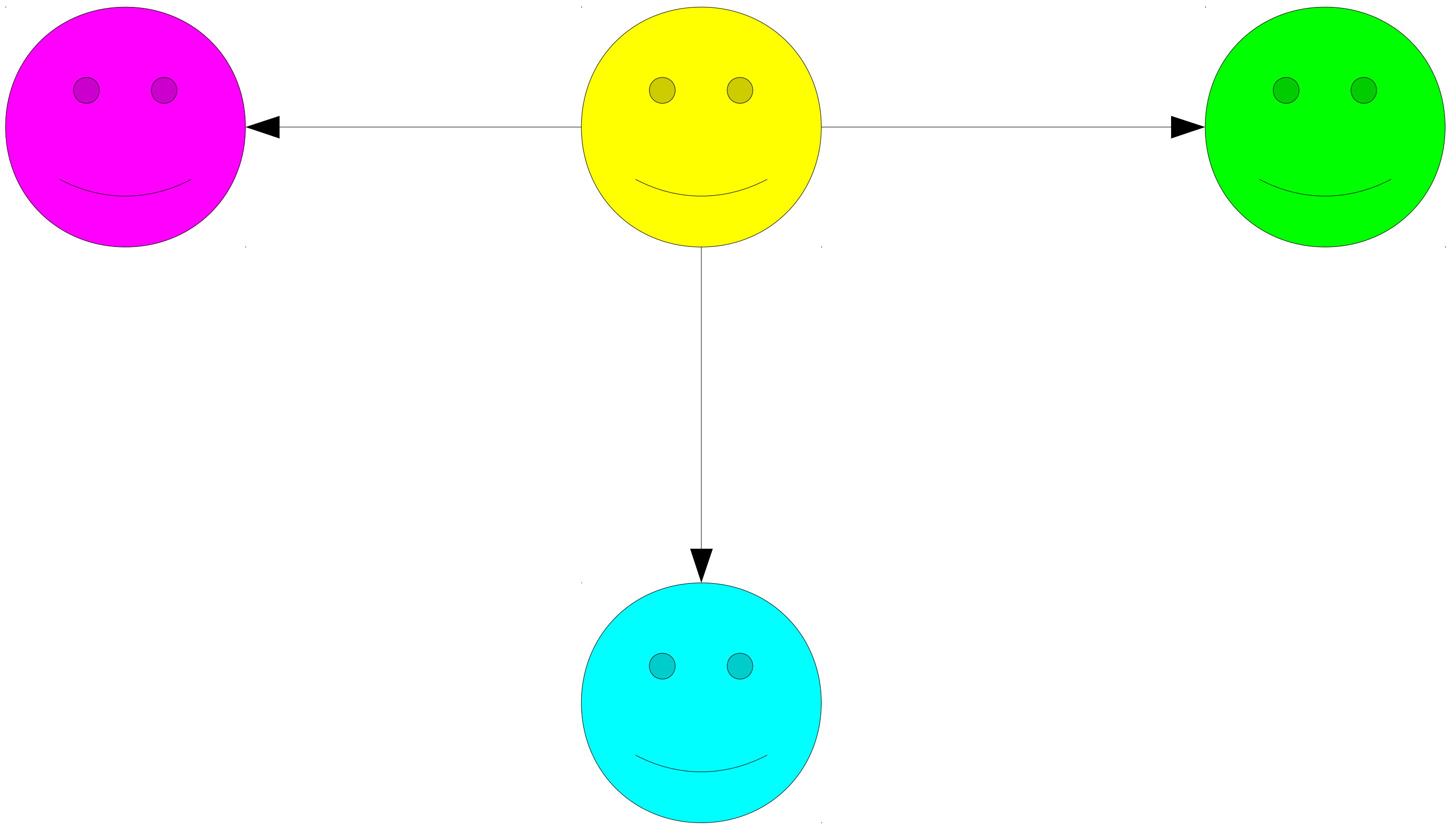
Irreflexivity

- 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:

