Graphs and Relations

Friday Four Square!
Today at 4:15PM outside Gates.

Announcements

- Problem Set 1 due right now.
- Problem Set 2 out.
  - Checkpoint due Monday, April 16.
  - Assignment due Friday, April 20.
  - Play around with induction and its applications!

Mathematical Structures

- Just as there are common data structures in programming, there are common mathematical structures in discrete math.
- So far, we've seen simple structures like sets and natural numbers, but there are many other important structures out there.
- For the next week, we'll explore several of them.

Some Formalisms

Ordered and Unordered Pairs

- An **unordered pair** is a set {a, b} of two elements (remember that sets are unordered).
  - {0, 1} = {1, 0}
- An **ordered pair** (a, b) is a pair of elements in a specific order.
  - (0, 1) ≠ (1, 0).
  - Two ordered pairs are equal iff each of their components are equal.
- An **unordered tuple** is a set $\{a_0, a_1, \ldots, a_{n-1}\}$ of n elements.
- An **ordered tuple** $(a_0, a_1, \ldots, a_{n-1})$ is an collection of n elements in a specific order.
  - This is sometimes called a **sequence**.
  - As with ordered pairs, two ordered tuples are equal iff each of their elements are equal.

The Cartesian Product

- Recall: The **power set** $\wp(S)$ of a set is the set of all its subsets.
- The **Cartesian Product** of A × B of two sets is defined as

$$A \times B \equiv \{\ (a, b) \mid a \in A \text{ and } b \in B\ \}$$

The Cartesian Product

- Recall: The **power set** $\wp(S)$ of a set is the set of all its subsets.
- The **Cartesian Product** of A × B of two sets is defined as

$$A \times B \equiv \{\ (a, b) \mid a \in A \text{ and } b \in B\ \}$$

$$\underset{A}{\{0, 1, 2\}} \qquad \underset{B}{\{a, b, c\}}$$

The Cartesian Product

- Recall: The **power set** $\wp(S)$ of a set is the set of all its subsets.
- The **Cartesian Product** of A × B of two sets is defined as

$$A \times B \equiv \{\ (a, b) \mid a \in A \text{ and } b \in B\ \}$$

$$\underset{A}{\left\{ 0, 1, 2 \right\}} \times \underset{B}{\left\{ a, b, c \right\}} =$$

The Cartesian Product

- Recall: The **power set** $\wp(S)$ of a set is the set of all its subsets.
- The **Cartesian Product** of A × B of two sets is defined as

$$A \times B \equiv \{\ (a, b) \mid a \in A \text{ and } b \in B\ \}$$

$$\underset{A}{\{0, 1, 2\}} \times \underset{B}{\{a, b, c\}} =$$

	a	b	c
0			
1			
2			

The Cartesian Product

- Recall: The **power set** $\wp(S)$ of a set is the set of all its subsets.
- The **Cartesian Product** of A × B of two sets is defined as

$$A \times B \equiv \{ (a, b) \mid a \in A \text{ and } b \in B \}$$

$$\underset{A}{\{0, 1, 2\}} \times \underset{B}{\{a, b, c\}} =$$

	a	b	c
0	(0, a)	(0, b)	(0, c)
1	(1, a)	(1, b)	(1, c)
2	(2, a)	(2, b)	(2, c)

The Cartesian Product

- Recall: The **power set** $\wp(S)$ of a set is the set of all its subsets.
- The **Cartesian Product** of A × B of two sets is defined as

$$A \times B \equiv \{\ (a, b) \mid a \in A \text{ and } b \in B\ \}$$

$$\underbrace{\{0, 1, 2\}}_{A} \times \underbrace{\{a, b, c\}}_{B} = \left\{ \begin{array}{l} (0, a), (0, b), (0, c), \\ (1, a), (1, b), (1, c), \\ (2, a), (2, b), (2, c) \end{array} \right\}$$

The Cartesian Product

- Recall: The **power set** $\wp(S)$ of a set is the set of all its subsets.
- The **Cartesian Product** of A × B of two sets is defined as

$$A \times B \equiv \{\ (a, b) \mid a \in A \text{ and } b \in B\ \}$$

- We denote $A^k \equiv \underbrace{A \times A \times \ldots \times A}_{k \text{ times}}$

$$\underset{A}{\{\,0, 1, 2\,\}} \times \underset{B}{\{\,a, b, c\,\}} = \left\{ \begin{array}{l} (0, a), (0, b), (0, c), \\ (1, a), (1, b), (1, c), \\ (2, a), (2, b), (2, c) \end{array} \right\}$$

The Cartesian Product

- Recall: The **power set** $\wp(S)$ of a set is the set of all its subsets.
- The **Cartesian Product** of A × B of two sets is defined as

$$A \times B \equiv \{\ (a, b) \mid a \in A \text{ and } b \in B\ \}$$

- We denote $A^k \equiv \underbrace{A \times A \times \ \dots \ \times A}_{k \text{ times}}$

$$\underset{A}{\{0, 1, 2\}} \times \underset{A}{\{0, 1, 2\}} = \left\{ \begin{array}{l} (0, 0),\ (0, 1),\ (0, 2), \\ (1, 0),\ (1, 1),\ (1, 2), \\ (2, 0),\ (2, 1),\ (2, 2) \end{array} \right\}$$

