Binary Relations

Outline for Today

- **Binary Relations**
  - Studying connections between objects from a different perspective.
- **Equivalence Relations**
  - Relations that break objects into groups.
- **Partial Orders**
  - Relations for ranking objects.
- **Quick Midterm Review**
  - Clarifying concepts from PS3

Studying Relationships

- We've explored graphs as a mathematical model of connections between objects.
- However, graphs are not the only formalism we can use to do this.
- Today, we'll study *binary relations*, a different way of studying and modeling relations between objects.

Relationships

- In CS103, you've seen examples of relationships
  - between sets:
    - $A \subseteq B$
  - between numbers:
    - $x < y \quad x \equiv_k y \quad x \leq y$
  - between nodes in a graph:
    - $u \leftrightarrow v$
- **Goal:** Focus on these types of relationships and study their properties.

Some Notation

- Consider these examples:

$$a = b \quad a < b \quad A \subseteq B \quad a \equiv_k b \quad u \leftrightarrow v$$

- In each case, we're indicating a specific relationship between two objects by writing those objects with an appropriate symbol between them.
- Today, we will be talking about general classes of relations.
- **Notation:** If R is some relationship and a is related to b by relation R, we write

$$\boldsymbol{aRb}$$

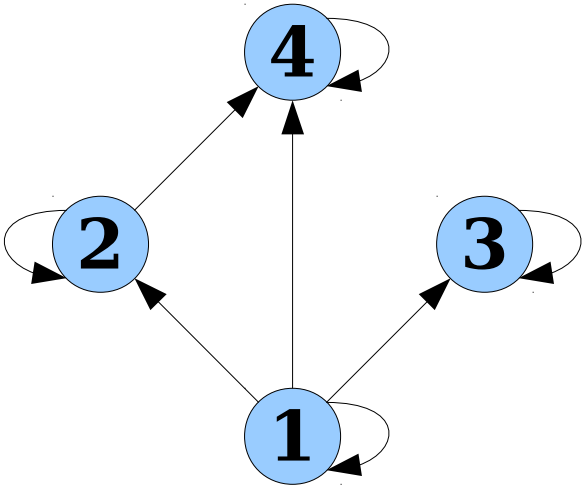
- *Order matters.* For example, if $a < b$, then we know for a fact that $b \not< a$.

Binary Relations

- A ***binary relation over a set A*** is some relation R where, for every x, $y \in A$, the statement xRy is either true or false.
- Examples:
  - $<$ can be a binary relation over $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$, etc.
  - $\leftrightarrow$ can be a binary relation over V for any undirected graph $G = (V, E)$.
  - $\equiv_k$ is a binary relation over $\mathbb{Z}$ for any integer k.

Binary Relations and Graphs

- We can visualize a binary relation R over a set A as a graph:
  - The nodes are the elements of A.
  - There is an edge from x to y iff xRy.
- Example: the relation $a \mid b$ (meaning "a divides b") over the set {1, 2, 3, 4} looks like this:



Categorizing Relations

- Collectively, there are few properties shared by all relations.
- We often categorize relations into different types to study relations with particular properties.
- General outline for today:
  - Find certain properties that hold of the relations we've seen so far.
  - Categorize relations based on those properties.
  - See what those properties entail.

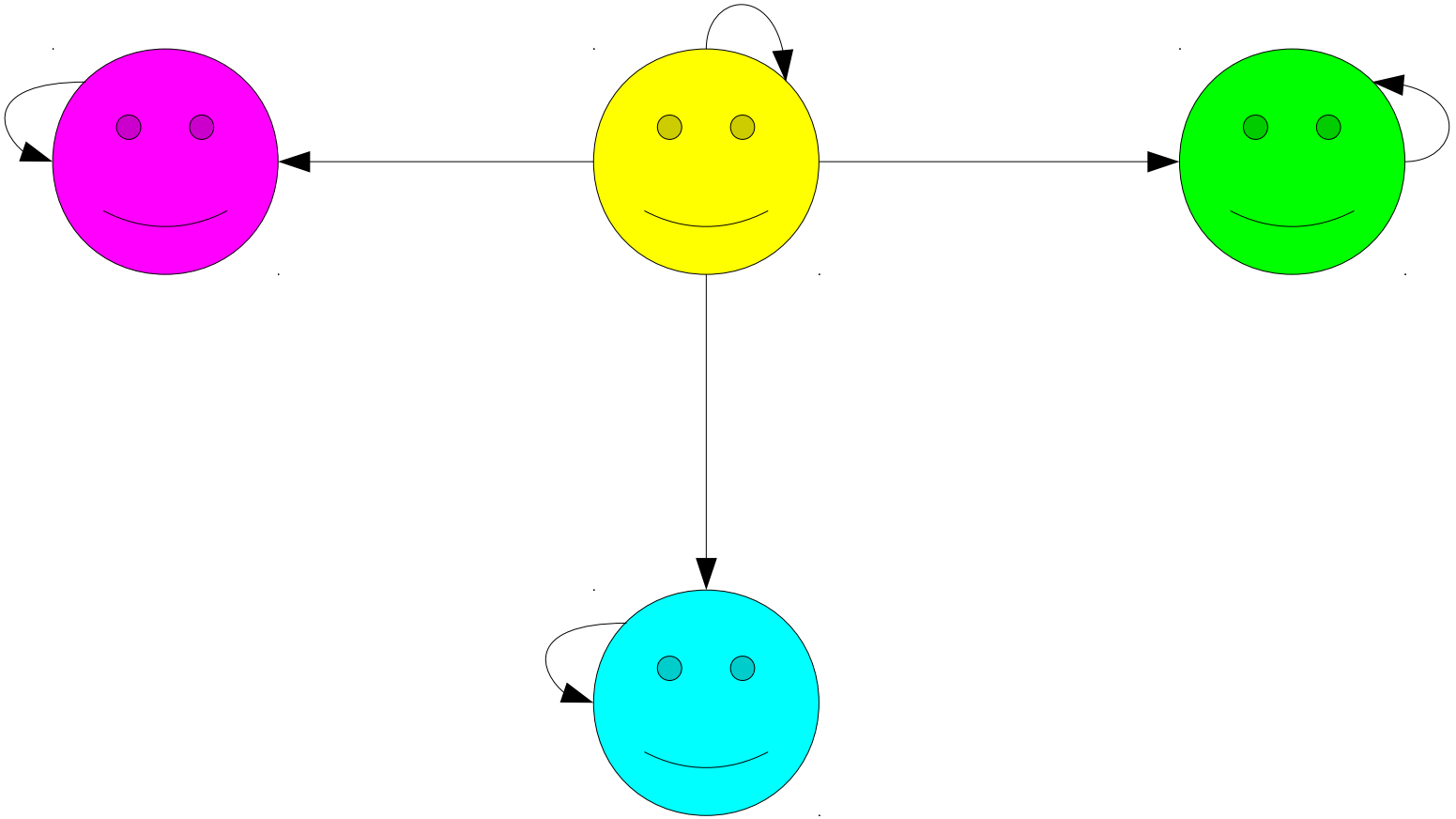
Reflexivity

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

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

(“Every element is related to itself.”)

An Intuition for Reflexivity



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

(“Every element is related to itself.”)