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

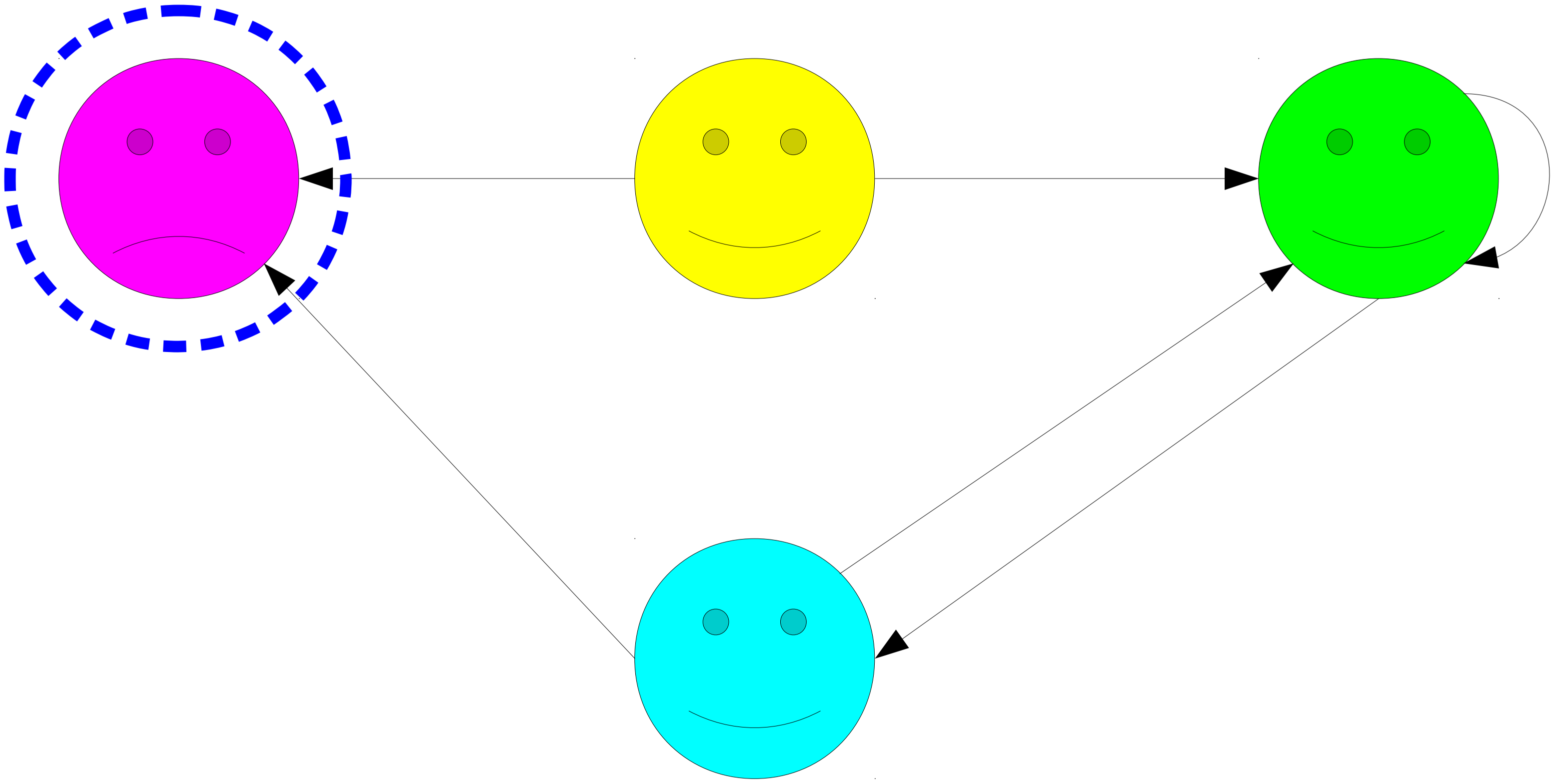
(*"No element is related to itself."*)

Irreflexivity Visualized

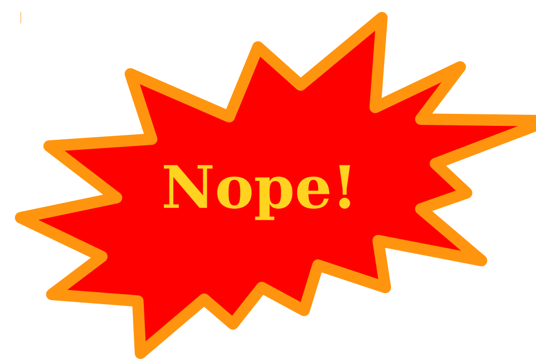


$$\forall a \in A.\ a \not R a$$

(*“No element is related to itself.”*)

