Binary Relations

Part Two

Outline for Today

- ***Recap from Last Time***
  - Where are we, again?
- ***A Fundamental Theorem***
  - What do equivalence relations *do*?
- ***Strict Orders***
  - Representing prerequisites.
- ***Hasse Diagrams***
  - Drawing prerequisite diagrams.

Recap from Last Time

Binary Relations

- A ***binary relation over a set A*** is a predicate R that can be applied to ordered 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: $3 = 3$, and $5 < 7$, 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: $4 \neq 3$, and $4 \not< 3$, and $\mathbb{N} \not\subseteq \emptyset$.

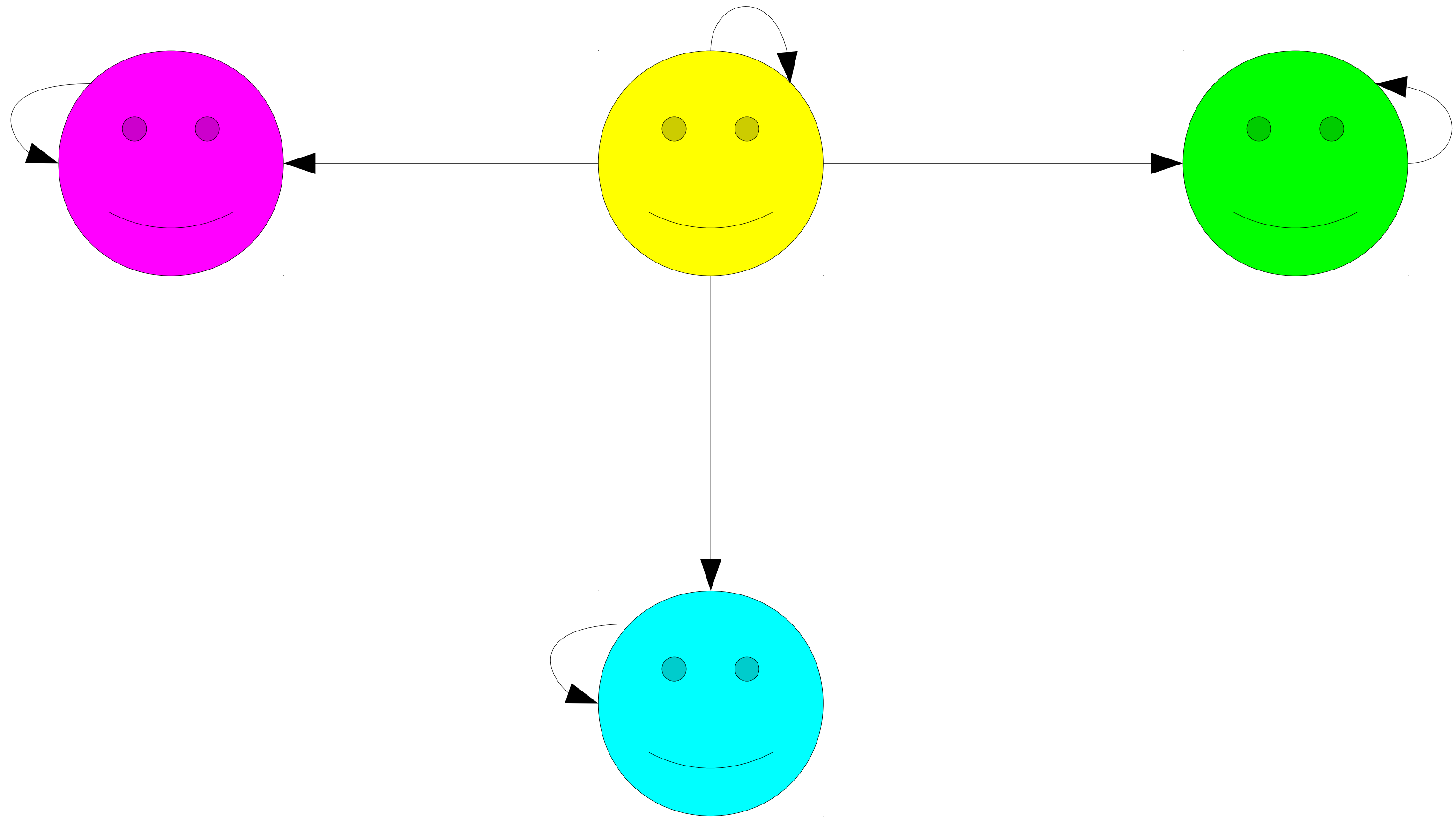
Reflexivity

- 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 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.")

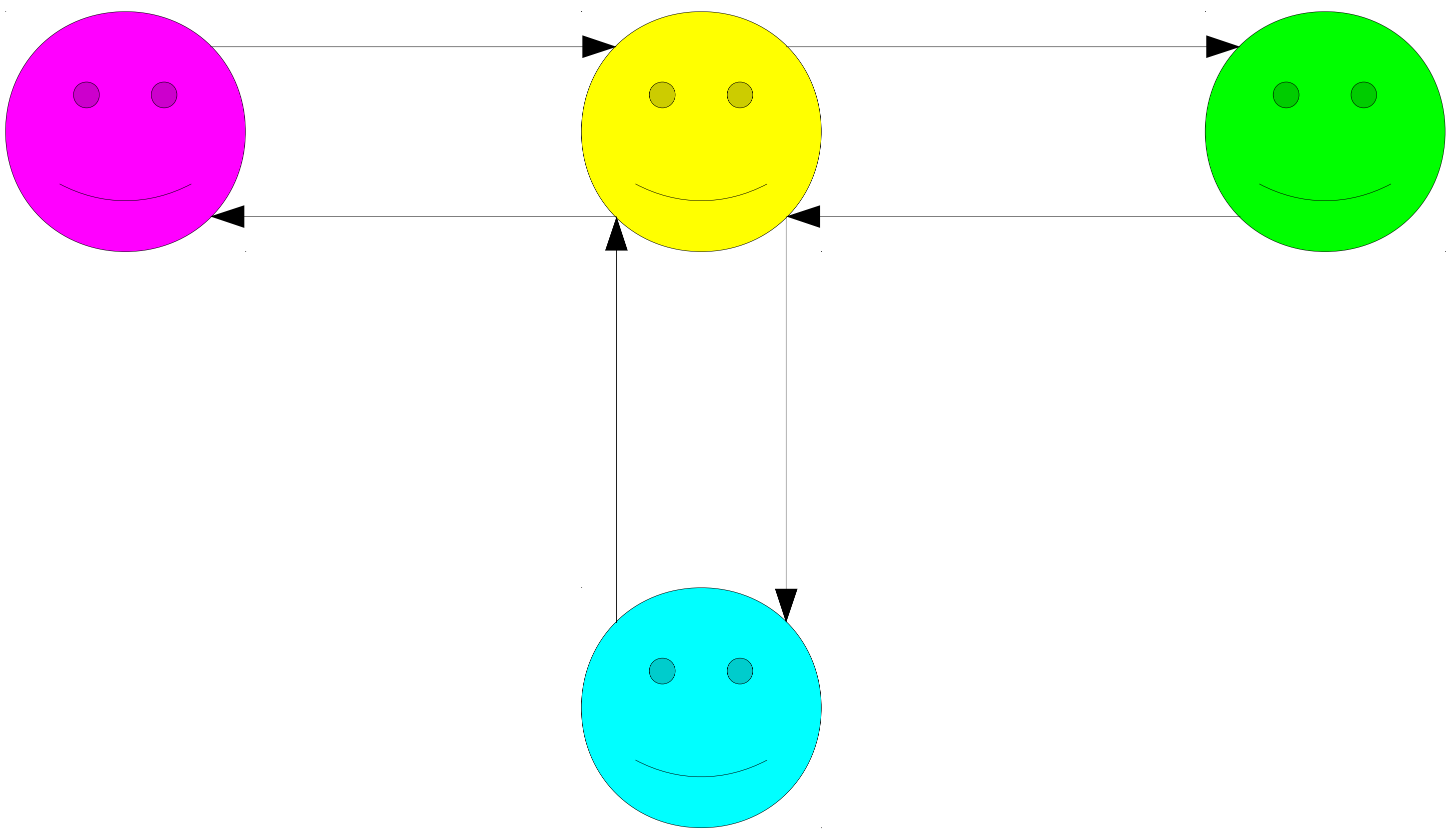
Symmetry

- 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."*)