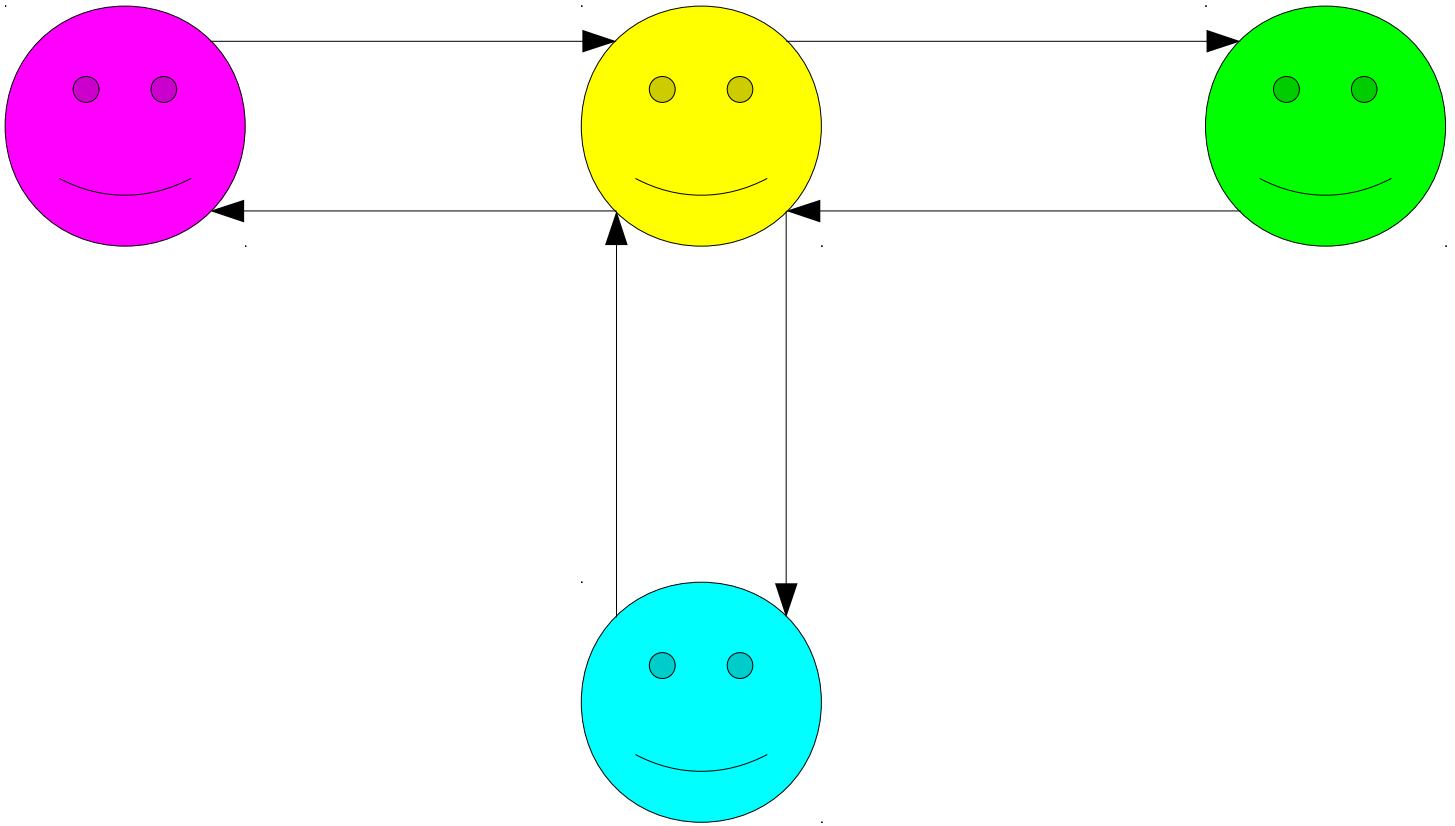
Symmetry

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

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

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

An Intuition for Symmetry



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

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

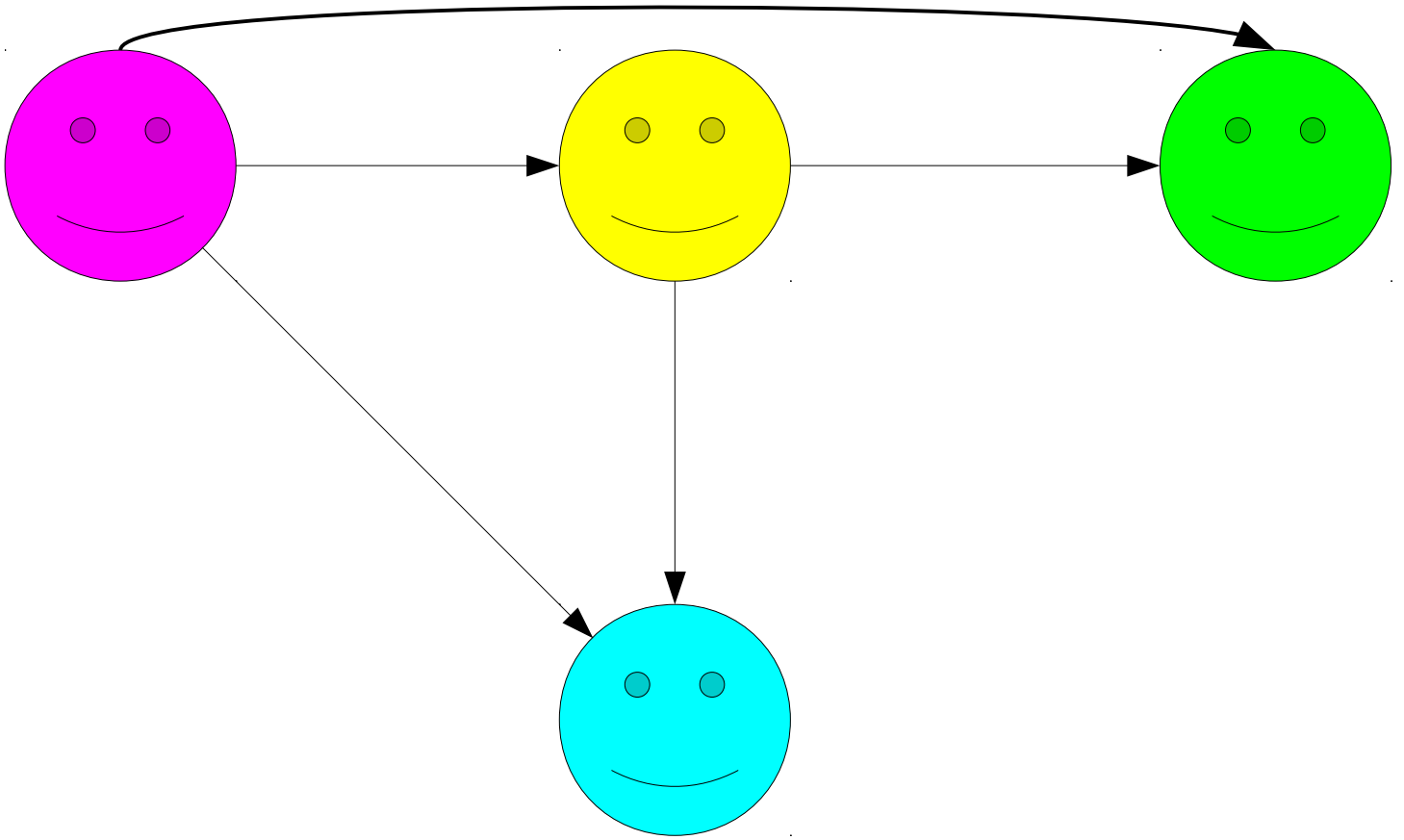
Transitivity