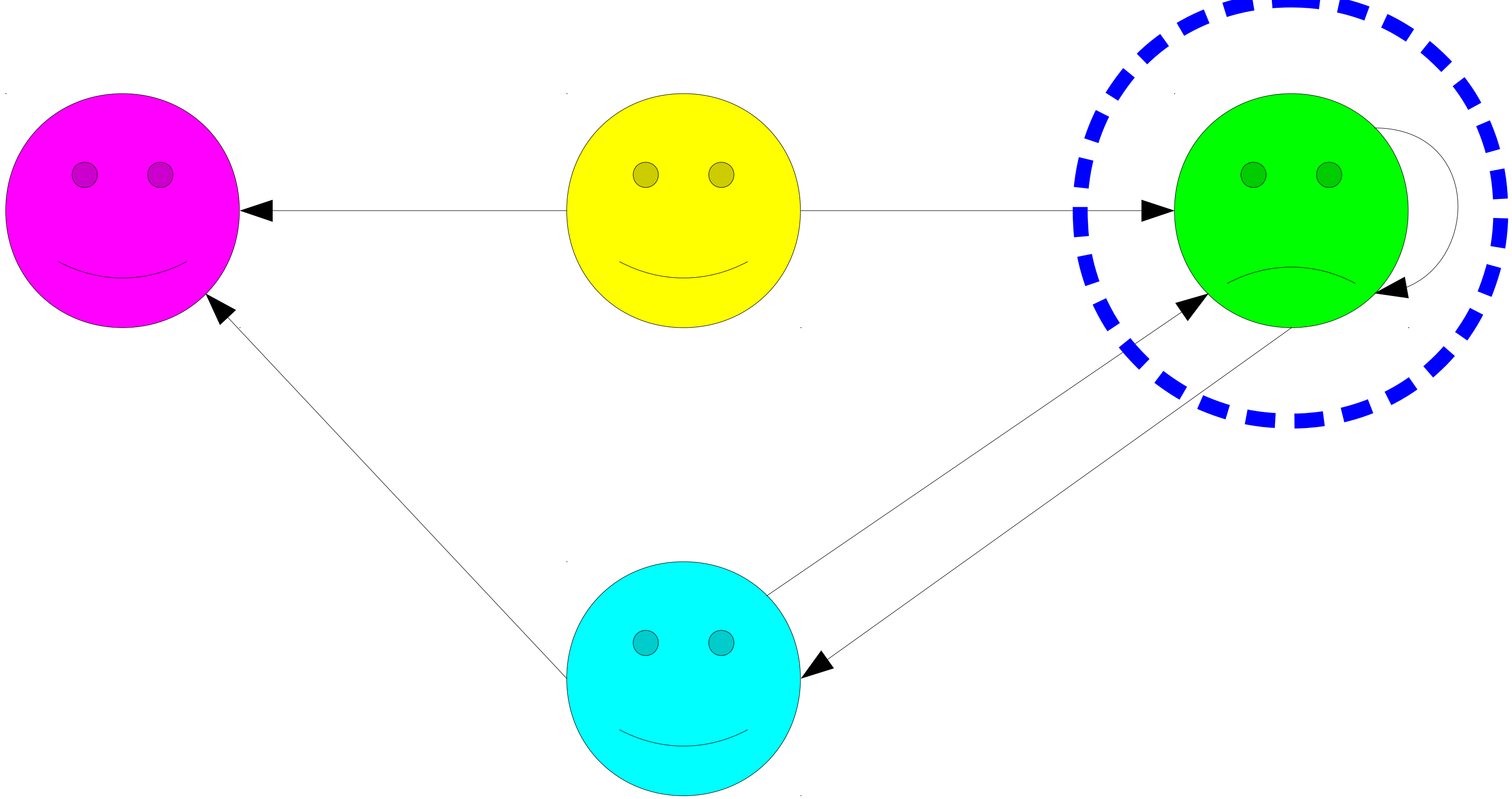


Is this relation reflexive?



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

(*"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?