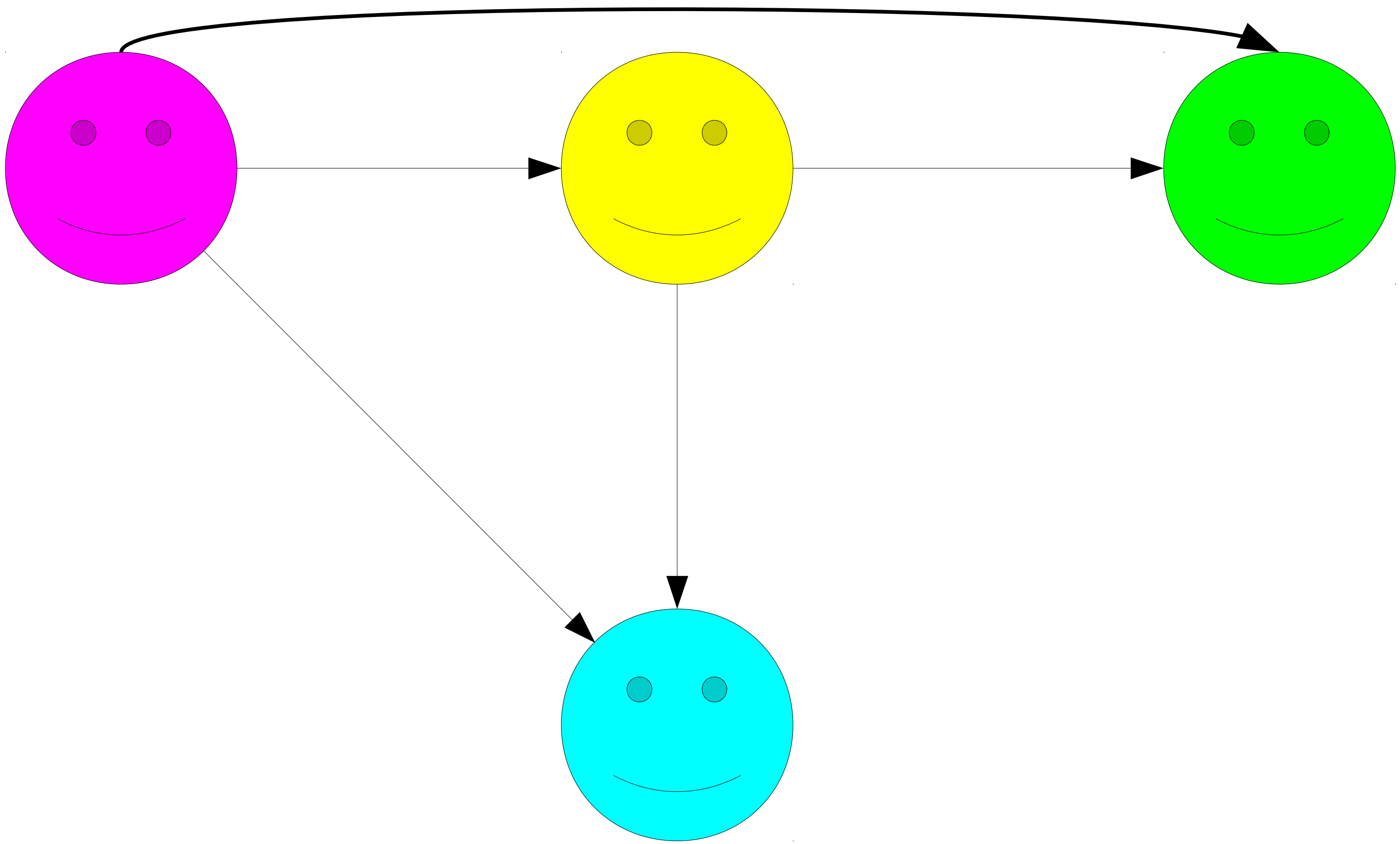
Transitivity

- 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



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

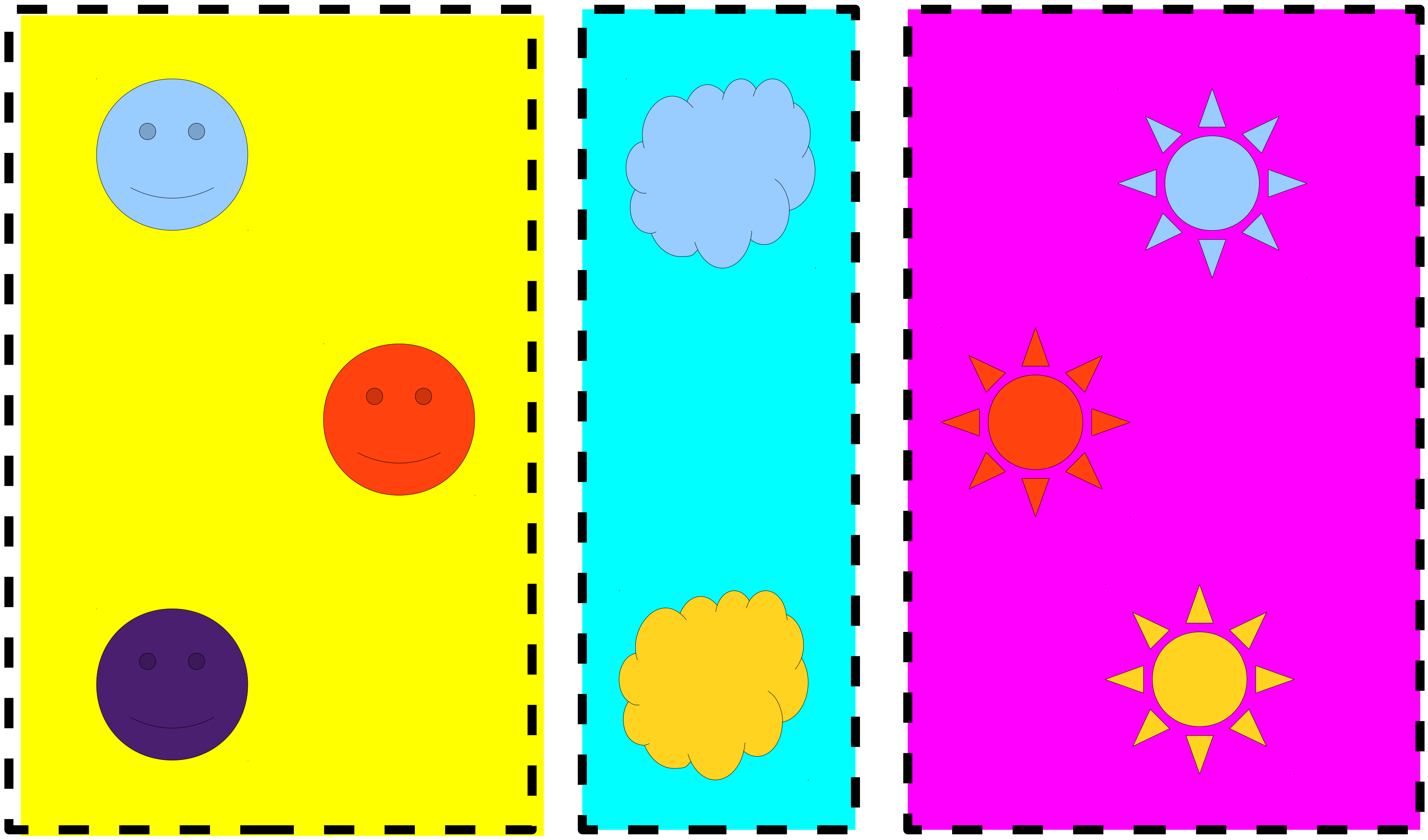
(*"Whenever a is related to b and b is related to c, we know a is related to c.*)

Equivalence Relations

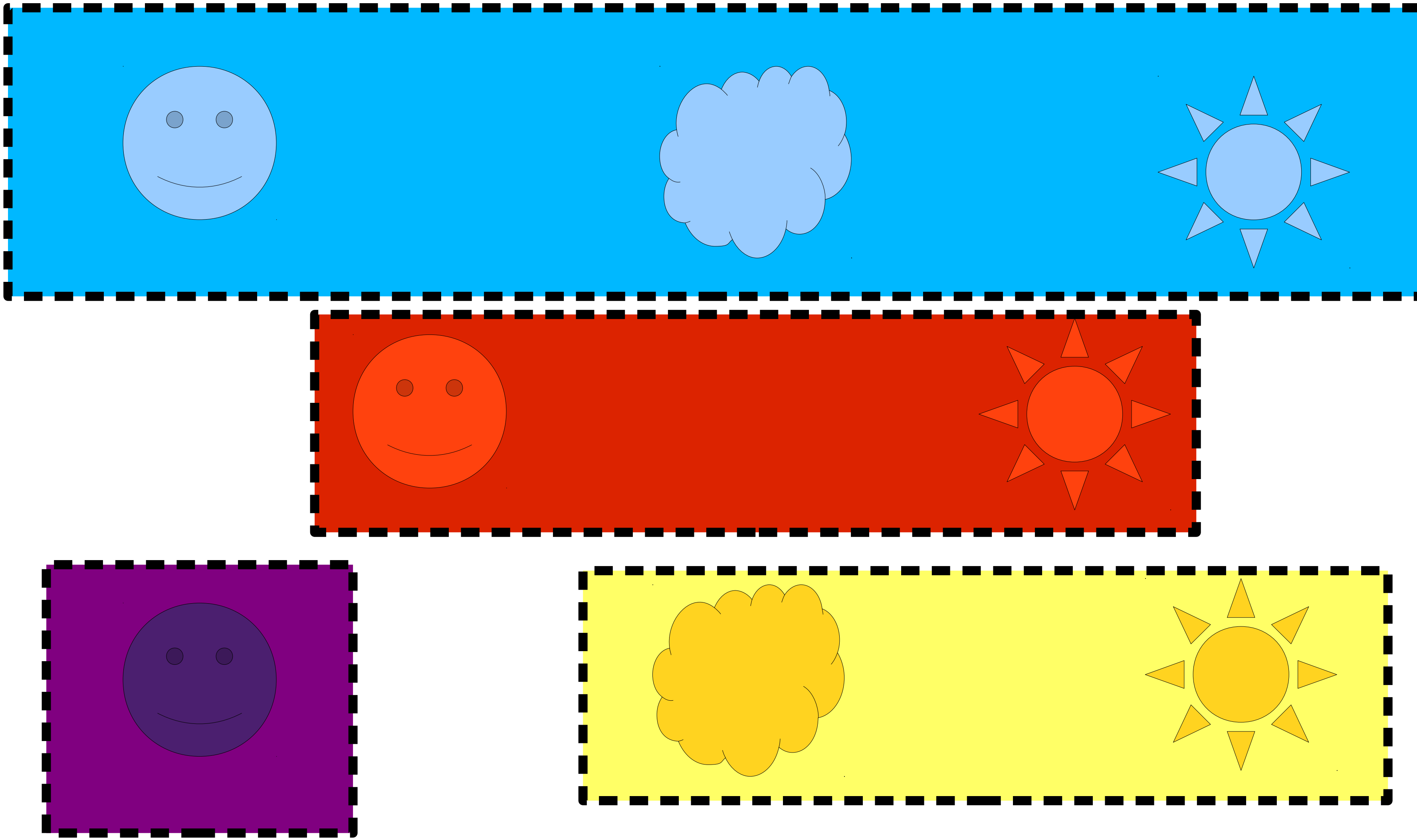
- An ***equivalence relation*** is a relation that is reflexive, symmetric and transitive.
- Some examples:
  - $x = y$
  - $x \equiv_k y$
  - x has the same color as y
  - x has the same shape as y.

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

Notice that $[0]_{\equiv_3} = [3]_{\equiv_3}$. These are *literally* the same sets, 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.

Proving the Theorem

- The FToER says that if R is an equivalence relation over a set A, then every $a \in A$ belongs to exactly one equivalence class of A.
- To prove this, we will show the following:
  - Every $a \in A$ belongs to *at least* one equivalence class of A.
  - Every $a \in A$ belongs to *at most* one equivalence class of A.

Lemma 1: Let R be an arbitrary equivalence relation over a set A. Then for any $a \in A$, the element a belongs to at least one equivalence class of R.

Proof: Let R be an arbitrary equivalence relation over a set A and choose any $a \in A$. Since R is an equivalence relation, it's reflexive, so we know that aRa. Therefore, by definition of $[a]_R$, we see that $a \in [a]_R$, so we see that a belongs to at least one equivalence class of R, as required. ■

Lemma 2: Let R be an arbitrary equivalence relation over a set A. Then for any $a \in A$, the element a belongs to at most one equivalence class of R.

Proof: Let R be an arbitrary equivalence relation over a set A and choose any $a \in A$. To show that a belongs to at most one equivalence class of R, suppose that $a \in [b]_R$ and $a \in [c]_R$. We will prove that $[b]_R = [c]_R$.

This is a general technique for proving there is at most one object with some property – assume you have two objects with the same property, then show that they're really the same object.

Hypothetically speaking, this might be relevant for the question about uniqueness on the problem set. ☺

Lemma 2: Let R be an arbitrary equivalence relation over a set A. Then for any $a \in A$, the element a belongs to at most one equivalence class of R.

Proof: Let R be an arbitrary equivalence relation over a set A and choose any $a \in A$. To show that a belongs to at most one equivalence class of R, suppose that $a \in [b]_R$ and $a \in [c]_R$. We will prove that $[b]_R = [c]_R$. To do so, we will show that $[b]_R \subseteq [c]_R$ and that $[c]_R \subseteq [b]_R$.

This is a really useful technique for showing that two sets are equal!

Lemma 2: Let R be an arbitrary equivalence relation over a set A. Then for any $a \in A$, the element a belongs to at most one equivalence class of R.

Proof: Let R be an arbitrary equivalence relation over a set A and choose any $a \in A$. To show that a belongs to at most one equivalence class of R, suppose that $a \in [b]_R$ and $a \in [c]_R$. We will prove that $[b]_R = [c]_R$. To do so, we will show that $[b]_R \subseteq [c]_R$ and that $[c]_\mathrm{R} \subseteq [b]_R$.

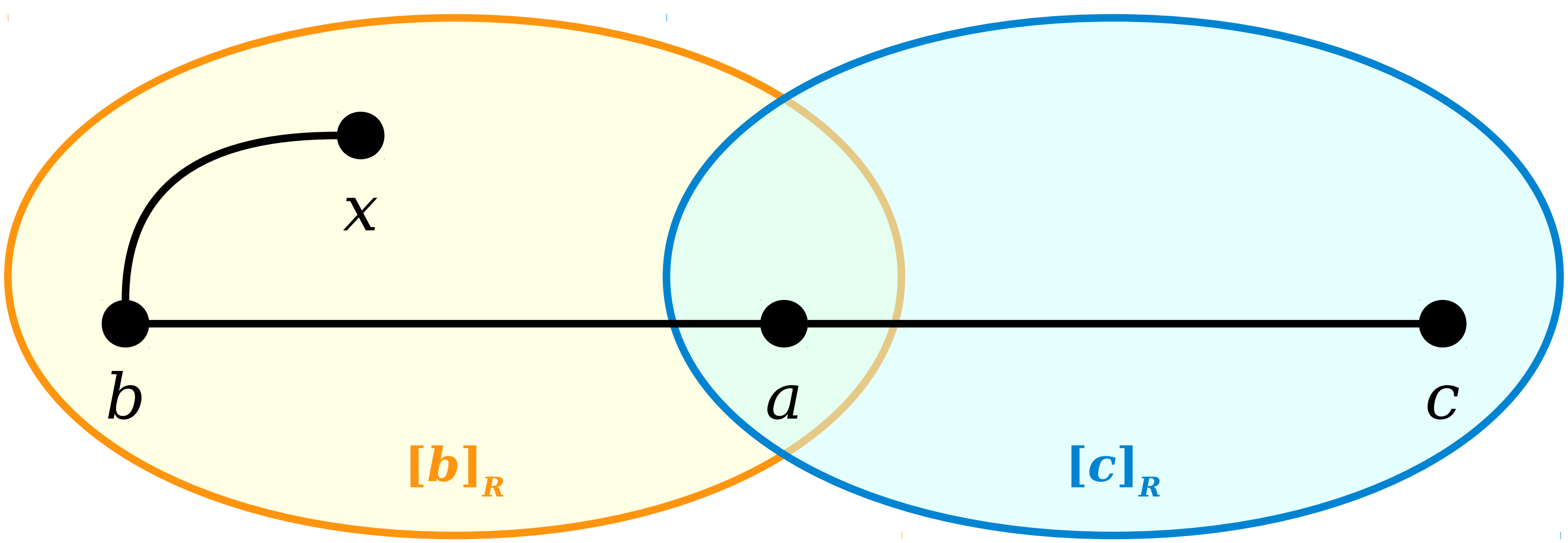
First, we'll prove $[b]_R \subseteq [c]_R$. Consider any $x \in [b]_R$. This means that bRx. We need to prove that $x \in [c]_R$, meaning that we need to show that cRx holds.