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

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

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

An Intuition for Transitivity

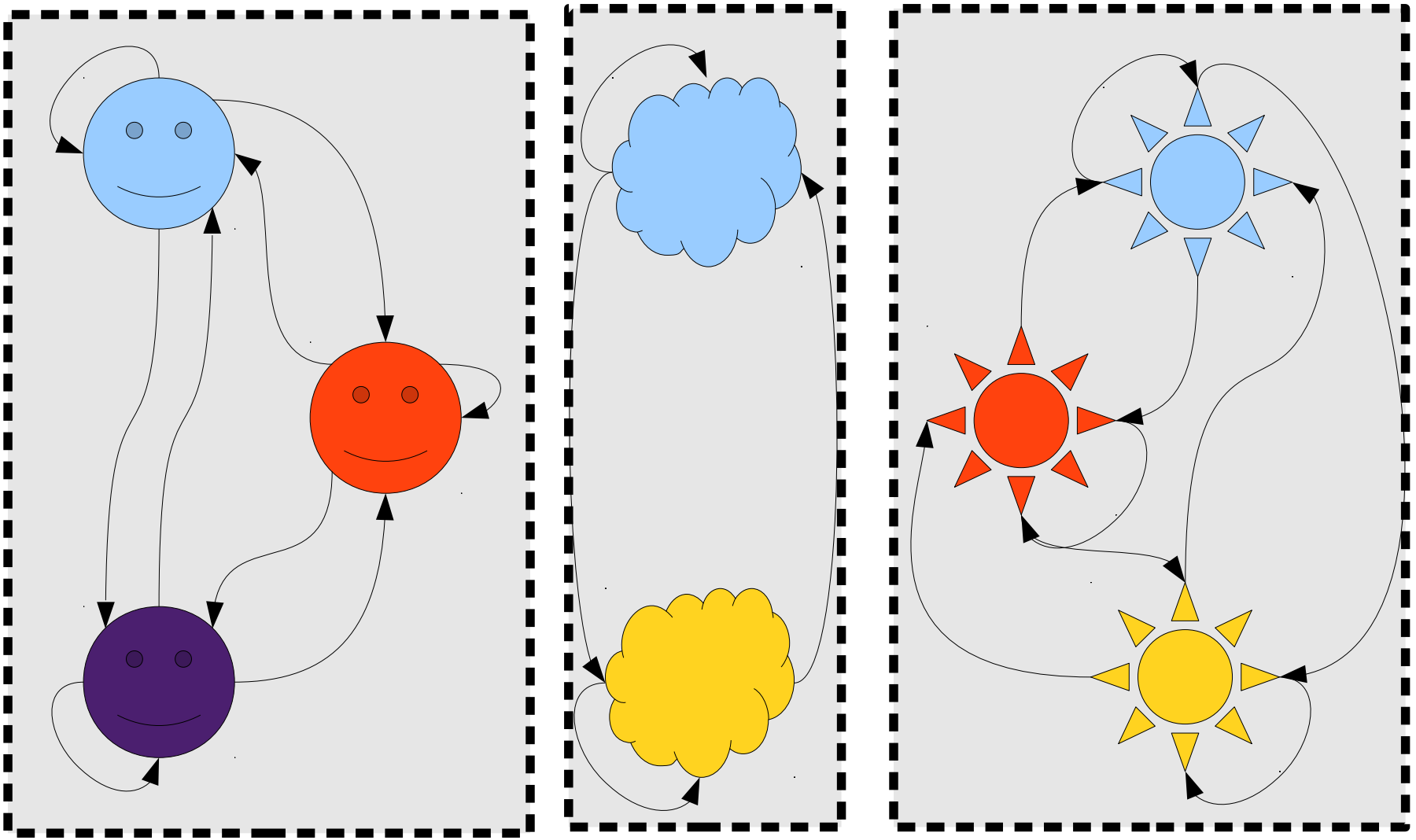


$$\forall a \in A.\ \forall b \in A.\ \forall c \in A.\ (aRb \wedge bRc \rightarrow aRc)$$

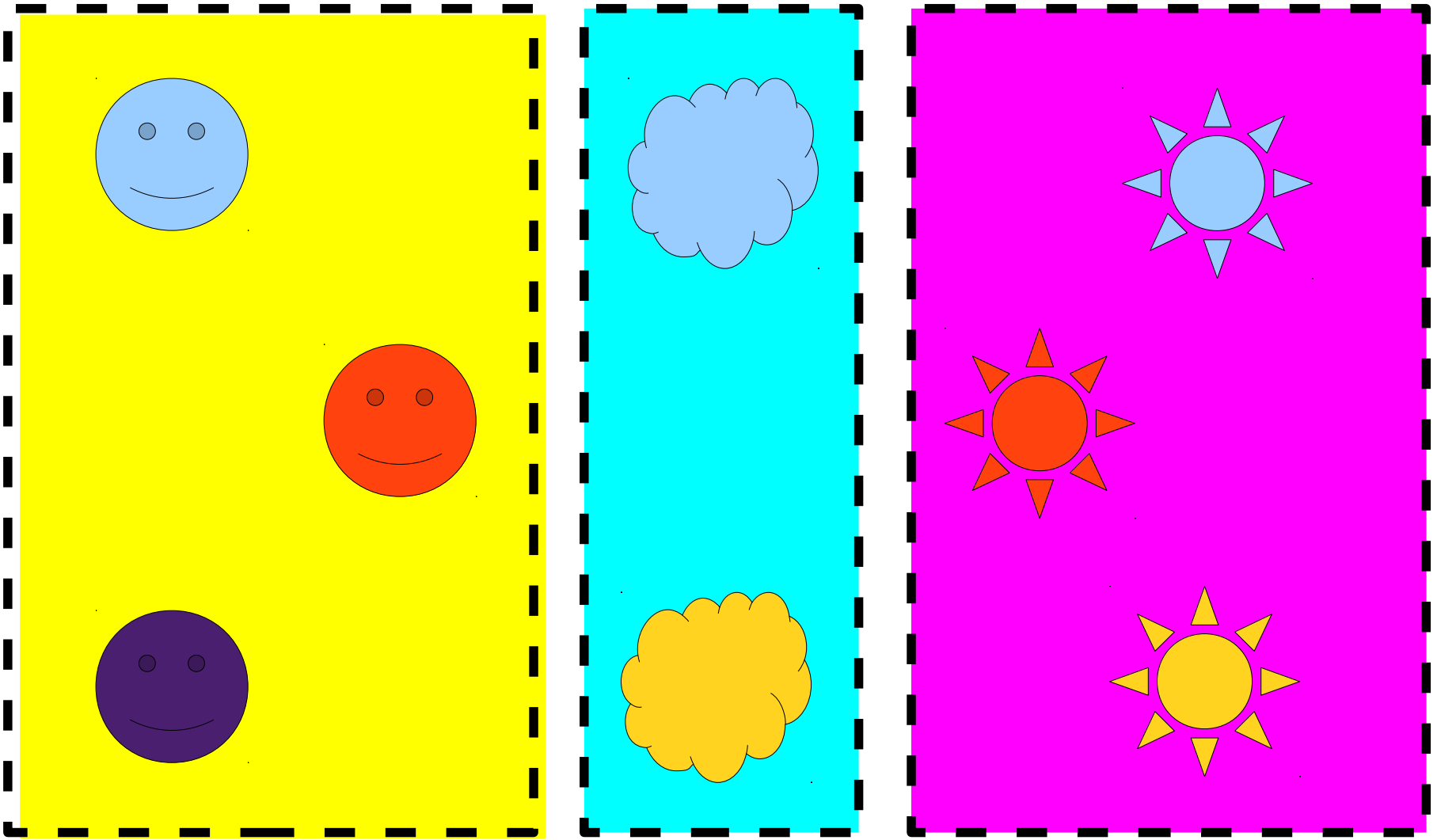
("Whenever a is related to b and b is related to c, we know a is related to c.)