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

Irreflexivity

Transitivity

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

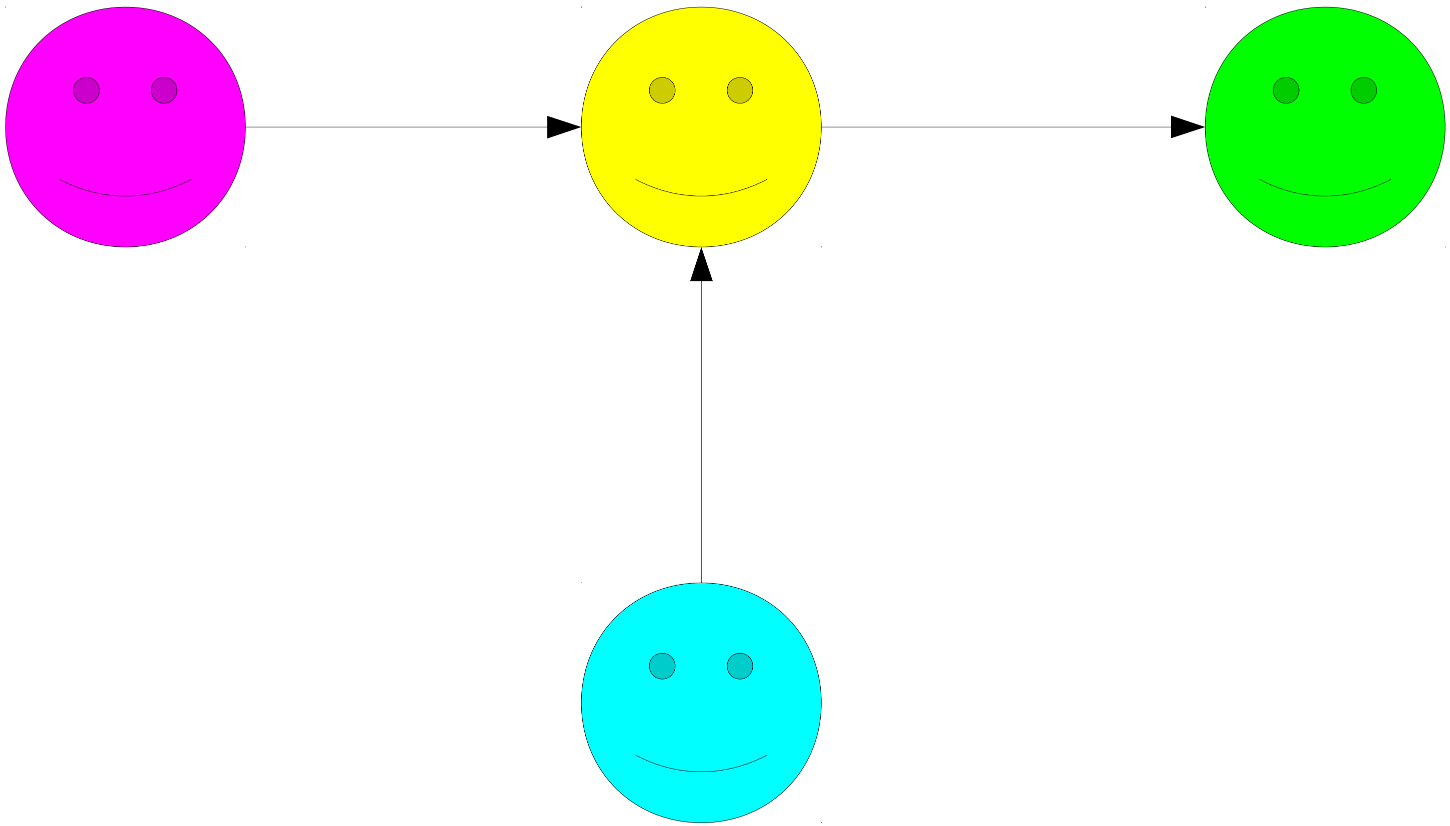
Asymmetry

- 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



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

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

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

Irreflexivity

Transitivity

Asymmetry

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).

Strict Order Proofs

- Let's suppose that you're asked to prove that a binary relation is a strict order.
- Calling back to the definition, you could prove that the relation is asymmetric, irreflexive, and transitive.
- However, there's a slightly easier approach we can use instead.

Theorem: Let R be a binary relation over a set A. If R is asymmetric, then R is irreflexive.

Proof: Let R be an arbitrary asymmetric binary relation over a set A. We will prove that R is irreflexive.

To do so, we will proceed by contradiction. Suppose that R is not irreflexive. That means that there must be some $x \in A$ such that xRx.

Since R is asymmetric, we know for any $a, b \in A$ that if aRb holds, then $b \not R a$ holds. Plugging in $a{=}x$ and $b{=}x$, we see that if xRx holds, then $x \not R x$ holds. We know by assumption that xRx is true, so we conclude that $x \not R x$ holds. However, this is impossible, since we can't have both xRx and $x \not R x$.

We have reached a contradiction, so our assumption must have been wrong. Thus R must be irreflexive. ■