Remember all that time we spent talking about doing proofs on subsets? All those lessons we learned still apply here!

Lemma 2: Let R be an arbitrary equivalence relation over a set A. Then for any $a \in A$, the element a belongs to at most one equivalence class of R.

Proof: Let R be an arbitrary equivalence relation over a set A and choose any $a \in A$. To show that a belongs to at most one equivalence class of R, suppose that $a \in [b]_R$ and $a \in [c]_R$. We will prove that $[b]_R = [c]_R$. To do so, we will show that $[b]_R \subseteq [c]_R$ and that $[c]_\mathrm{R} \subseteq [b]_R$.

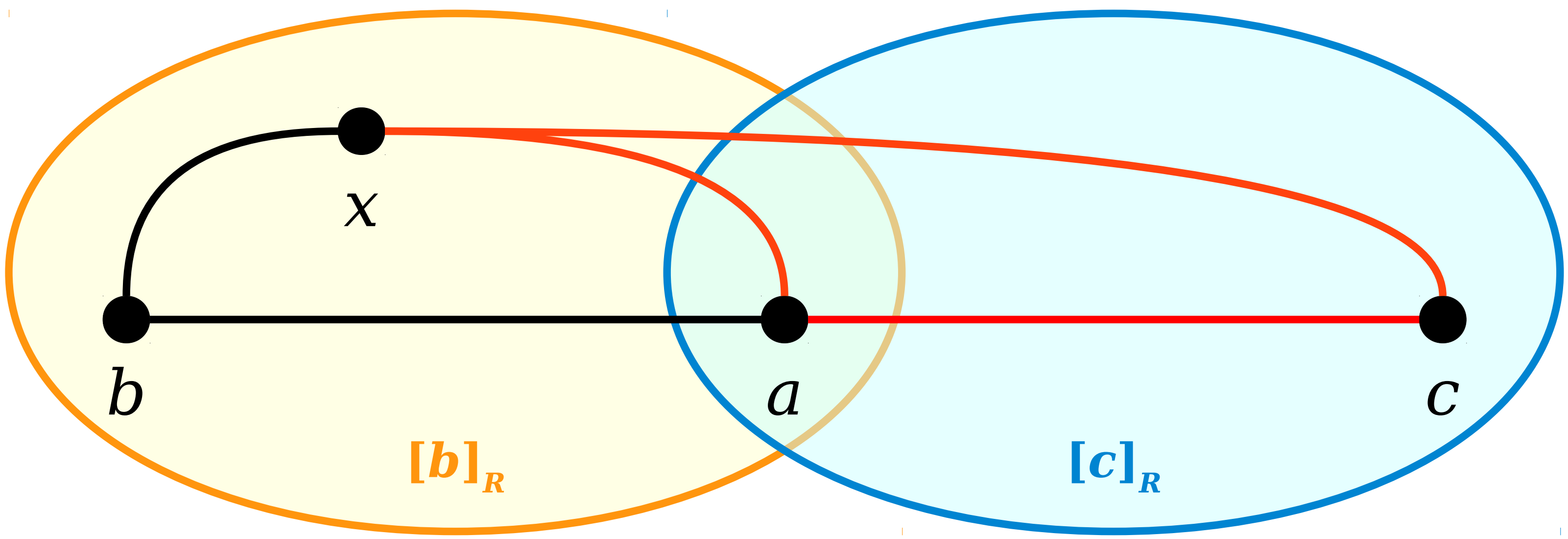
First, we'll prove $[b]_R \subseteq [c]_R$. Consider any $x \in [b]_R$. This means that bRx. We need to prove that $x \in [c]_R$, meaning that we need to show that cRx holds.



Lemma 2: Let R be an arbitrary equivalence relation over a set A. Then for any $a \in A$, the element a belongs to at most one equivalence class of R.

Proof: Let R be an arbitrary equivalence relation over a set A and choose any $a \in A$. To show that a belongs to at most one equivalence class of R, suppose that $a \in [b]_R$ and $a \in [c]_R$. We will prove that $[b]_R = [c]_R$. To do so, we will show that $[b]_R \subseteq [c]_R$ and that $[c]_R \subseteq [b]_R$.

First, we'll prove $[b]_R \subseteq [c]_R$. Consider any $x \in [b]_R$. This means that bRx. We need to prove that $x \in [c]_R$, meaning that we need to show that cRx holds.



Lemma 2: Let R be an arbitrary equivalence relation over a set A. Then for any $a \in A$, the element a belongs to at most one equivalence class of R.

Proof: Let R be an arbitrary equivalence relation over a set A and choose any $a \in A$. To show that a belongs to at most one equivalence class of R, suppose that $a \in [b]_R$ and $a \in [c]_R$. We will prove that $[b]_R = [c]_R$. To do so, we will show that $[b]_R \subseteq [c]_R$ and that $[c]_\mathrm{R} \subseteq [b]_R$.

First, we'll prove $[b]_R \subseteq [c]_R$. Consider any $x \in [b]_R$. This means that bRx. We need to prove that $x \in [c]_R$, meaning that we need to show that cRx holds.

Earlier, we said that $a \in [b]_R$. As a result, we know that bRa. Since R is symmetric and we know that bRa, we conclude that aRb. Then, since aRb and bRx, we know that aRx via transitivity. Similarly, we mentioned earlier that $a \in [c]_R$, which means that cRa. Then, since cRa and aRx, by transitivity we see that cRx. This is what we needed to show to conclude that $[b]_R \subseteq [c]_R$.

The proof that $[c]_\mathrm{R} \subseteq [b]_R$ is identical to the above, with the roles of b and c interchanged. ■

Equivalences and Partitions

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

  Partition of Elements ⇒ Equivalence Relation

- The Fundamental Theorem of Equivalence Relations shows that the reverse direction holds as well!

  Equivalence Relation ⇒ Partition of Elements

Time-Out for Announcements!

Problem Set One

- Problem Set One has been graded. Feedback is available on GradeScope.
  - ***Please review all your feedback***. Our feedback is designed to help you learn how to write better proofs.
  - ***Make sure you understand everything we've written***. You don't want to keep making the same mistakes over and over again!
- Solutions are available online – *please be sure to review them!*

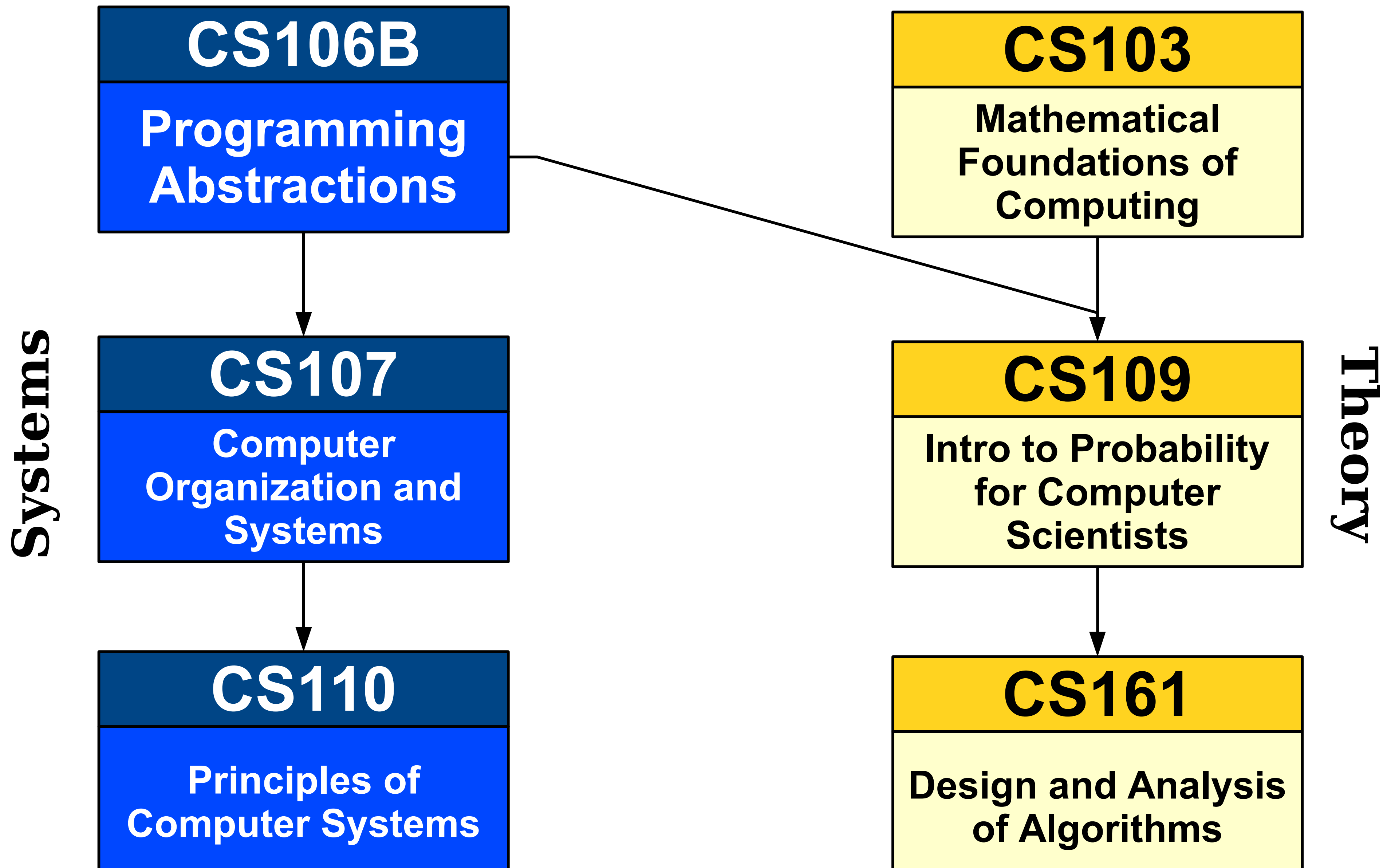
Problem Set Two

- Problem Set Two is due on Friday at 3:00PM.
  - Can use late days to extend that deadline to Monday at 3:00PM.
- Please ask questions on Piazza, in office hours, or to the staff mailing list! We're here to help out.

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

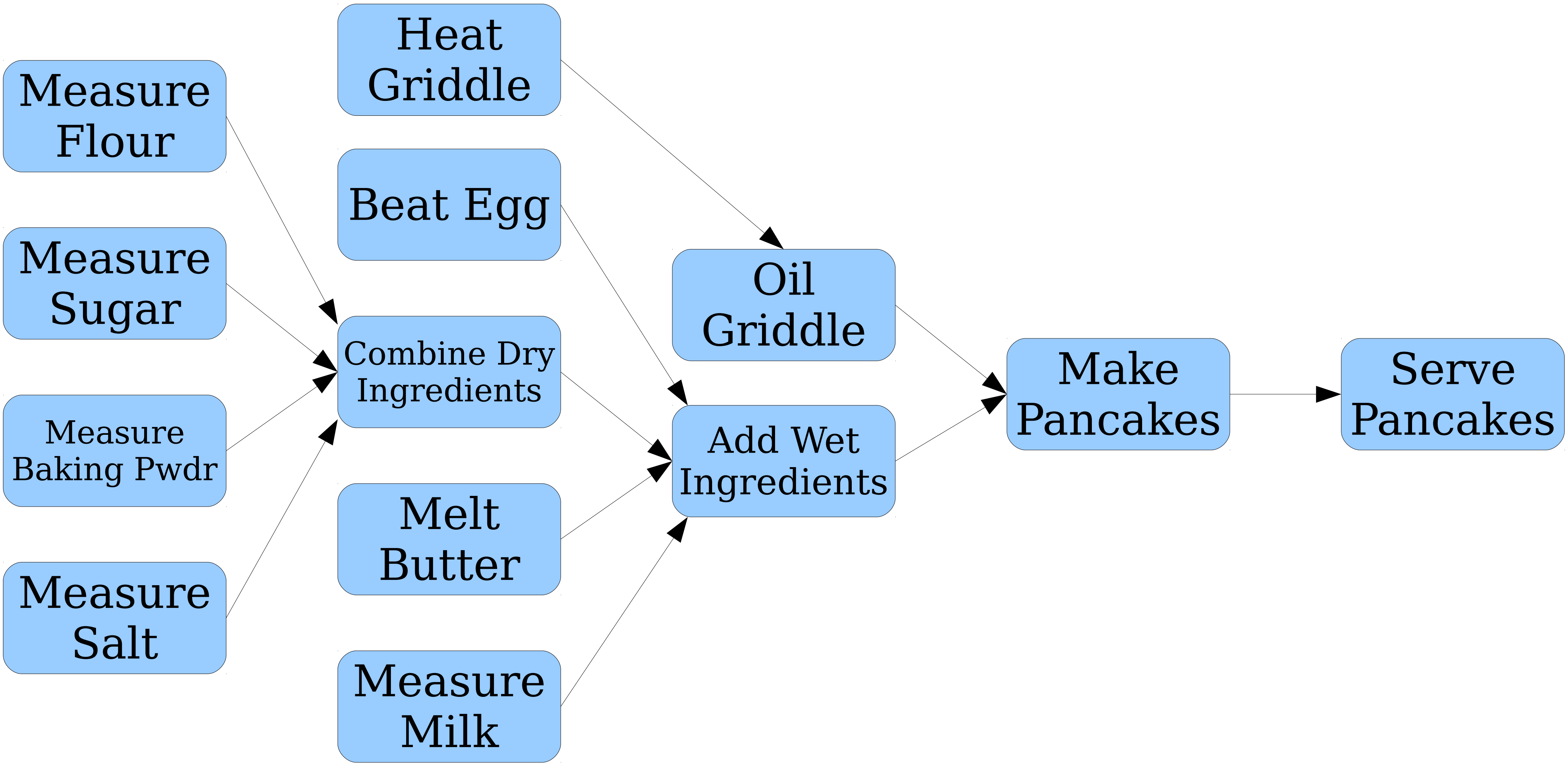
Everyone's got a pancake recipe. This one comes from Food Wishes (http://foodwishes.blogspot.com/2011/08/grandma-kellys-good-old-fashioned.html).