Equivalence Relations

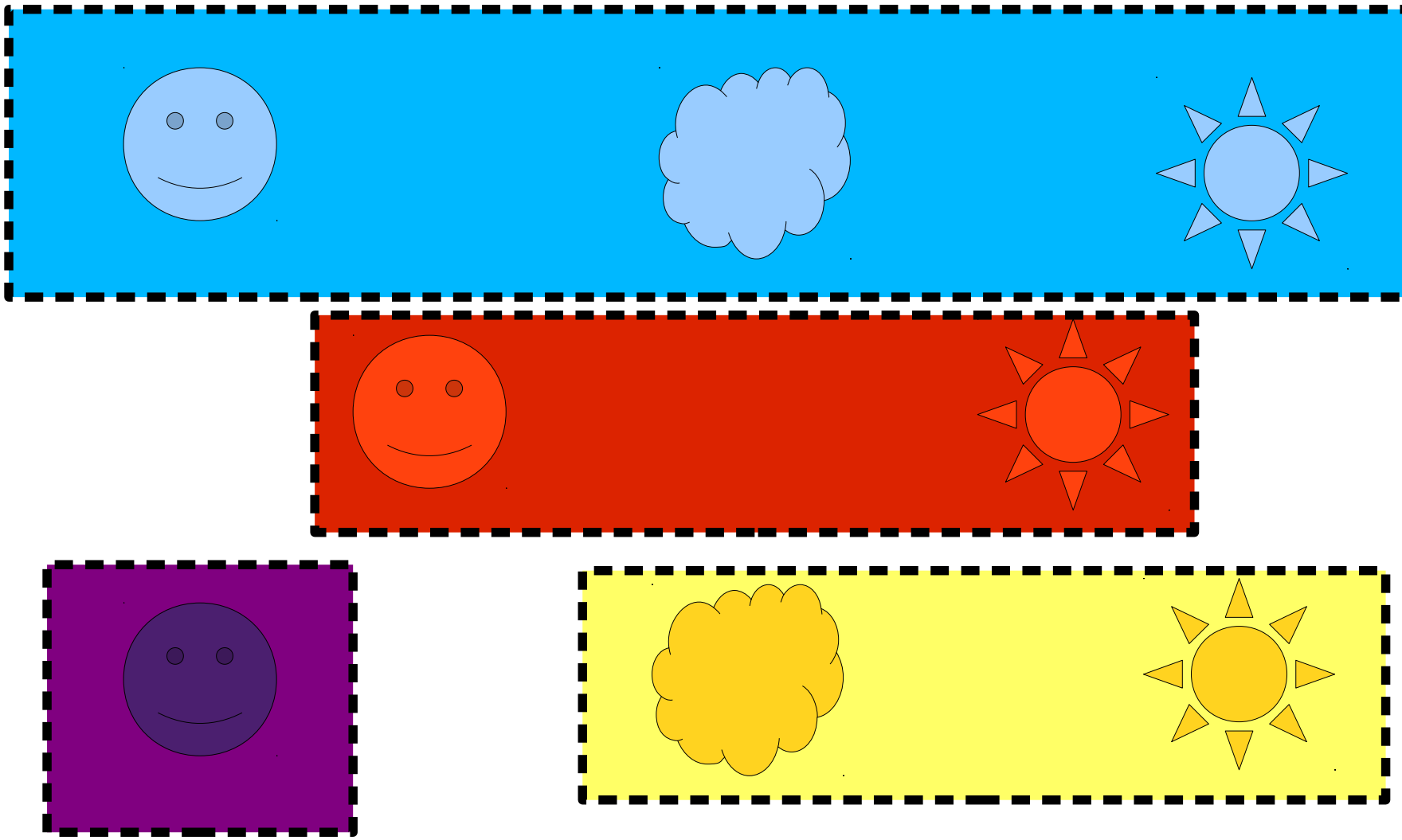
- Some relations are reflexive, symmetric, and transitive:
  - $x = y$
  - $u \leftrightarrow v$
  - $x \equiv_k y$
- Definition: An ***equivalence relation*** is a relation that is reflexive, symmetric and transitive.



Let R be “has the same shape as”



Let R be “has the same shape as”



Let T be “is the same **color** as”

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.
- ***Theorem:*** If R is an equivalence relation over A, then every $a \in A$ belongs to exactly one equivalence class.

Closing the Loop

- In any graph $G = (V, E)$, we saw that the connected component containing a node $v \in V$ is given by

$$\{ x \in V \mid v \leftrightarrow x \}$$

- What is the equivalence class for some node $v \in V$ under the relation $\leftrightarrow$?

$$[v]_{\leftrightarrow} = \{ x \in V \mid v \leftrightarrow x \}$$

- *Connected components are just equivalence classes of* $\leftrightarrow$!

Why This Matters

- Developing the right definition for a connected component was challenging.
- Proving every node belonged to exactly one equivalence class was challenging.
- Now that we know about equivalence relations, we get both of these for free!
- **If you arrive at the same concept in two or more ways, it is probably significant!**

Time-Out for Announcements!

Midterm Logistics

- Midterm exam is this Thursday, 6PM – 9PM.
- Room locations divvied up by last name:
  - Abr – Hsu: Go to **420-040**
  - Hua – Med: Go to **420-041**
  - Mei – Raj: Go to **Art 2**
  - Ram – Saw: Go to **Art 4**
  - Sch – Zie: Go to **Bishop Auditorium**
- Alternate exam locations will be handled over email; you should hear back with location information later today.

Midterm Logistics

- Exam is closed-book, closed-computer.
- You may have one double-sided sheet of notes on 8.5” × 11” paper.
- Covers material up through and including first-order logic.

Extra Review

- There are now three sets of extra review problems, one released last Wednesday, one released last Friday, and one released today.
- Solutions to the first two now available for pickup in hard copy.
- Solutions to the third set of practice problems goes out on Wednesday.

Practice Midterm

- We will be holding a practice midterm exam from 7PM – 10PM ***tonight*** in Annenberg Auditorium.
- Purely optional, but would be a great way to practice for the exam.
- Course staff will be available outside the practice exam to answer your questions afterwards.
- We'll release the practice exam questions tonight / Tuesday morning.

Problem Set Four

- PS4 checkpoint will be returned later than usual (Friday instead of Wednesday) so that we can get all PS3's graded before the midterm.
- Sorry about that!

Reviewing Q6

$$(\forall x.\ Happy(x)) \rightarrow (\forall y.\ Happy(y))$$

$$\forall x.\ (Happy(x) \rightarrow (\forall y.\ Happy(y)))$$

$$\forall x.\ \forall y.\ (Happy(x) \rightarrow Happy(y))$$

Approaching Math Problems

- ***Try lots of examples***. Lots of math problems look much harder than they are. Working through examples often helps you spot patterns.
- ***Expand out definitions***. Sometimes, writing out the definitions involved in a problem will help guide how you go about approaching the problem.
- ***Think about related problems***. The more math you do, the more problems you'll have seen, and the more solution routes you'll know.

Back to CS103!

Partial Orders

Partial Orders

- Many relations are equivalence relations:

$$x = y \qquad x \equiv_k y \qquad u \leftrightarrow v$$

- What about these sorts of relations?

$$x \leq y \qquad x \subseteq y$$

- These relations are called ***partial orders***, and we'll explore their properties next.

Properties of Partial Orders

$$x \leq y$$

$$x \leq y \quad \textbf{and} \quad y \leq z$$

$$x \leq z$$

Transitivity

Properties of Partial Orders

$$x \leq y$$

$$x \leq x$$

Reflexivity

Properties of Partial Orders

$x \leq y$

$19 \leq 21$

~~$21 \leq 19$~~?

Properties of Partial Orders

$x \leq y$

$42 \leq 137$

~~$137 \leq 42$~~?

Properties of Partial Orders

$x \leq y$

$137 \leq 137$

$137 \leq 137$

Antisymmetry

- A binary relation R over a set A is called ***antisymmetric*** if the following is true:

$$\forall a \in A.\, \forall b \in A.\, (a \neq b \wedge aRb \rightarrow \neg(bRa))$$