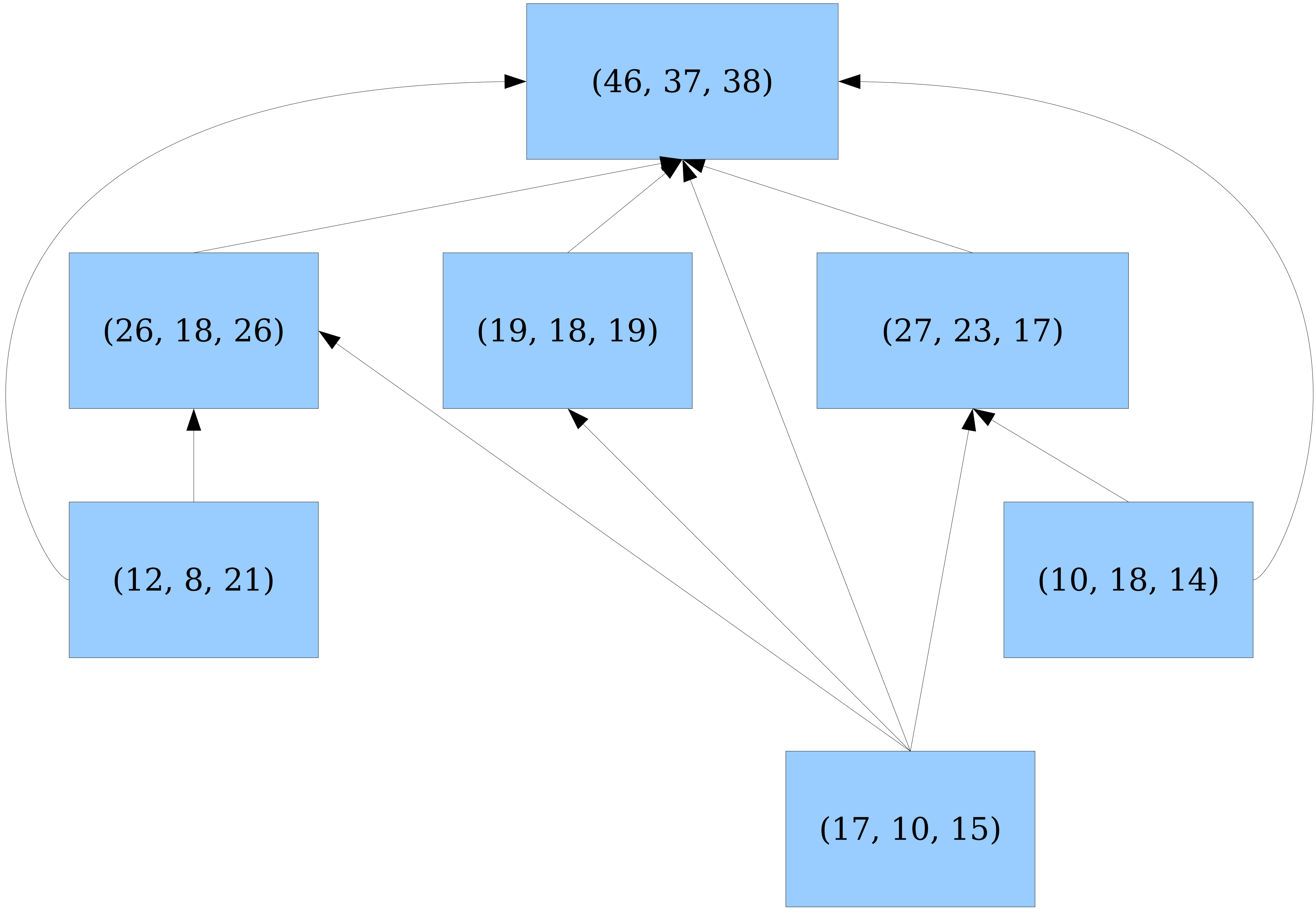
Theorem: If a binary relation R is asymmetric and transitive, then R is a strict order.

Proof: Let R be a binary relation that is asymmetric and transitive. Since R is asymmetric, by our previous theorem we know that R is also irreflexive. Therefore, R is asymmetric, irreflexive, and transitive, so by definition R is a strict order. ■

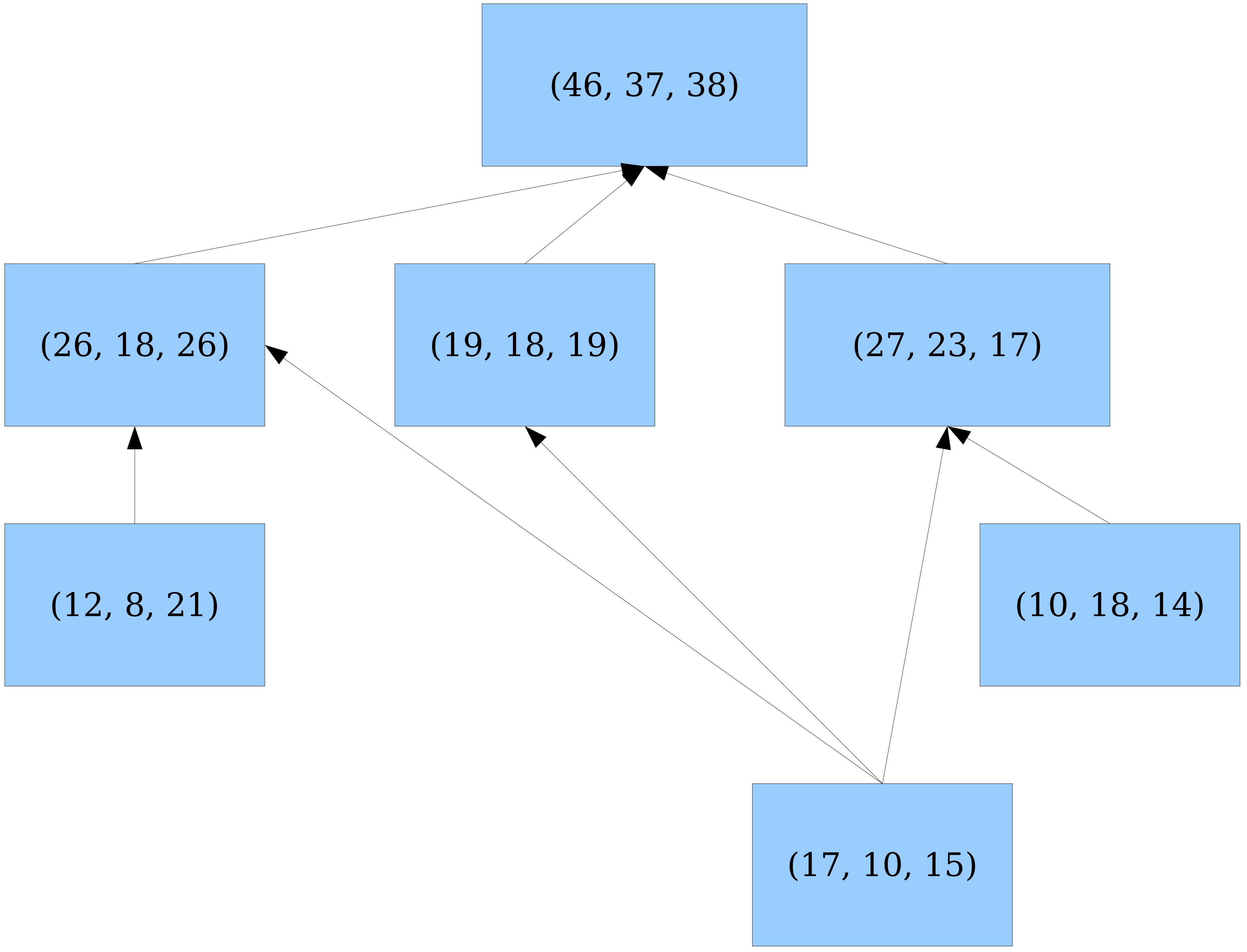
To prove that some binary relation R is a strict order, you can just prove that R is asymmetric and transitive. In the next problem set, you'll see an even simpler technique!