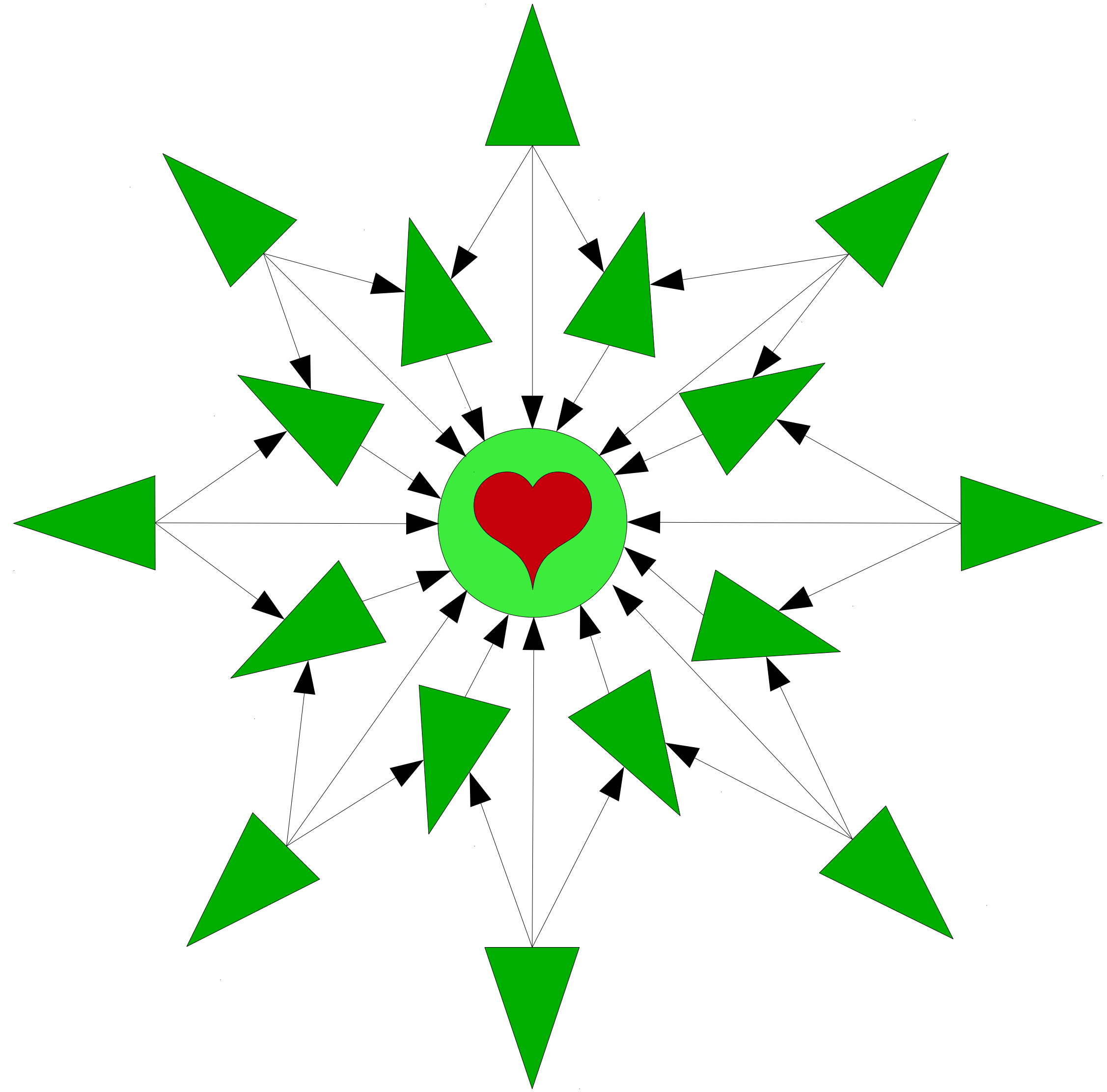
Ingredients

- 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

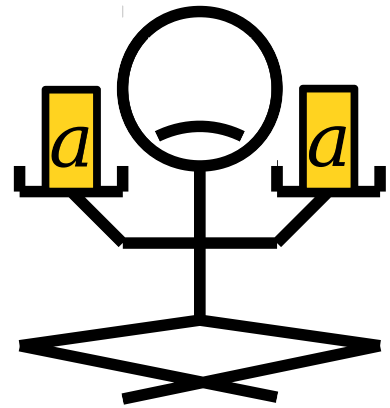


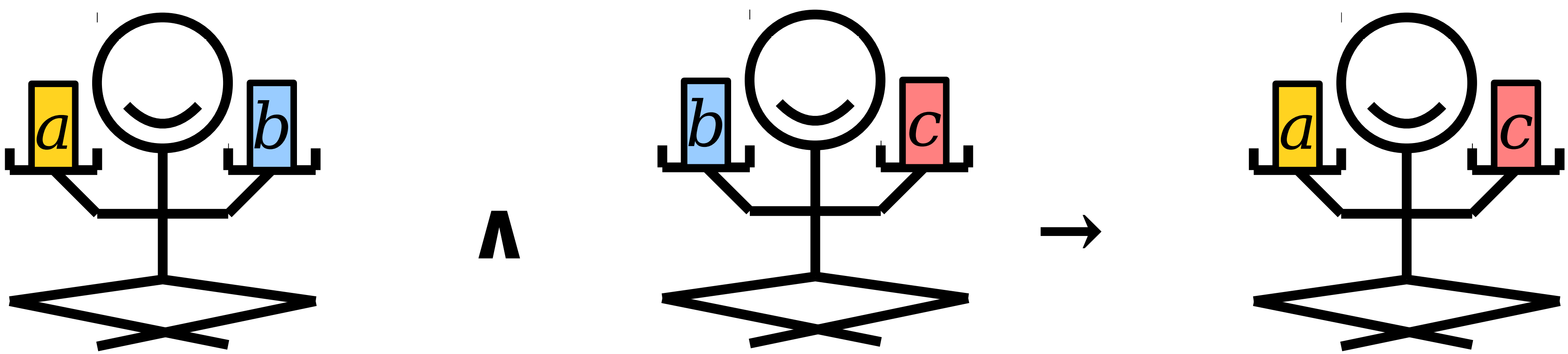
Relations and Prerequisites

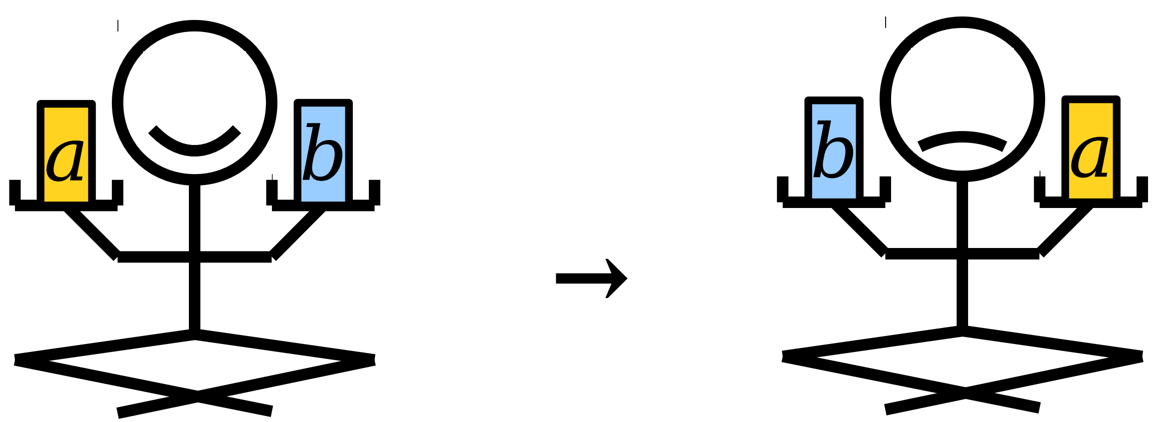
- Let's imagine that we have a prerequisite structure with no circular dependencies.
- We can think about a binary relation R where aRb means

“***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

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

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

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

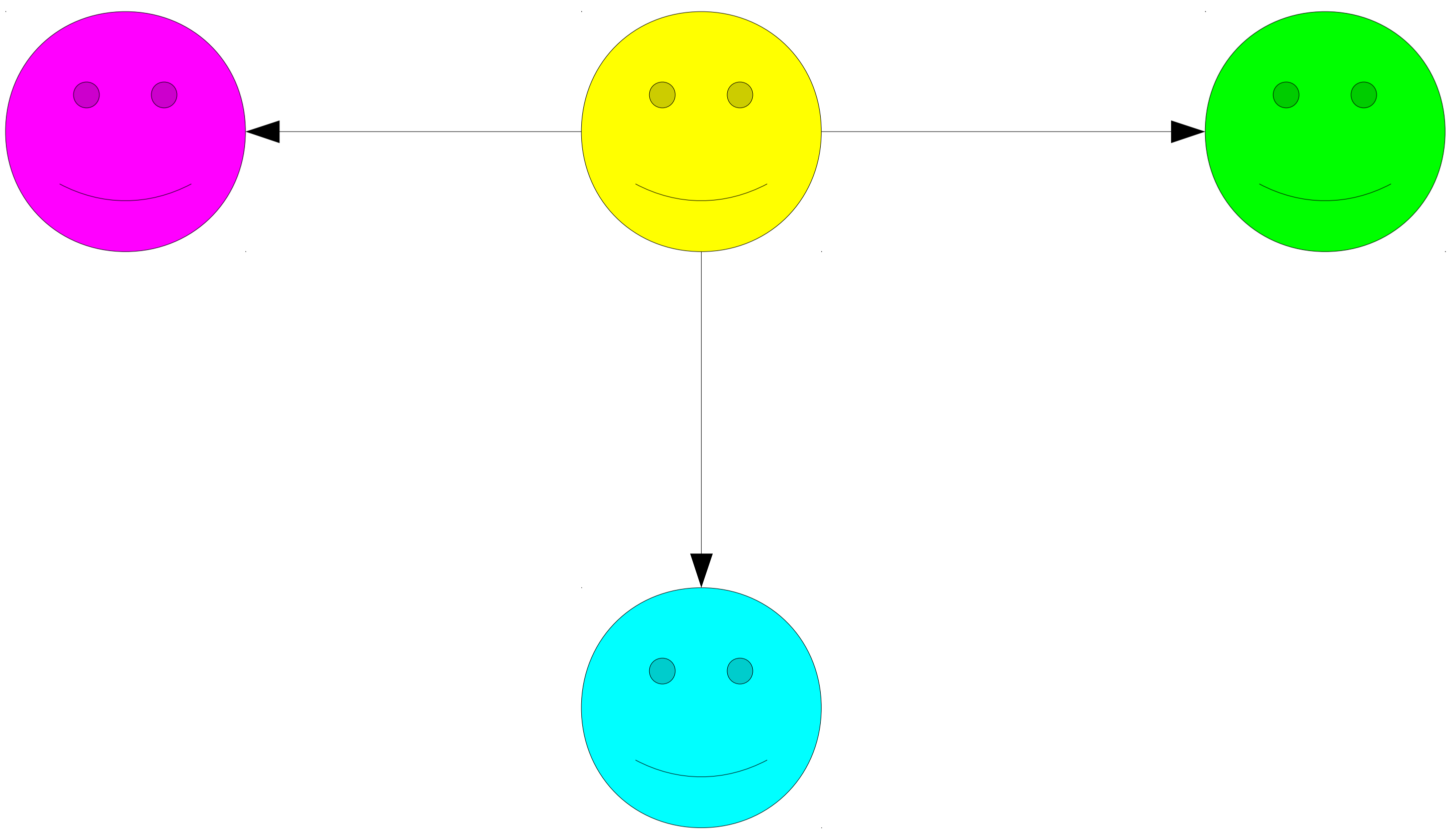
Irreflexivity

- 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:

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

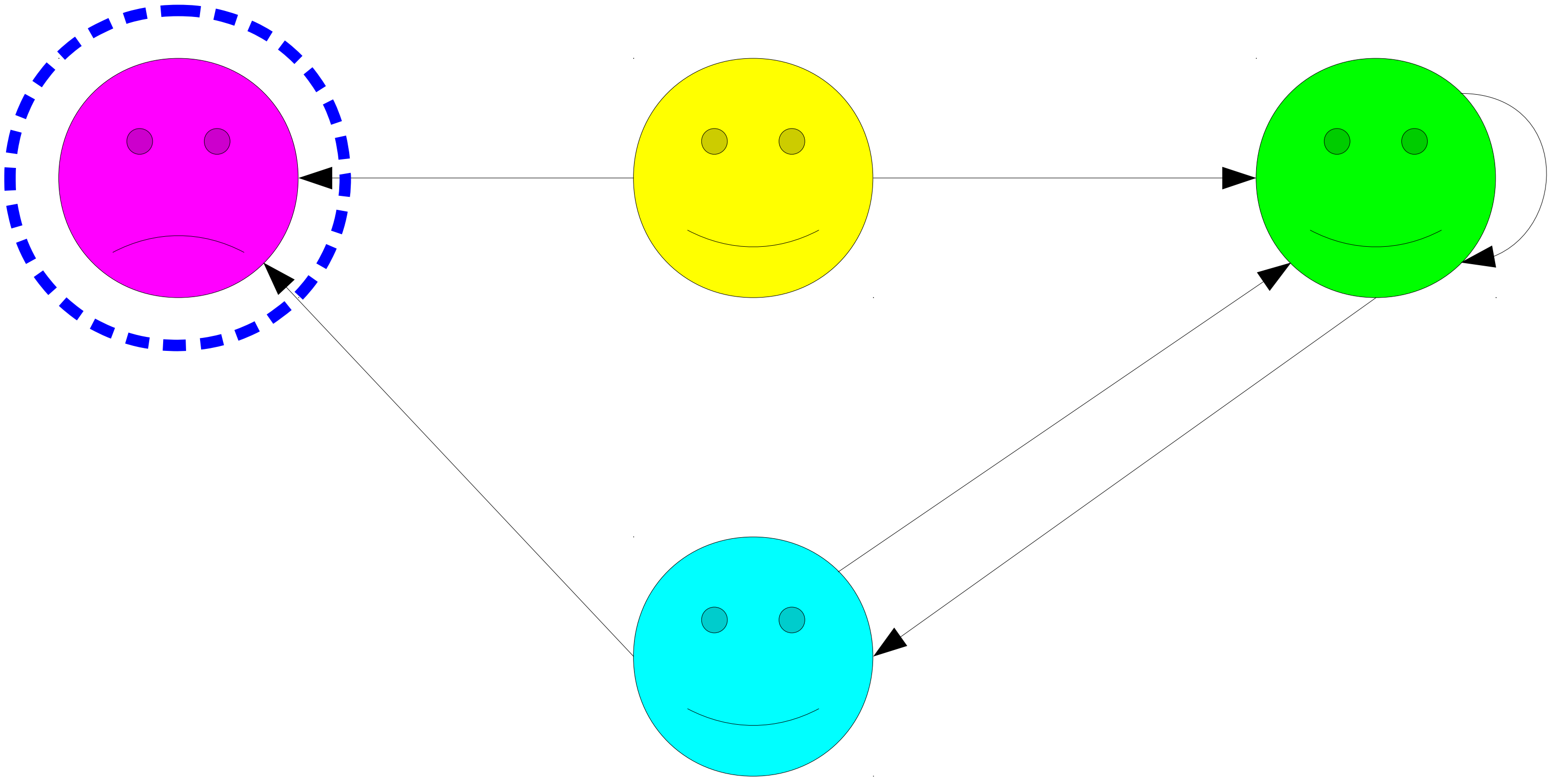
(*“No element is related to itself.”*)

Irreflexivity Visualized



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

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

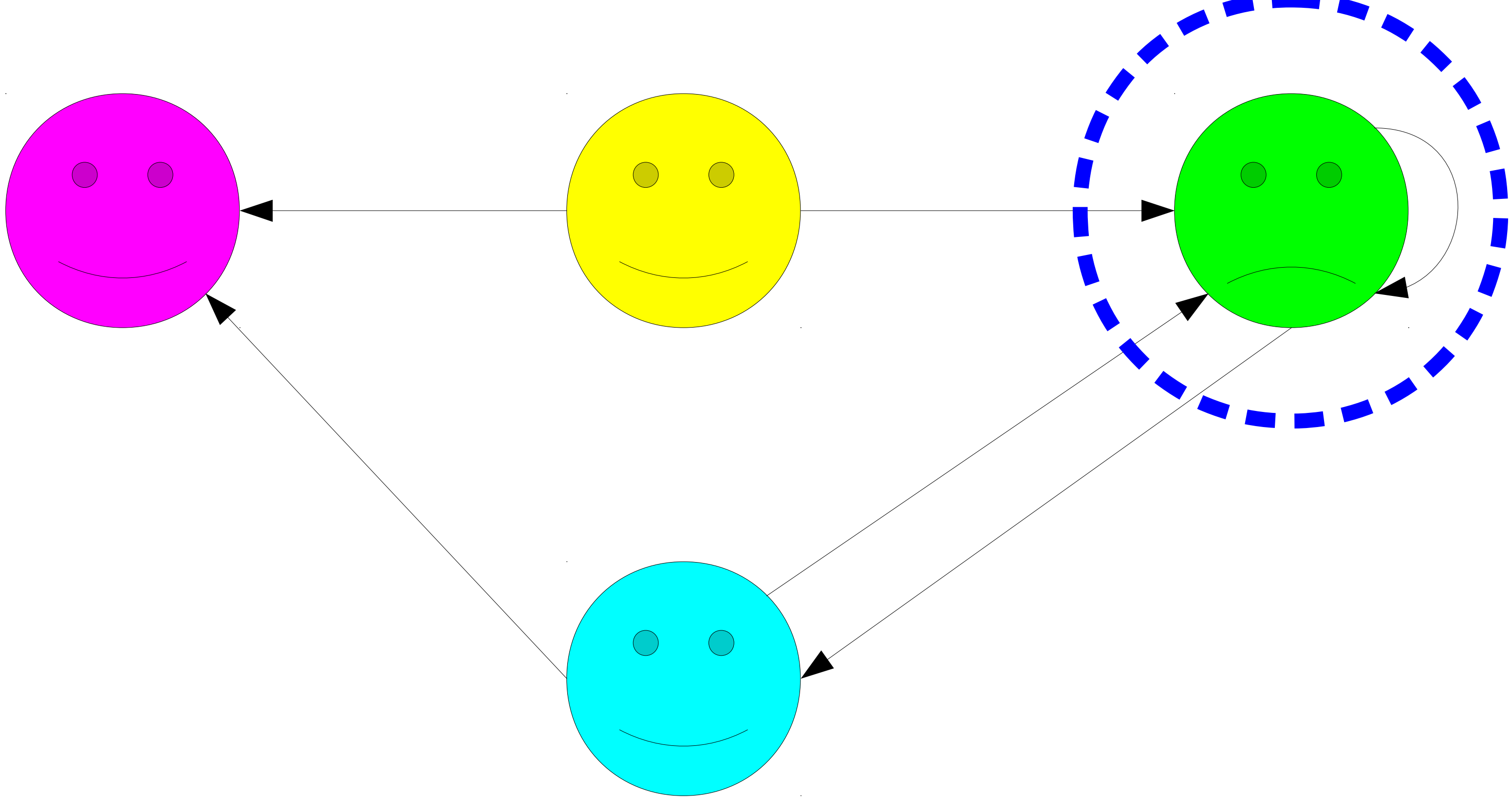


Is this relation reflexive?

Nope!

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

("Every element is related to itself.")



Is this relation irreflexive?

Nope!

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

(“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?

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

Bad Naming Choices

- The opposite of reflexive is “not reflexive.” This is not the same as “irreflexive.”
- The opposite of irreflexive is “not irreflexive.” This is not the same as “reflexive.”
- *Don't let the naming conventions trip you up!* It's unfortunate that these are the names we have, but, c'est la vie.

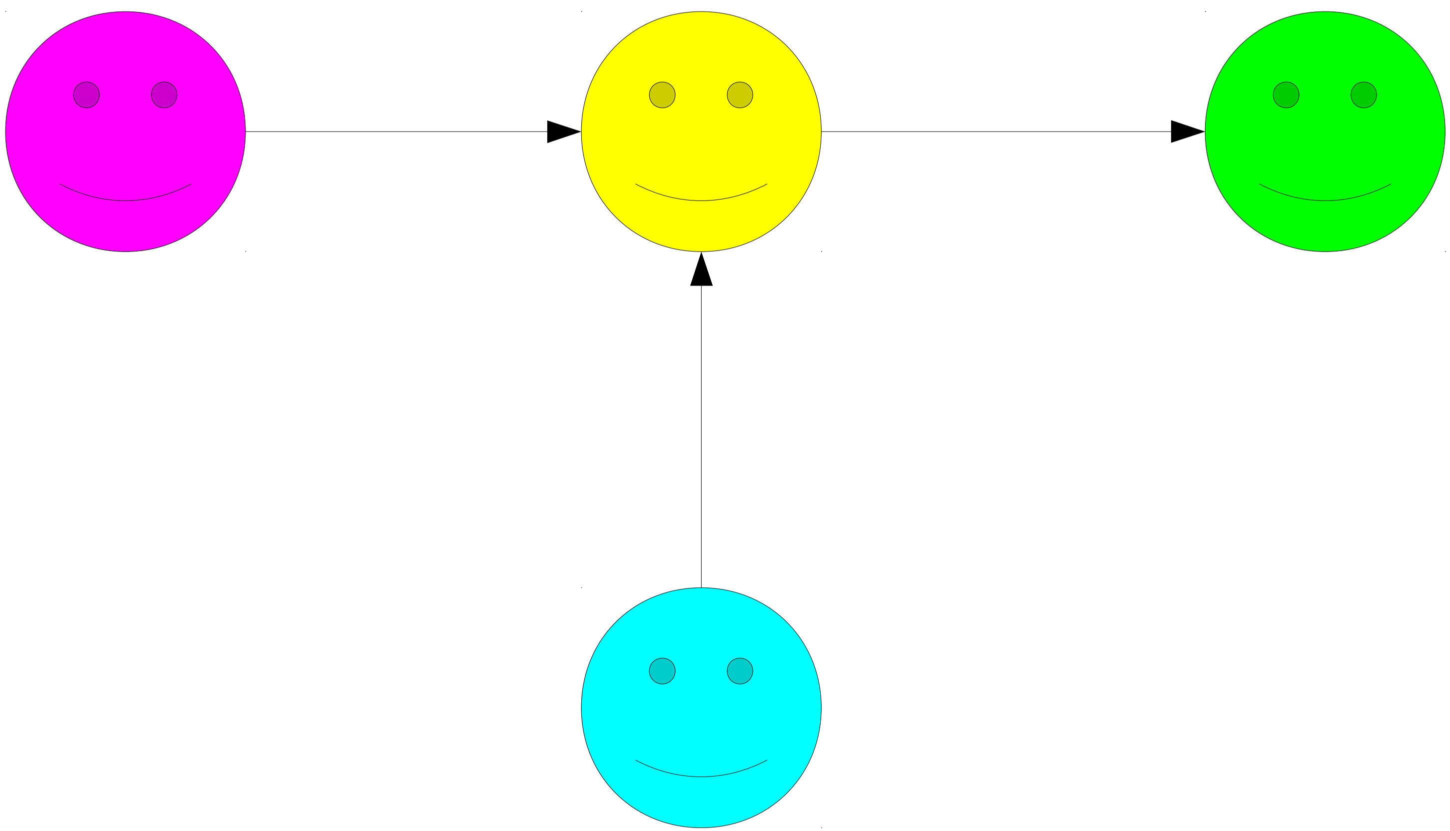
Asymmetry

- 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:

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

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

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?