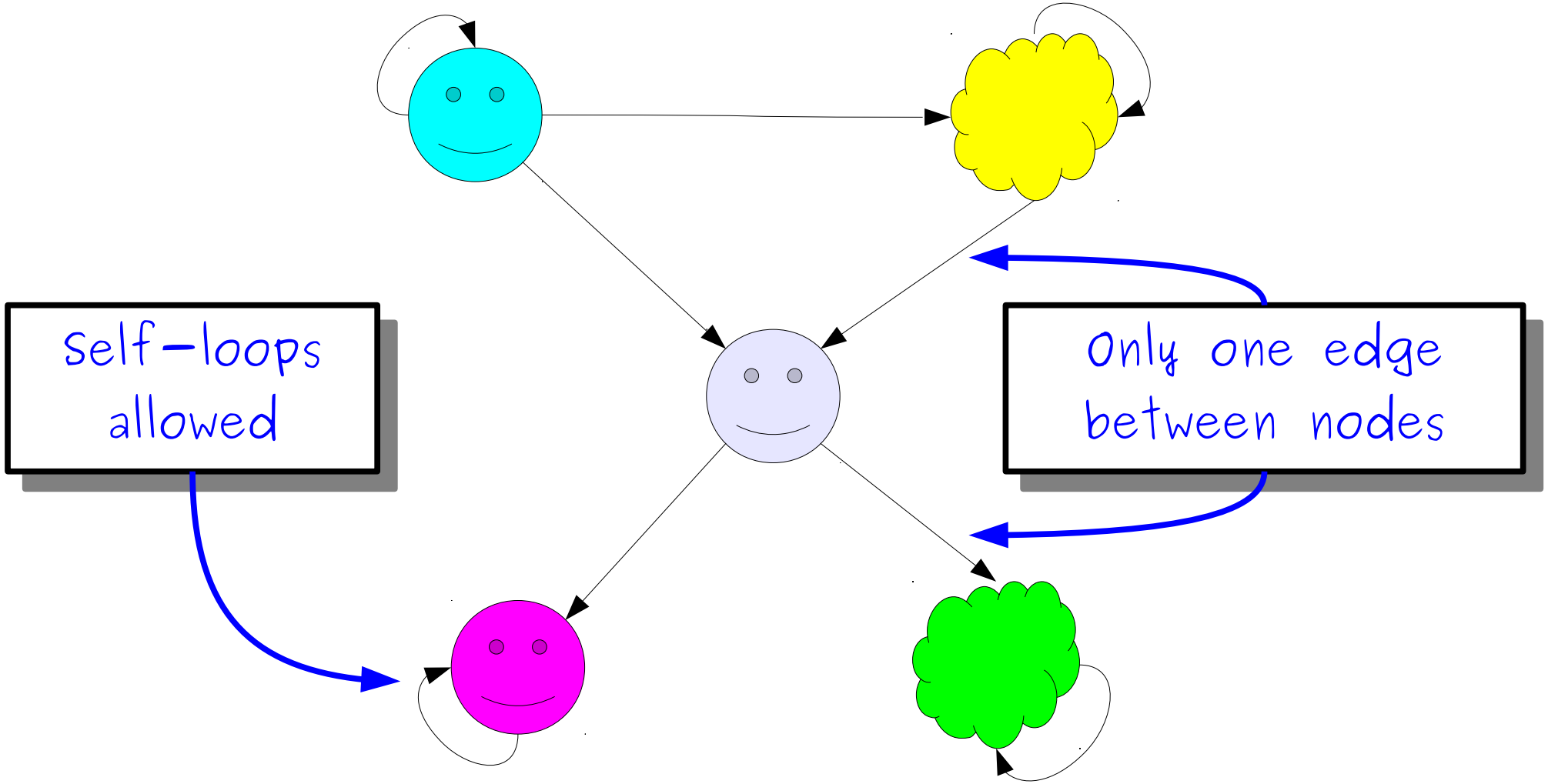
("If a is related to b and a ≠ b, then b is not related back to a.")

- Equivalently:

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

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

An Intuition for Antisymmetry



$$\forall a \in A.\, \forall b \in A.\, (a \neq b \wedge aRb \rightarrow \neg(bRa))$$

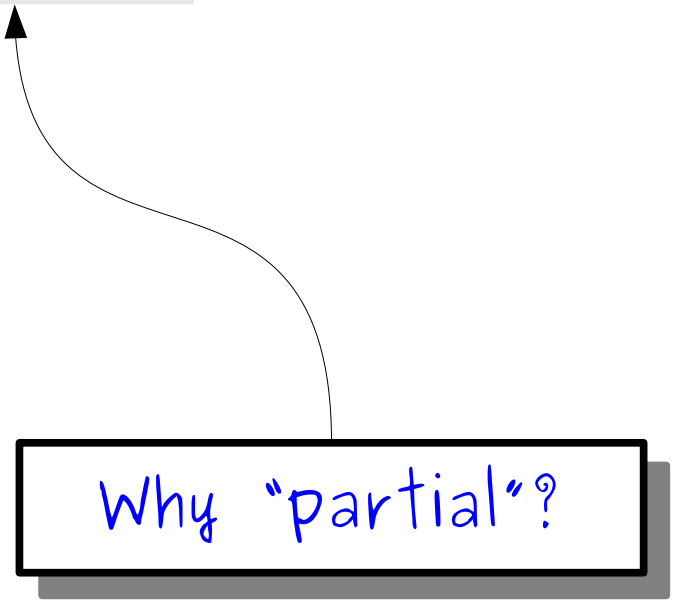
("If a is related to b and a ≠ b, then b is not related back to a.")

Antisymmetry

- Antisymmetry is probably the least intuitive of the four properties we've just seen.
- A few notes:
  - It's helpful to think of $\leq$ or $\subseteq$ as canonical examples of antisymmetric relations.
  - Antisymmetry is ***not*** the opposite of symmetry. There are relations that are both symmetric and antisymmetric (as you'll see in Problem Set Four).

Partial Orders

- A binary relation R is a ***partial order*** over a set A if it is
  - ***reflexive***,
  - ***antisymmetric***, and
  - ***transitive***.



2012 Summer Olympics

Gold	Silver	Bronze	Total
46	29	29	104
38	27	23	88
29	17	19	65
24	26	32	82
13	8	7	28
11	19	14	44
11	11	12	34

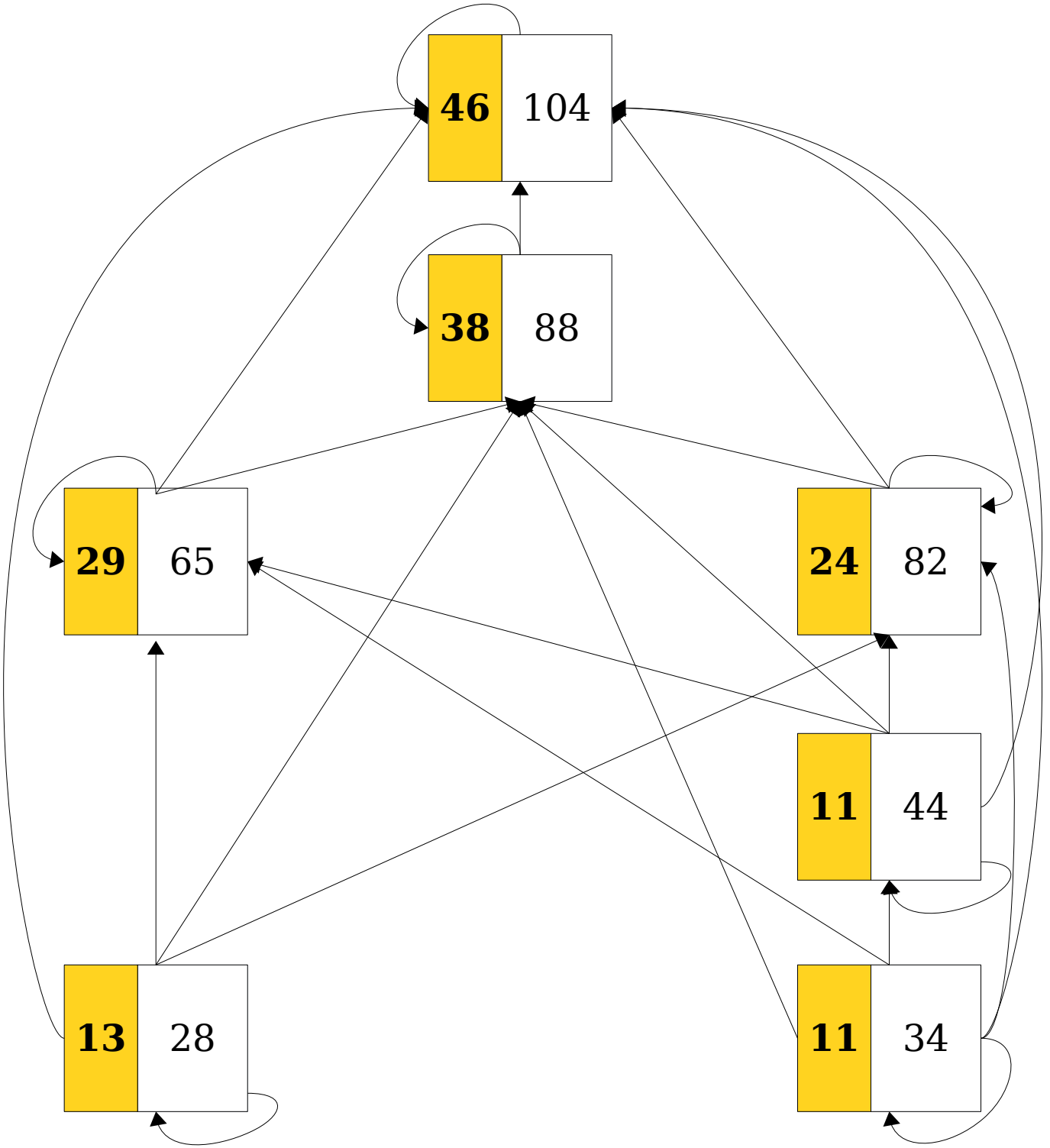
Inspired by http://tartarus.org/simon/2008-olympics-hasse/
Data from http://www.london2012.com/medals/medal-count/