Drawing Strict Orders

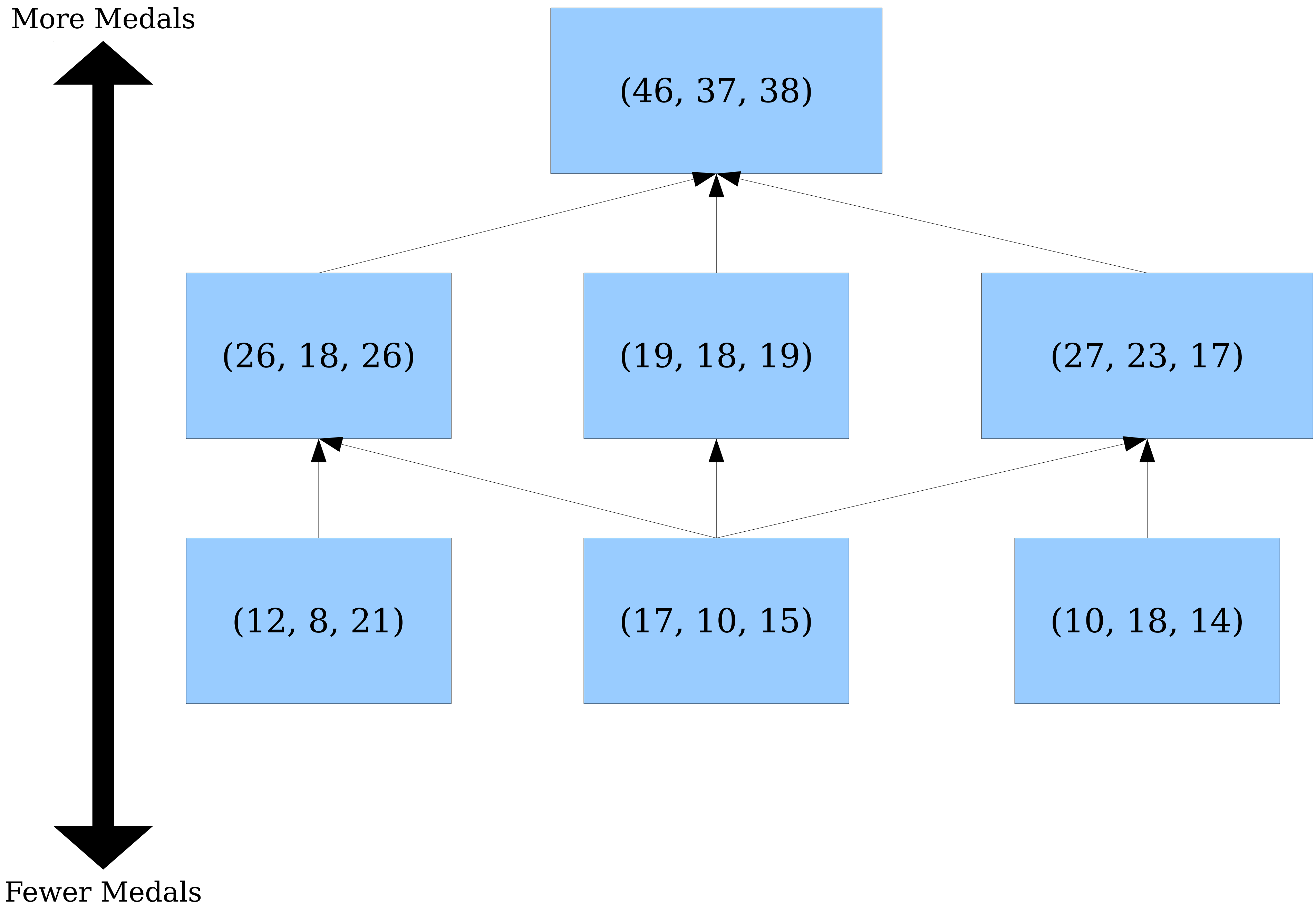
Gold	Silver	Bronze
46	37	38
27	23	17
26	18	26
19	18	19
17	10	15
12	8	21
10	18	14
9	3	9
8	12	8
8	11	10
8	7	4
8	3	4
7	6	6
7	4	6
6	6	1
6	3	2