Strict Orders

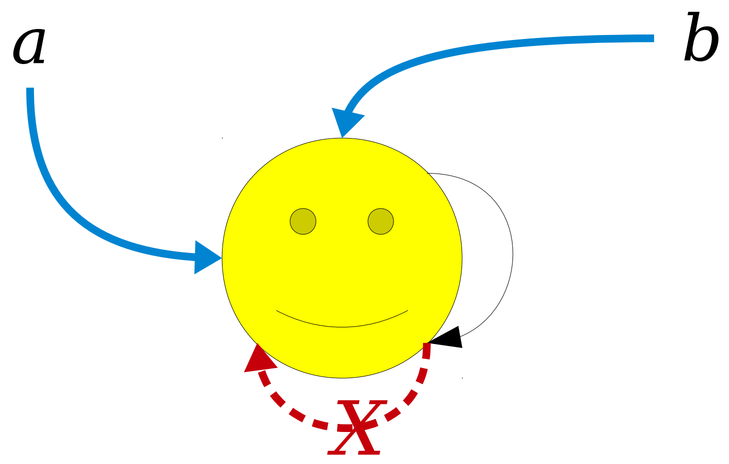
- A ***strict order*** is a relation that is irreflexive, asymmetric and transitive.
- Some examples:
  - $x < y$.
  - a can run faster than b.
  - $A \subsetneq B$ (that is, $A \subseteq B$ and $A \neq B$).
- Strict orders are useful for representing prerequisite structures and have applications in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

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

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



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

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

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.

What's the high-level structure of this proof?

$$\forall R.\ (\textit{Asymmetric}(R) \to \textit{Irreflexive}(R))$$

Therefore, we'll choose an arbitrary asymmetric relation R, then go and prove that R is irreflexive.

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.

What is the definition of irreflexivity?

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

What is the negation of this statement?

$$\exists x \in A.\ xRx$$

So let's suppose that there is some element $x \in A$ such that xRx and proceed from there.

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. ■

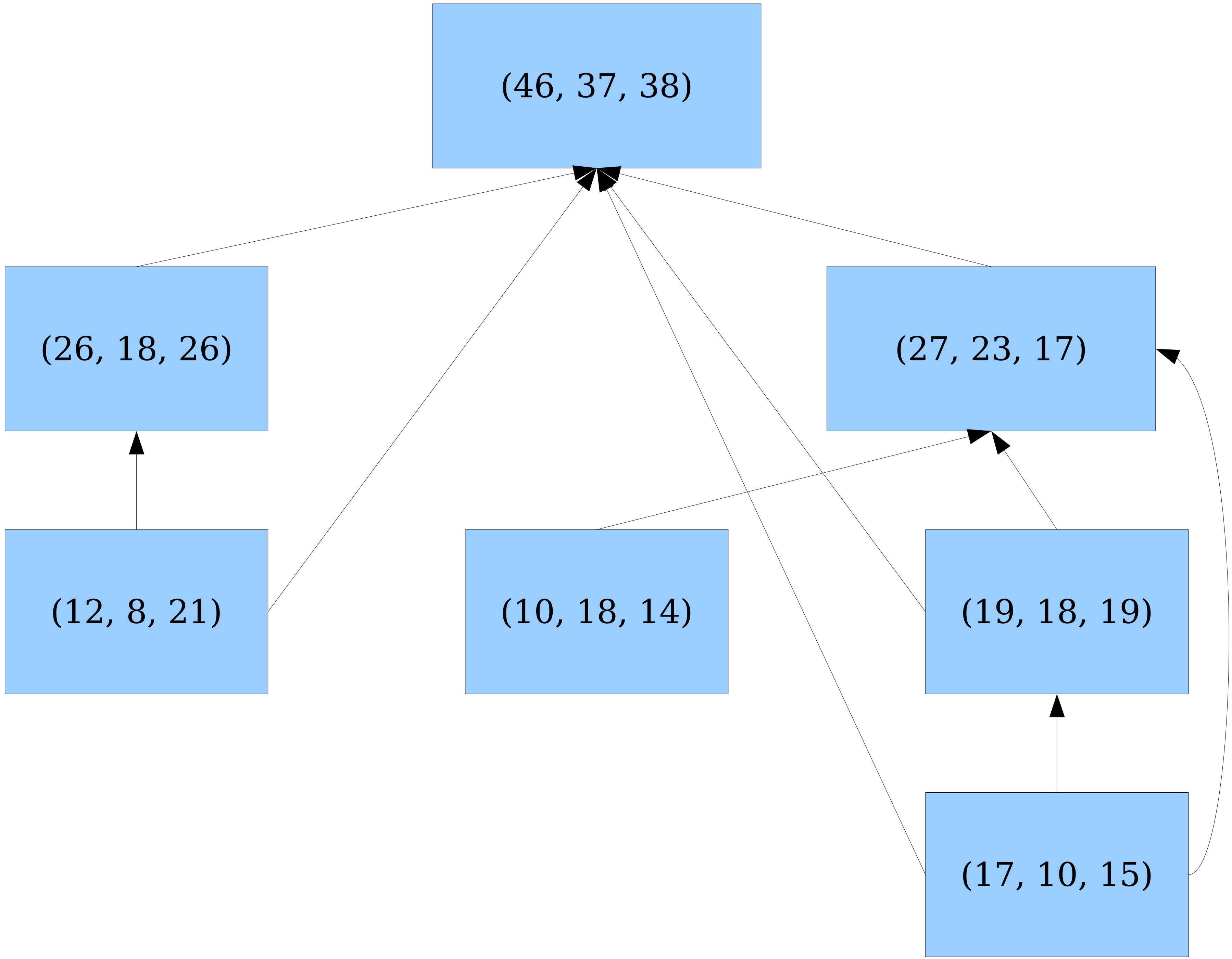
Theorem: If a binary relation R is asymmetric and transitive, then R is a strict order.

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. ■

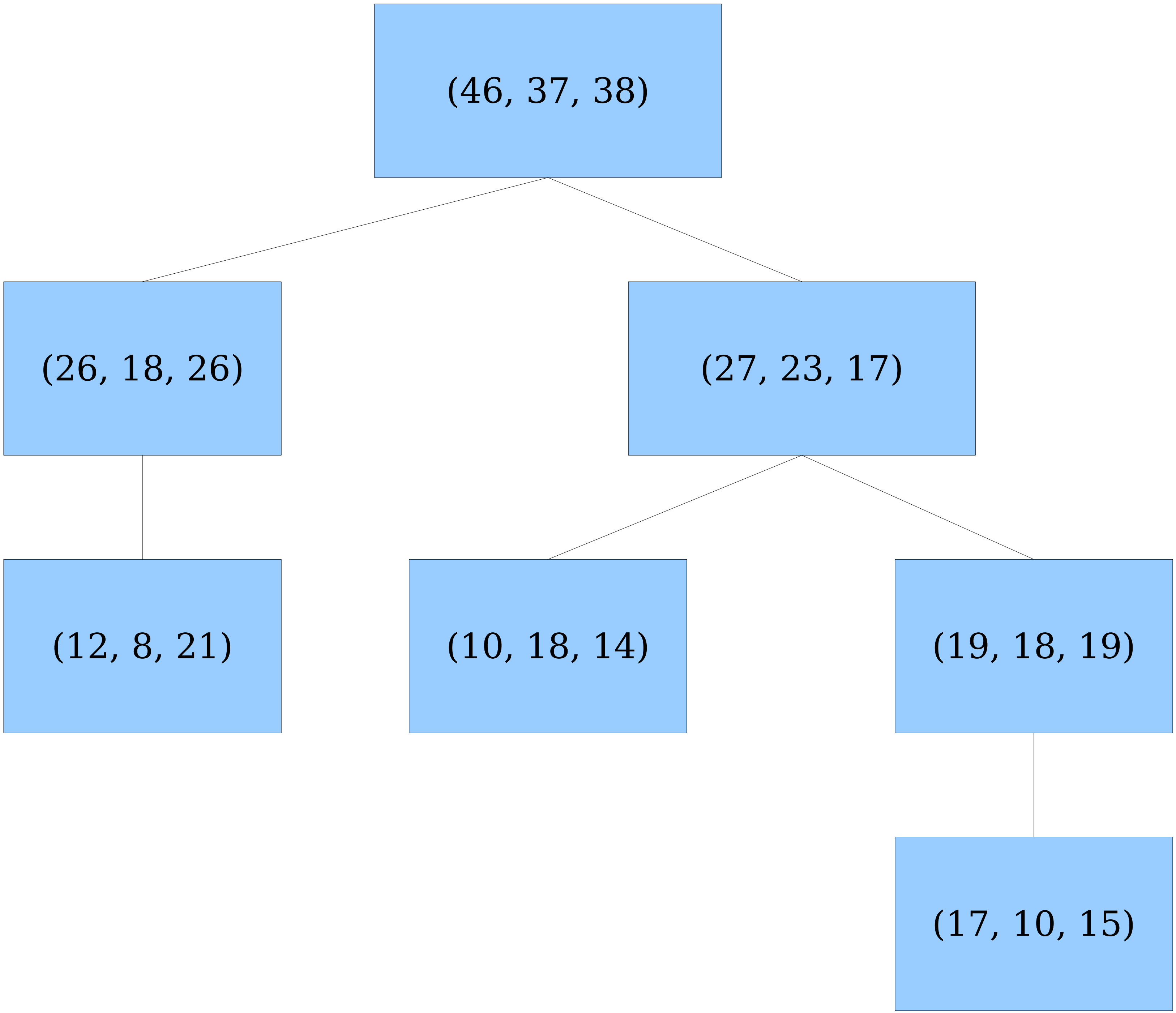
To prove that some binary relation R is a strict order, you can just prove that R is asymmetric and transitive. In the next problem set, you'll see an even simpler technique!

Drawing Strict Orders

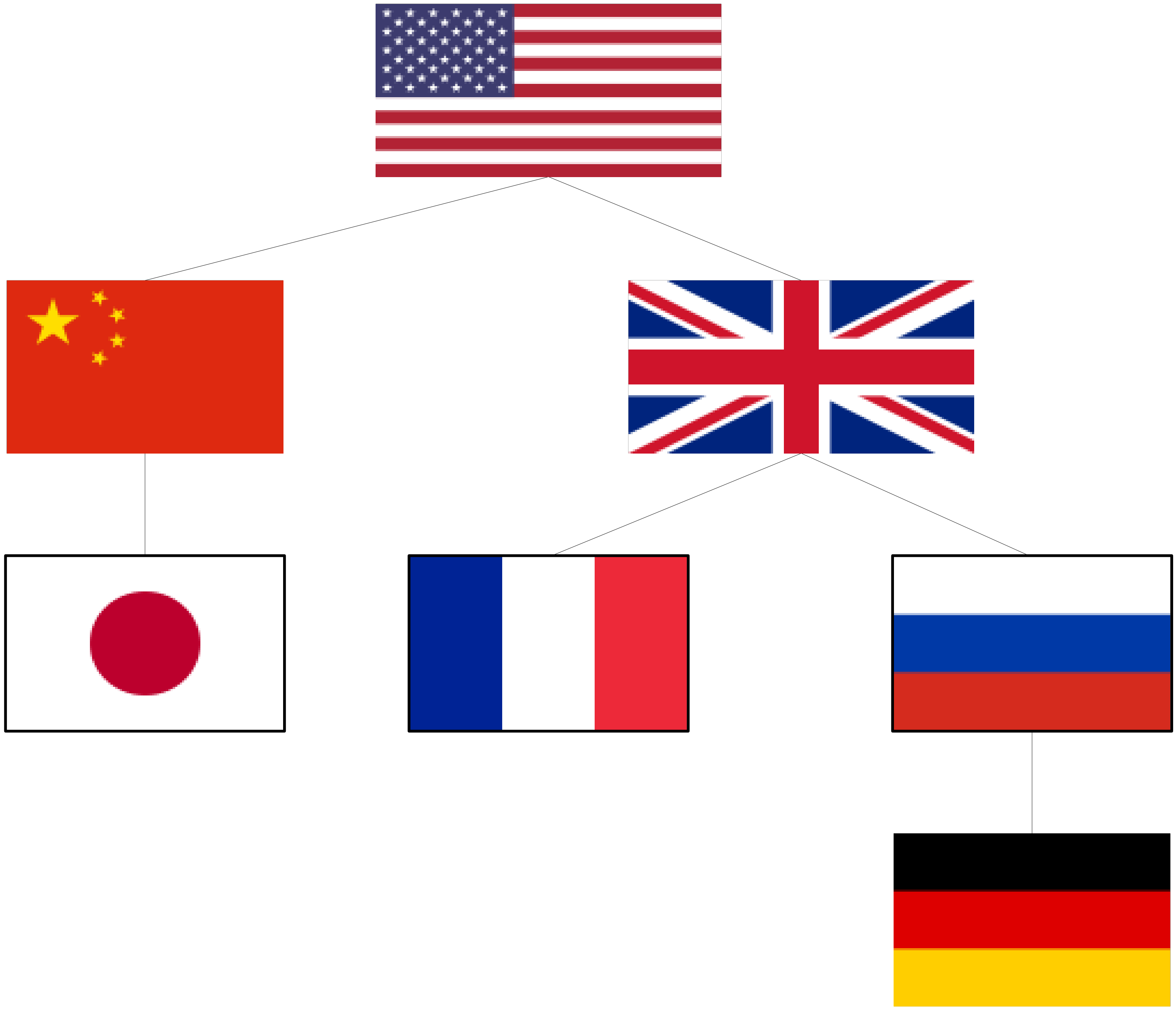
Gold	Silver	Bronze
46	37	38
27	23	17
26	18	26
19	18	19
17	10	15
12	8	21
10	18	14
9	3	9
8	12	8
8	11	10
8	7	4
8	3	4
7	6	6
7	4	6
6	6	1
6	3	2