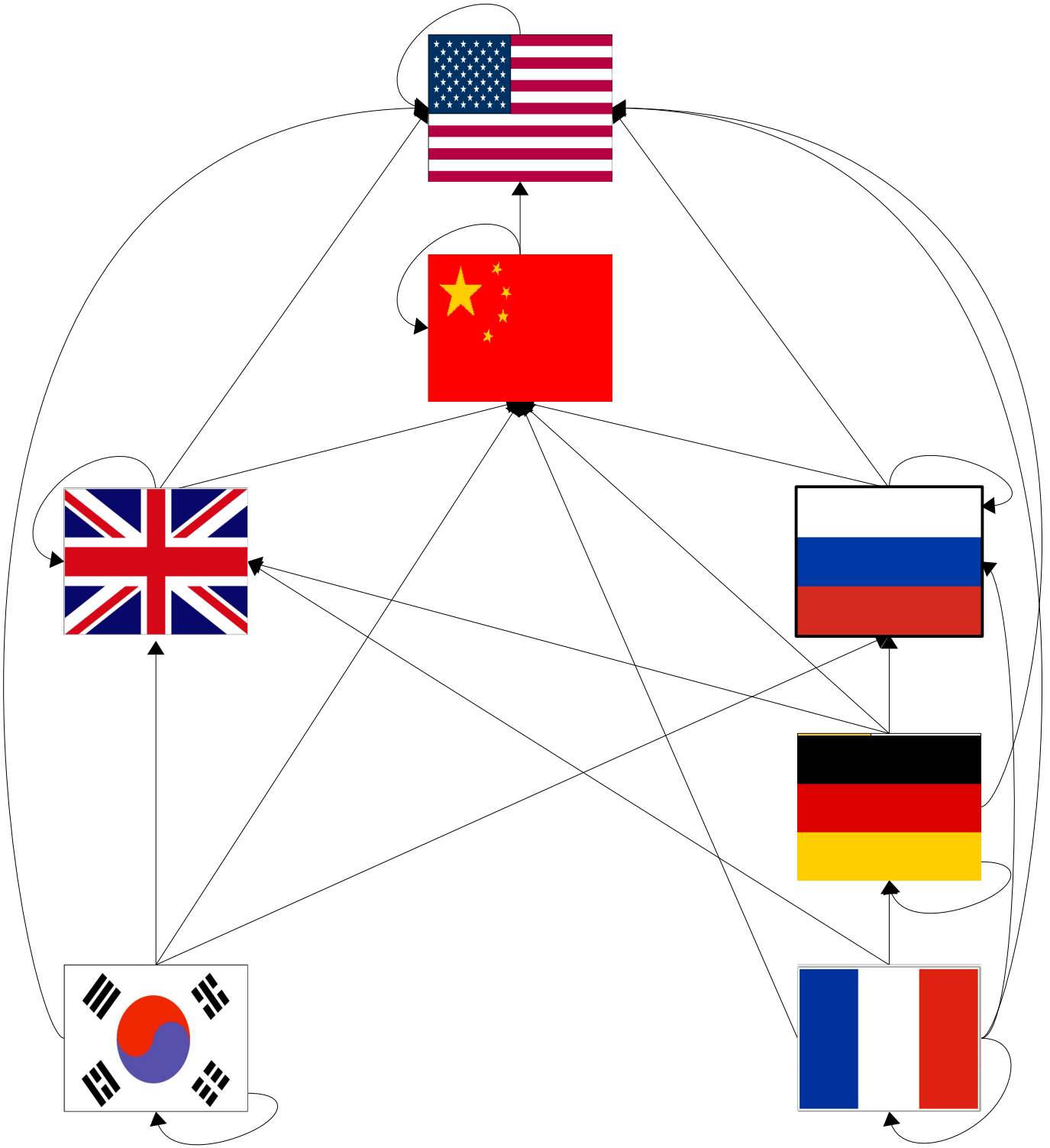
Define the relationship

$$(\mathbf{gold}_0, \mathbf{total}_0) \boldsymbol{R} (\mathbf{gold}_1, \mathbf{total}_1)$$

to be true when

$$\mathbf{gold}_0 \leq \mathbf{gold}_1 \text{ and } \mathbf{total}_0 \leq \mathbf{total}_1$$


46
104
38
88
29
65
24
82
11
44
13
28
11
34



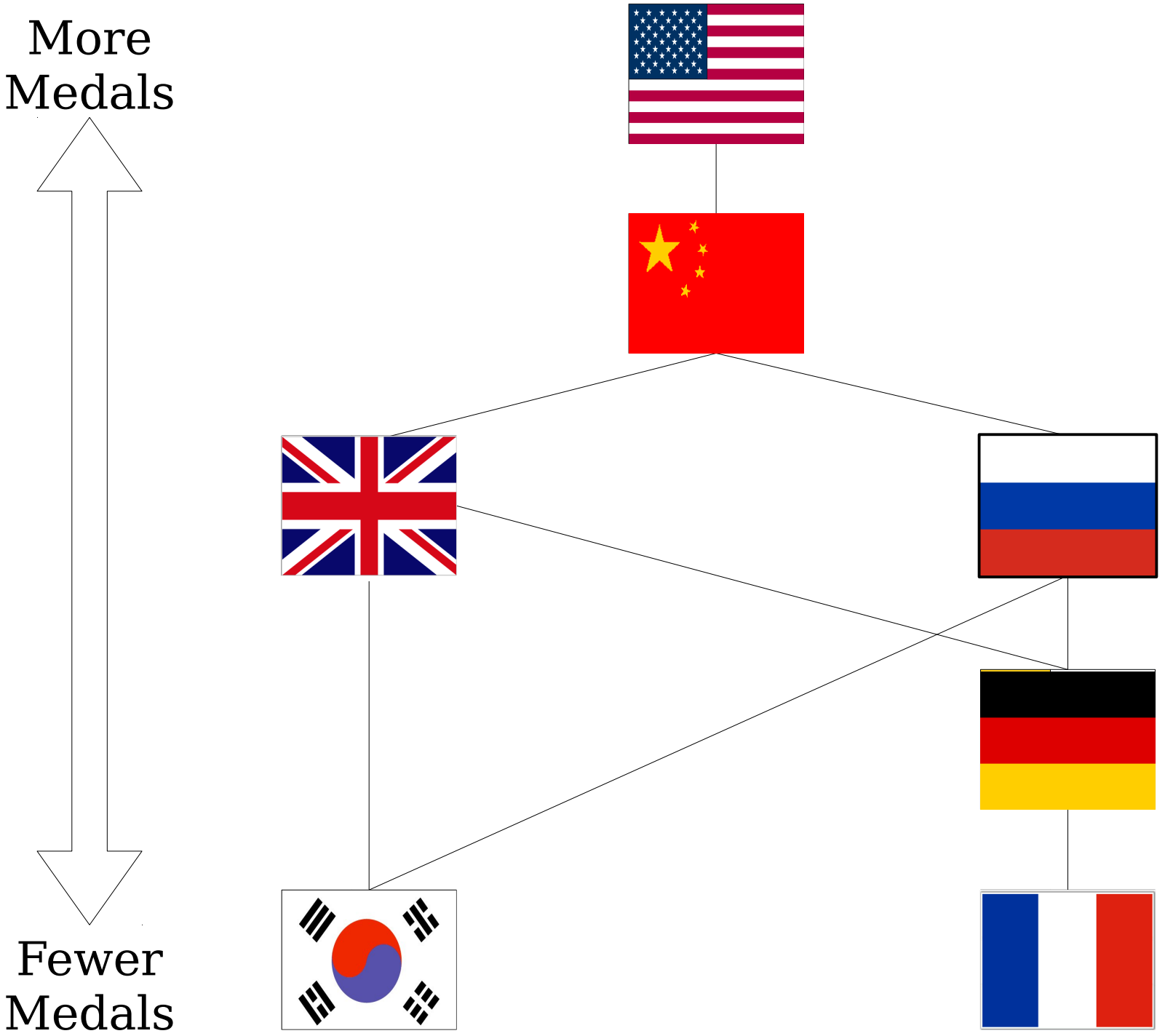
Total Orders

- If R is a partial order relation over a set A, it's possible that some elements of A are incomparable by R.
- In some partial orders, any pair of elements can be compared.
- A binary relation R over a set A is ***total*** if