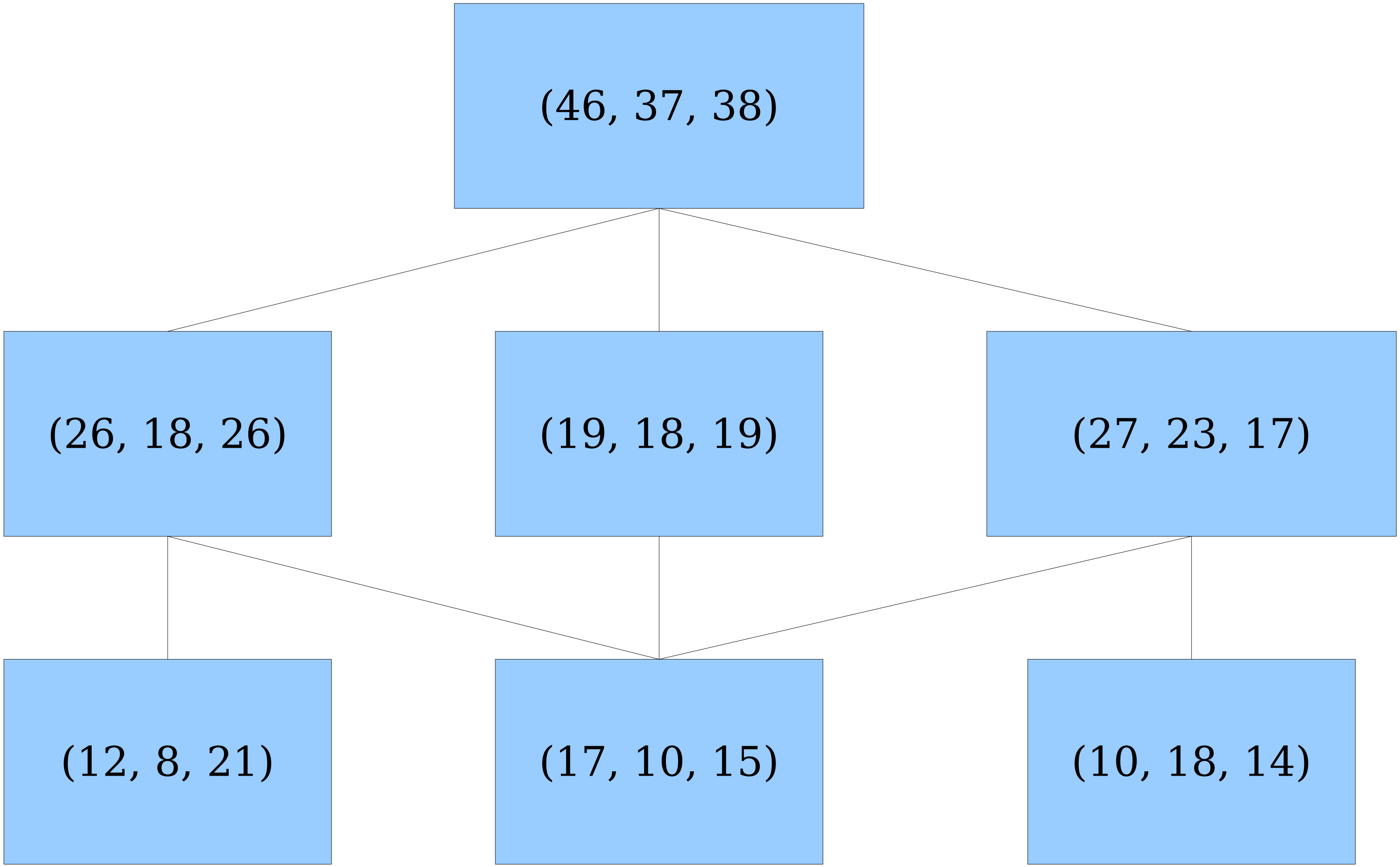
$$(g_1, s_1, b_1)\ R\ (g_2, s_2, b_2) \qquad \text{if} \qquad g_1 < g_2 \ \wedge \ s_1 < s_2 \ \wedge \ b_1 < b_2$$