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



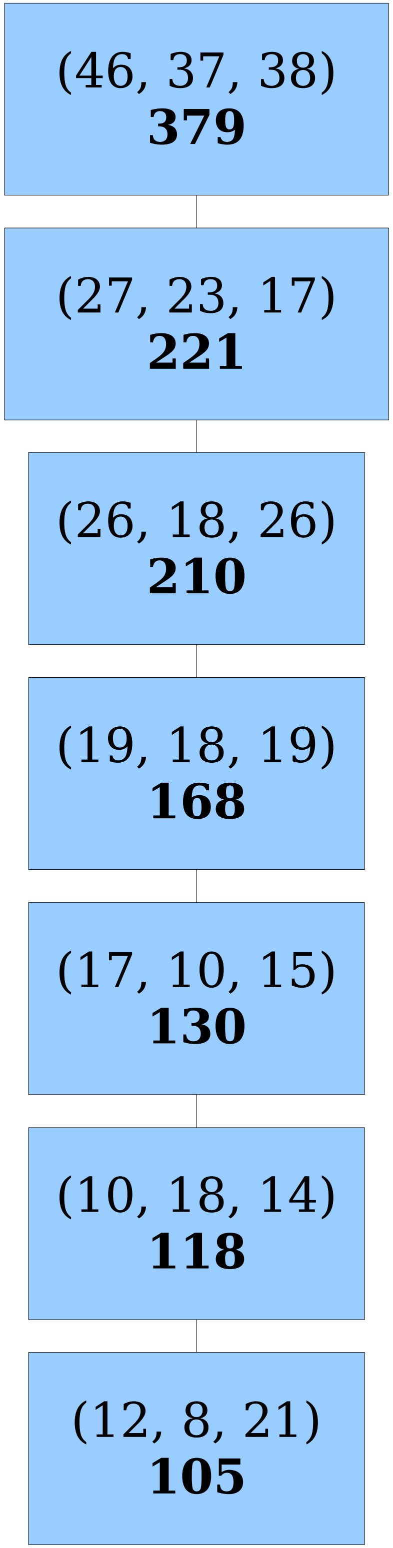
$$(g_1, s_1, b_1)\ R\ (g_2, s_2, b_2) \quad \text{if} \quad g_1 < g_2 \ \wedge\ s_1 < s_2 \ \wedge\ b_1 < b_2$$



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

Hasse Diagrams

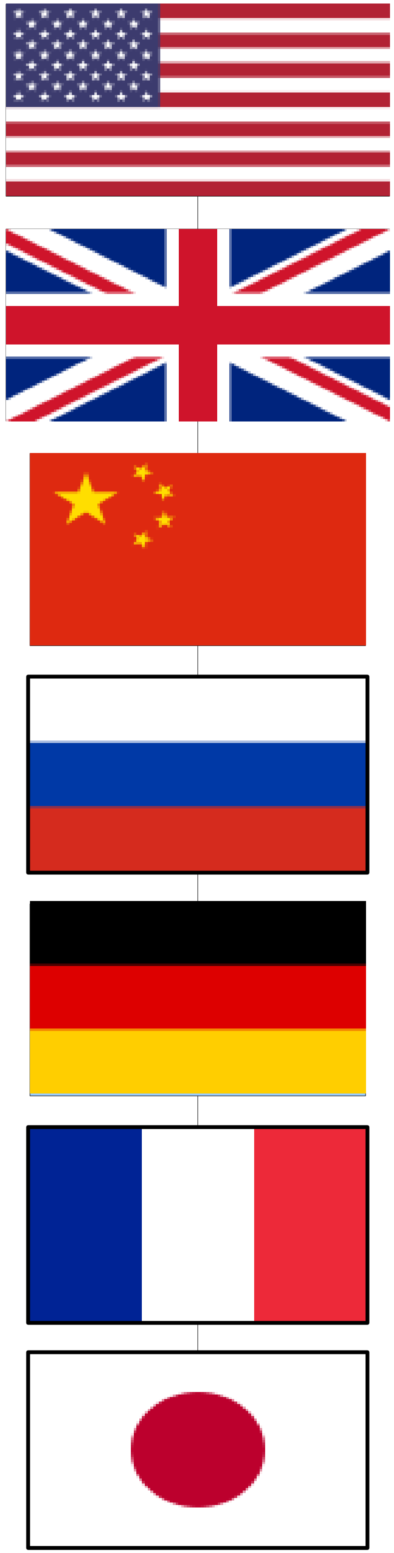
- A ***Hasse diagram*** is a graphical representation of a strict order.
- Elements are drawn from bottom-to-top.
- Higher elements are bigger than lower elements: by ***asymmetry***, the edges can only go in one direction.
- No redundant edges: by ***transitivity***, we can infer the missing edges.



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

if

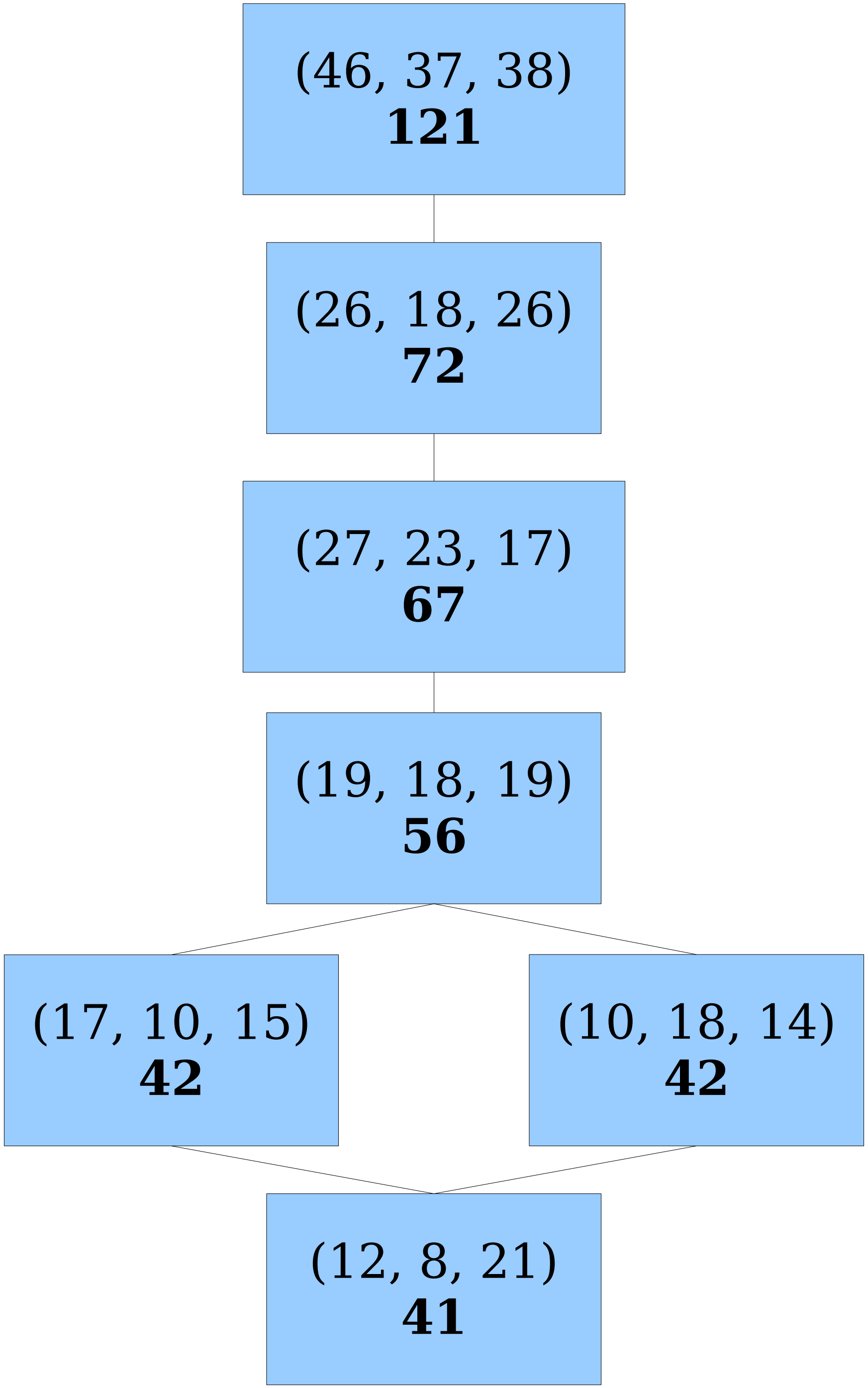
$5g_1 + 3s_1 + b_1 < 5g_2 + 3s_2 + b_2$



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

if

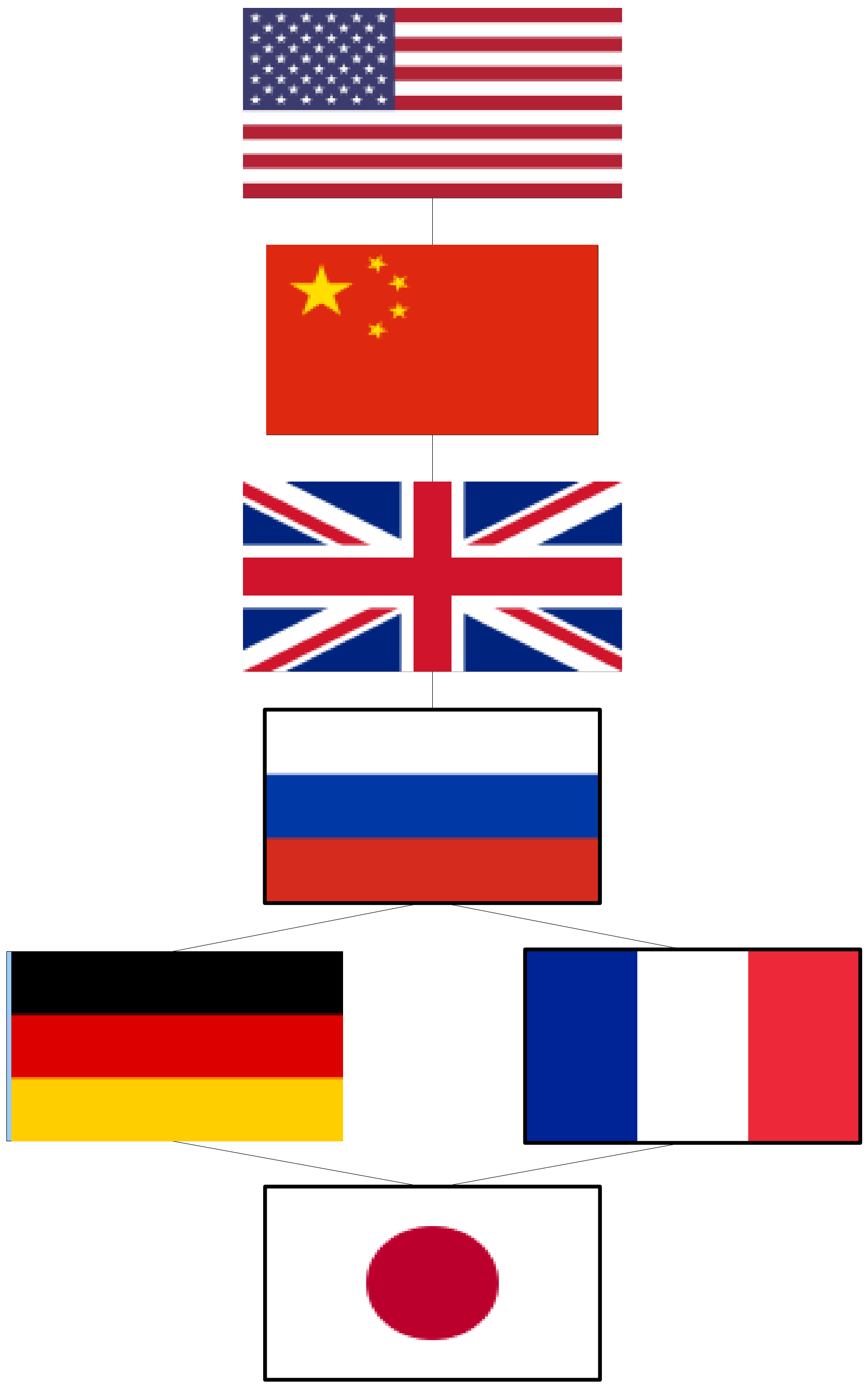
$$5g_1 + 3s_1 + b_1 < 5g_2 + 3s_2 + b_2$$