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

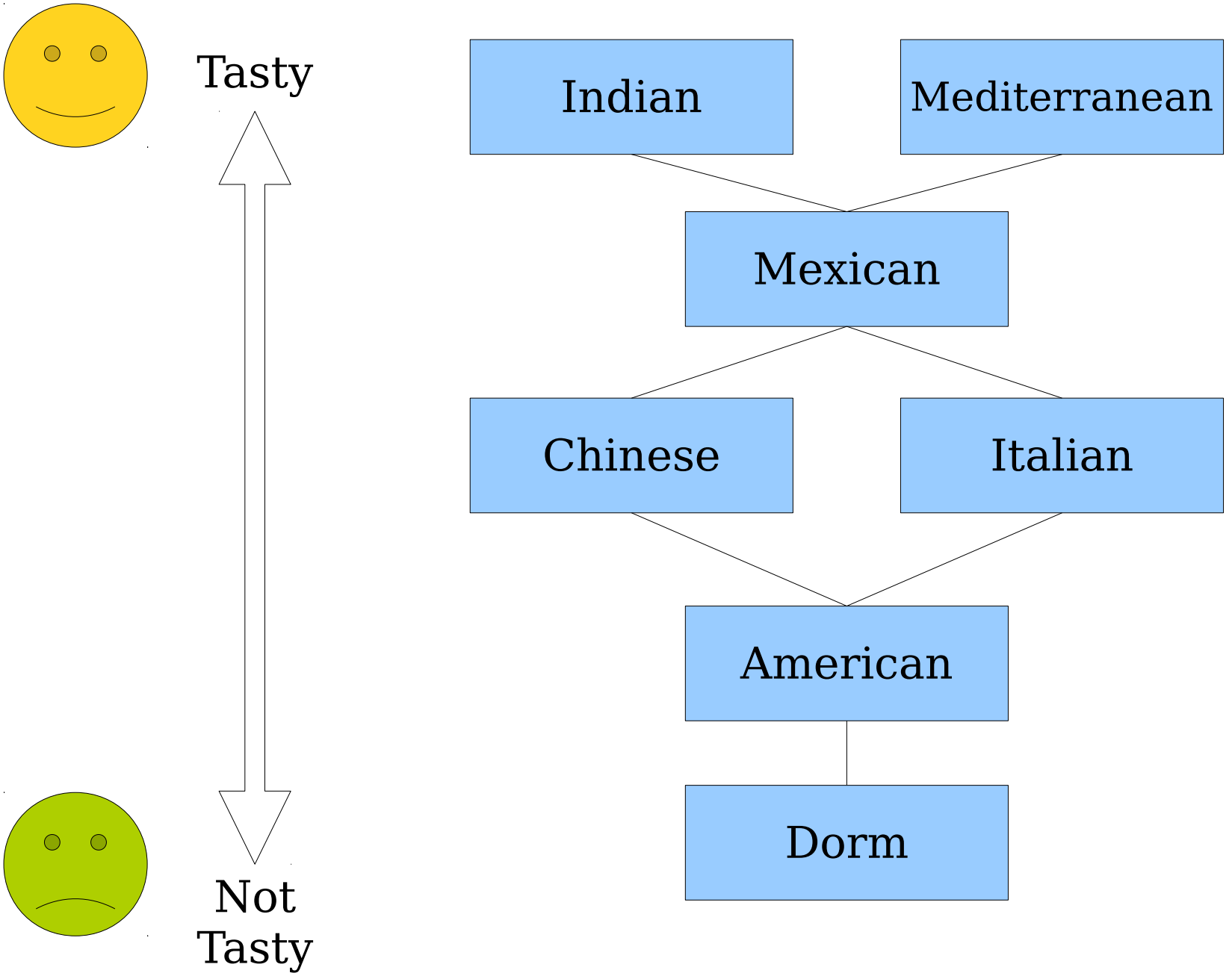
("Any two elements can be compared by R")

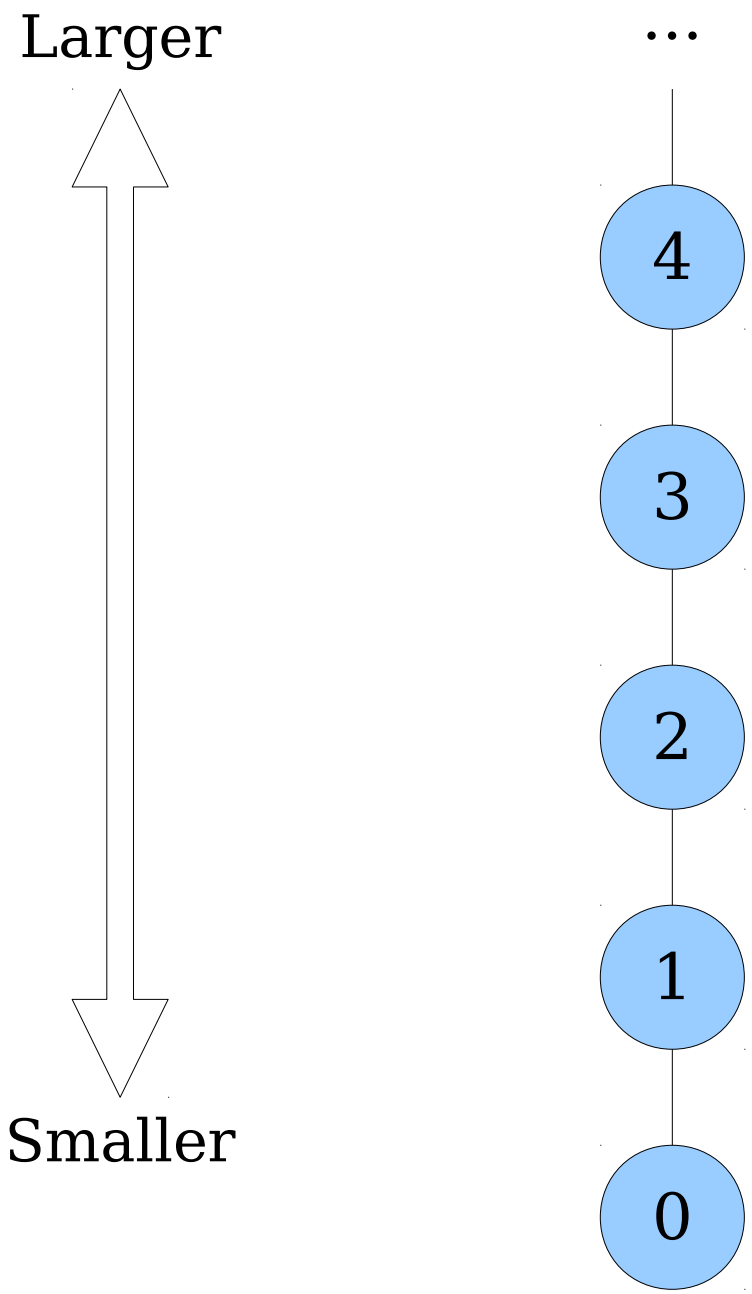
- A binary relation R over a set A is called a ***total order*** if R is a partial order and R is total.
  - The ≤ relation is a great example of a total order.