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



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

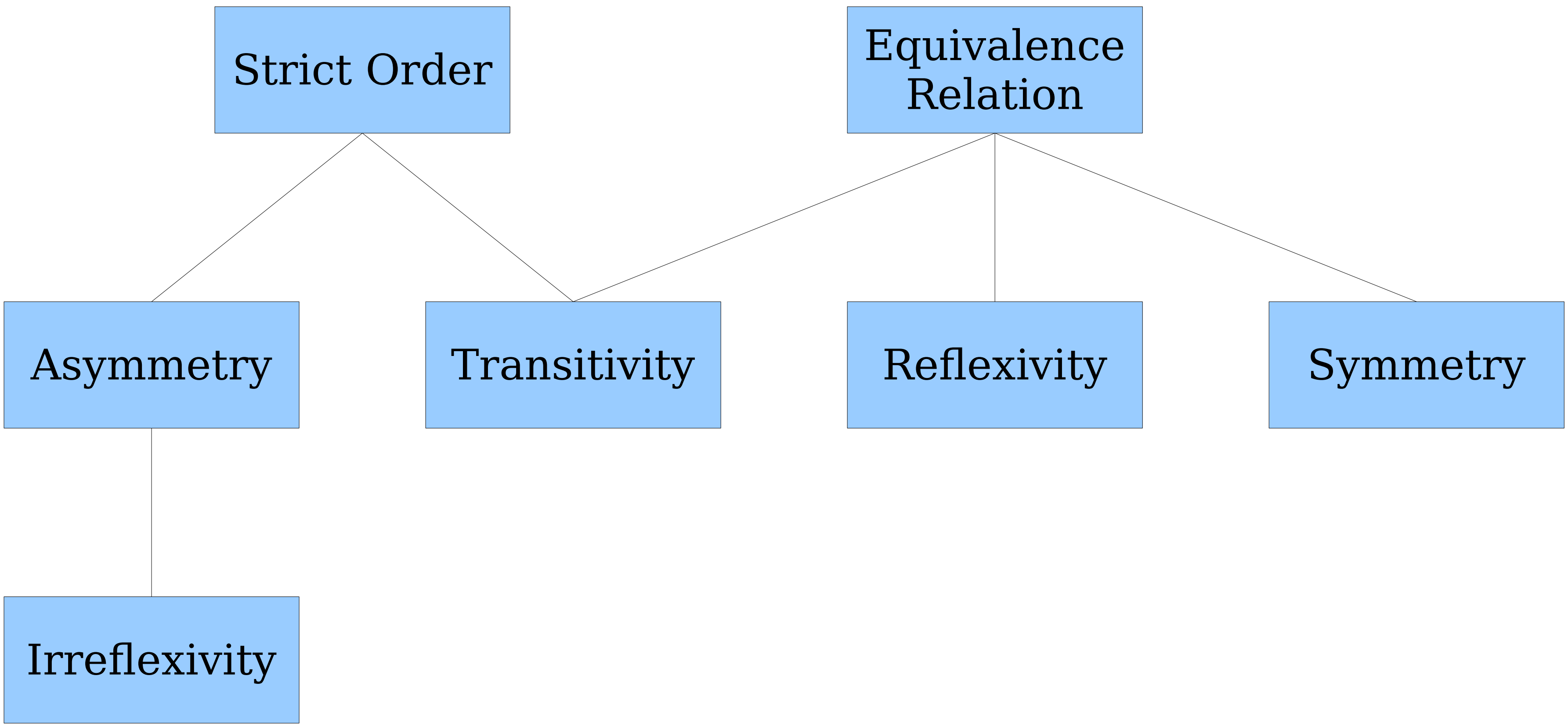


$$(g_1, s_1, b_1)\ R\ (g_2, s_2, b_2) \quad \text{if} \quad g_1 < g_2 \ \wedge\ s_1 < s_2 \ \wedge\ b_1 < b_2$$

Hasse Diagrams

- A ***Hasse diagram*** is a graphical representation of a strict order.
- Elements are drawn from bottom-to-top.
- 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.

The Meta Strict Order



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

The Binary Relation Editor