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

if

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

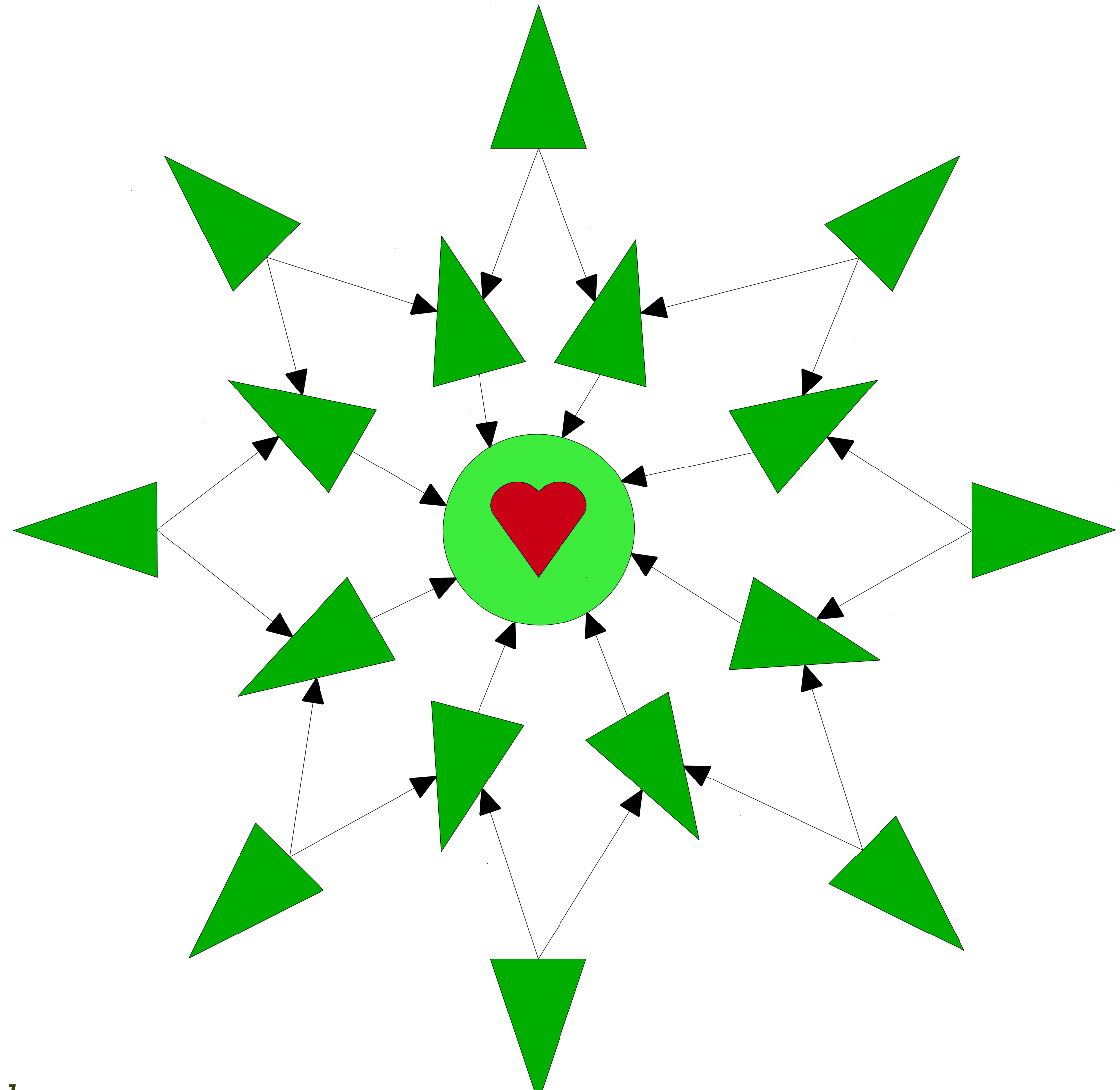


$$(g_1, s_1, b_1)\ U\ (g_2, s_2, b_2)$$

if

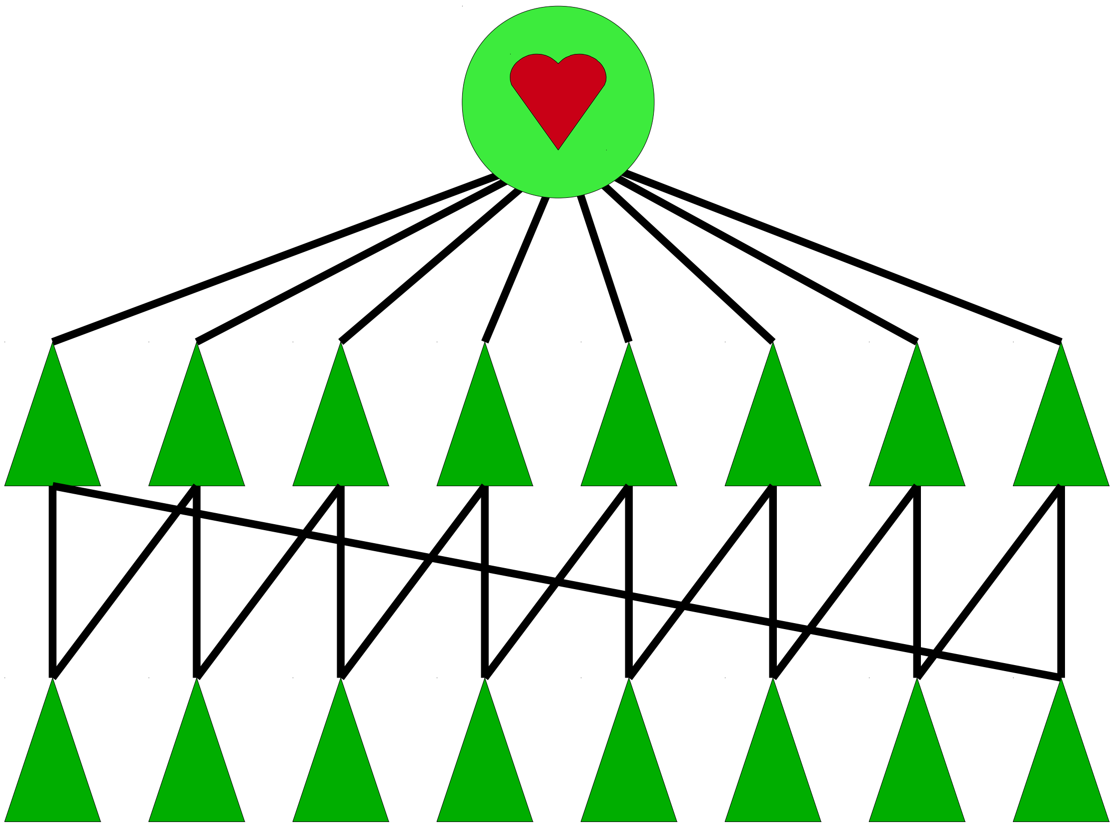
$$g_1 + s_1 + b_1 < g_2 + s_2 + b_2$$

Hasse Artichokes



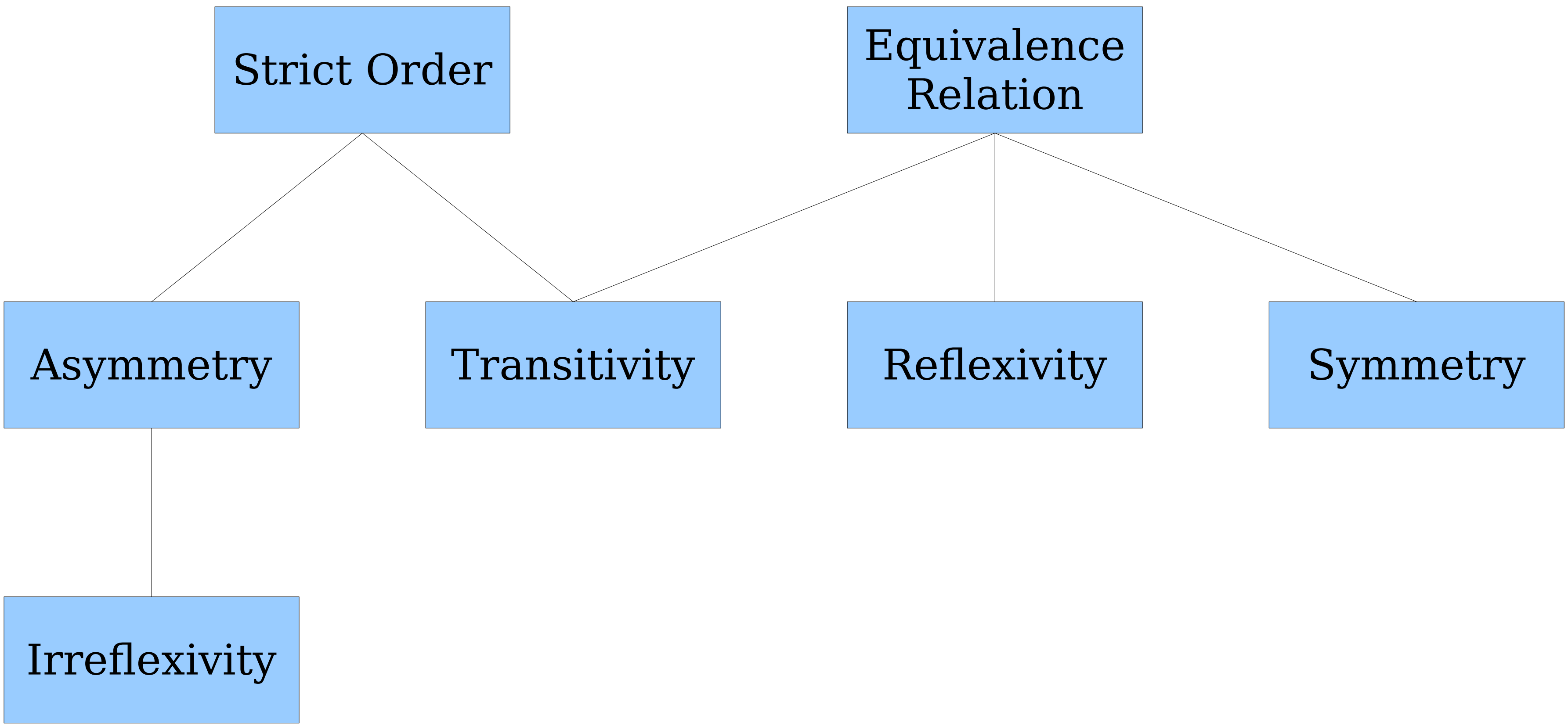
xRy   if   x must be eaten before y

Hasse Artichokes



xRy   if   x must be eaten before y

The Meta Strict Order



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

The Binary Relation Editor

Next Time

- ***Functions***
  - How do we model transformations between objects?
- ***Injections and Surjections***
  - Special and highly useful classes of functions.
- ***Function Composition***
  - How do we combine functions together?