More
Medals
Fewer
Medals

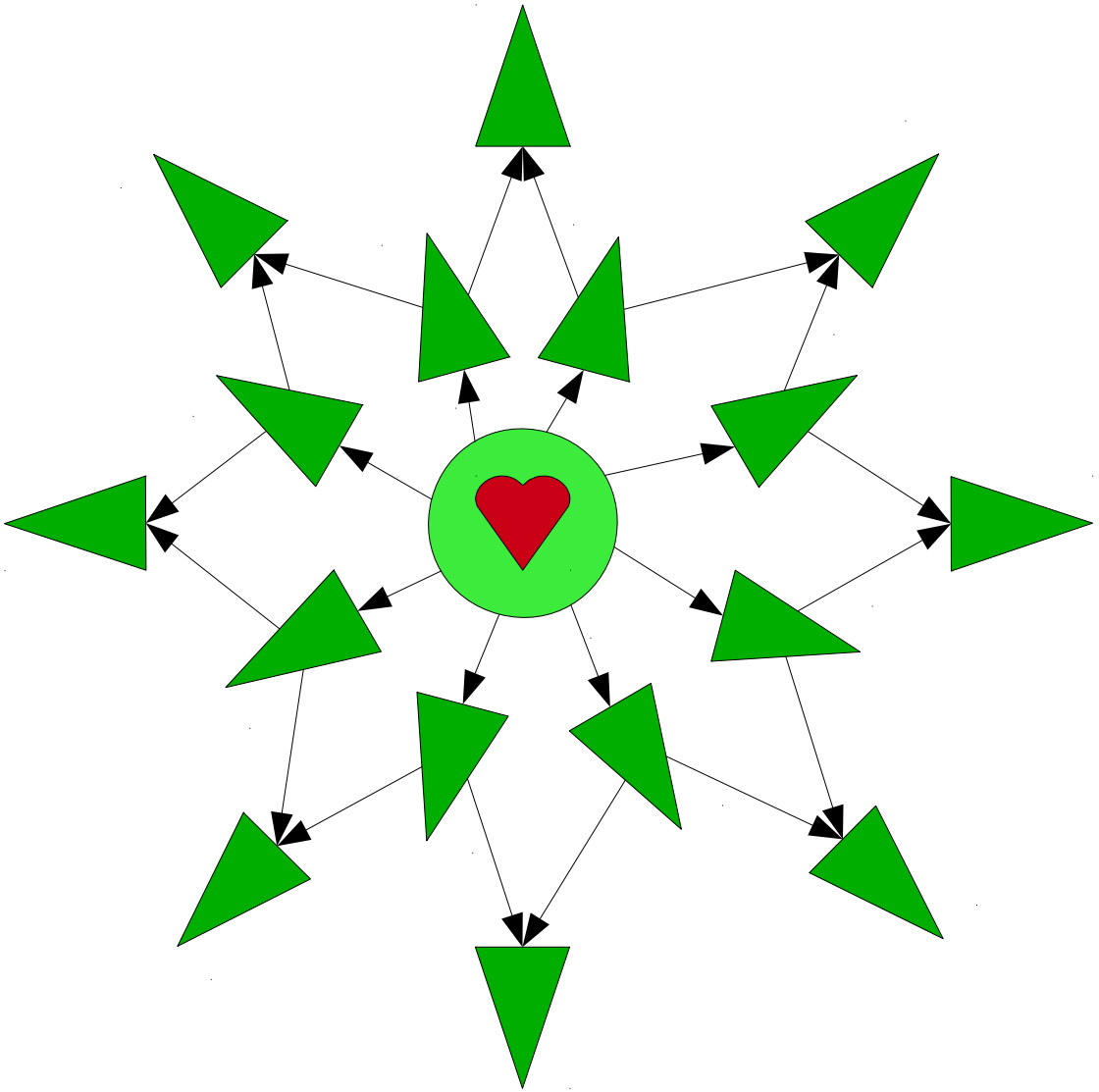
Hasse Diagrams

- A ***Hasse diagram*** is a graphical representation of a partial order.
- No self-loops: by ***reflexivity***, we can always add them back in.
- Higher elements are bigger than lower elements: by ***antisymmetry***, the edges can only go in one direction.
- No redundant edges: by ***transitivity***, we can infer the missing edges.

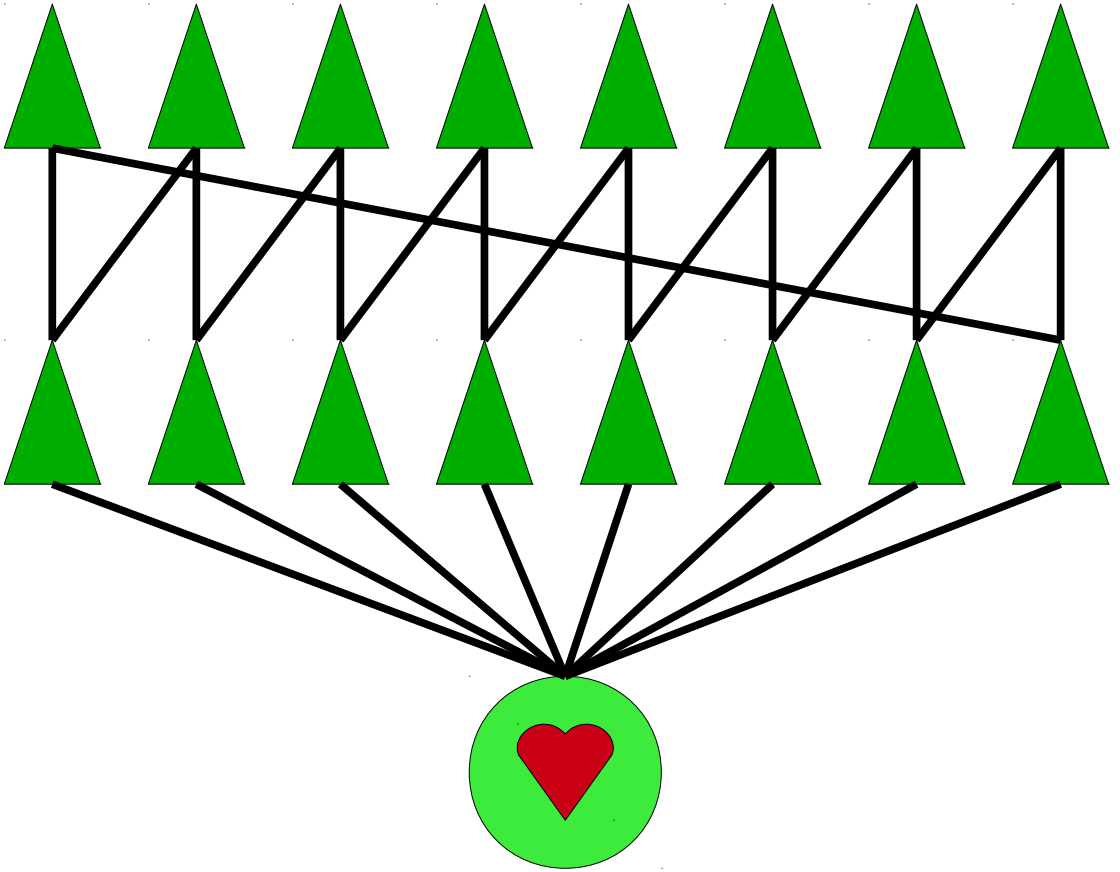

Tasty
Not Tasty
Indian
Mediterranean
Mexican
Chinese
Italian
American
Dorm


Larger
Smaller
...
4
3
2
1
0

Hasse Artichokes



Hasse Artichokes



For More on the Olympics:

http://www.nytimes.com/interactive/2012/08/07/sports/olympics/the-best-and-worst-countries-in-the-medal-count.html

An Important Milestone

Recap: **Discrete Mathematics**

- The past four weeks have focused exclusively on discrete mathematics:

Induction	Functions
Graphs	The Pigeonhole Principle
Relations	Logic
Set Theory	Cardinality

- These are the building blocks we will use throughout the rest of the quarter.
- These are the building blocks you will use throughout the rest of your CS career.

Next Up: **Computability Theory**

- It's time to switch gears and address the limits of what can be computed.
- We'll explore these questions:
  - What is the formal definition of a computer?
  - What might computers look like with various resource constraints?
  - What problems can be solved by computers?
  - What problems ***can't*** be solved by computers?
- ***Get ready to explore the boundaries of what computers could ever be made to do.***

Next Time

- **Formal Language Theory**
  - How are we going to formally model computation?
- **Finite Automata**
  - A simple but powerful computing device made entirely of math!
- **DFAs**
  - A fundamental building block in computing.