The Cartesian Product

- Recall: The **power set** $\wp(S)$ of a set is the set of all its subsets.
- The **Cartesian Product** of A × B of two sets is defined as

$$A \times B \equiv \{\ (a, b) \mid a \in A \text{ and } b \in B\ \}$$

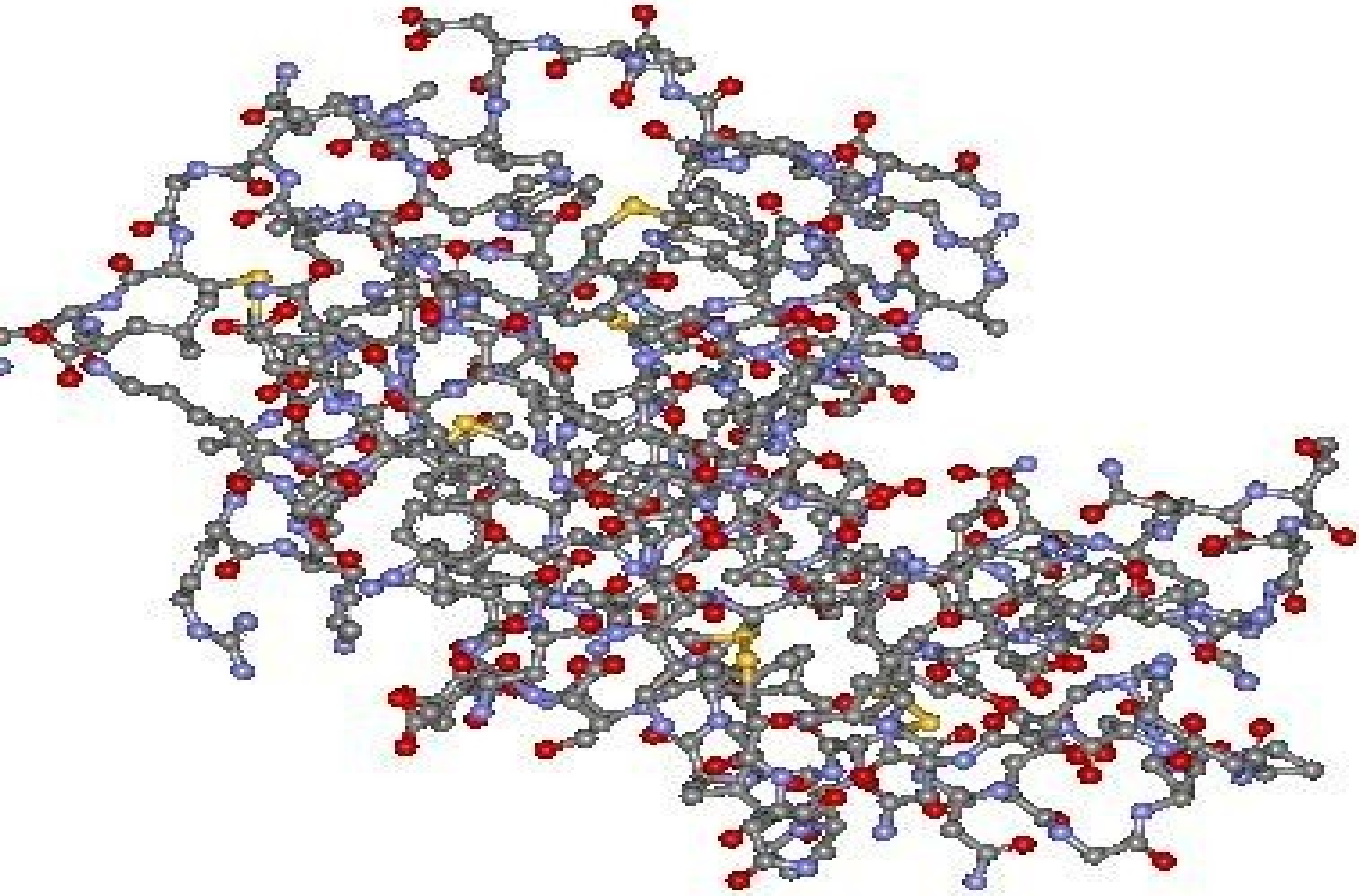
- We denote $A^k \equiv \underbrace{A \times A \times \ \dots\ \times A}_{\text{k times}}$

$$\underbrace{\{0, 1, 2\}}_{A}{}^{2} = \left\{ \begin{array}{lll} (0, 0), & (0, 1), & (0, 2), \\ (1, 0), & (1, 1), & (1, 2), \\ (2, 0), & (2, 1), & (2, 2) \end{array} \right\}$$

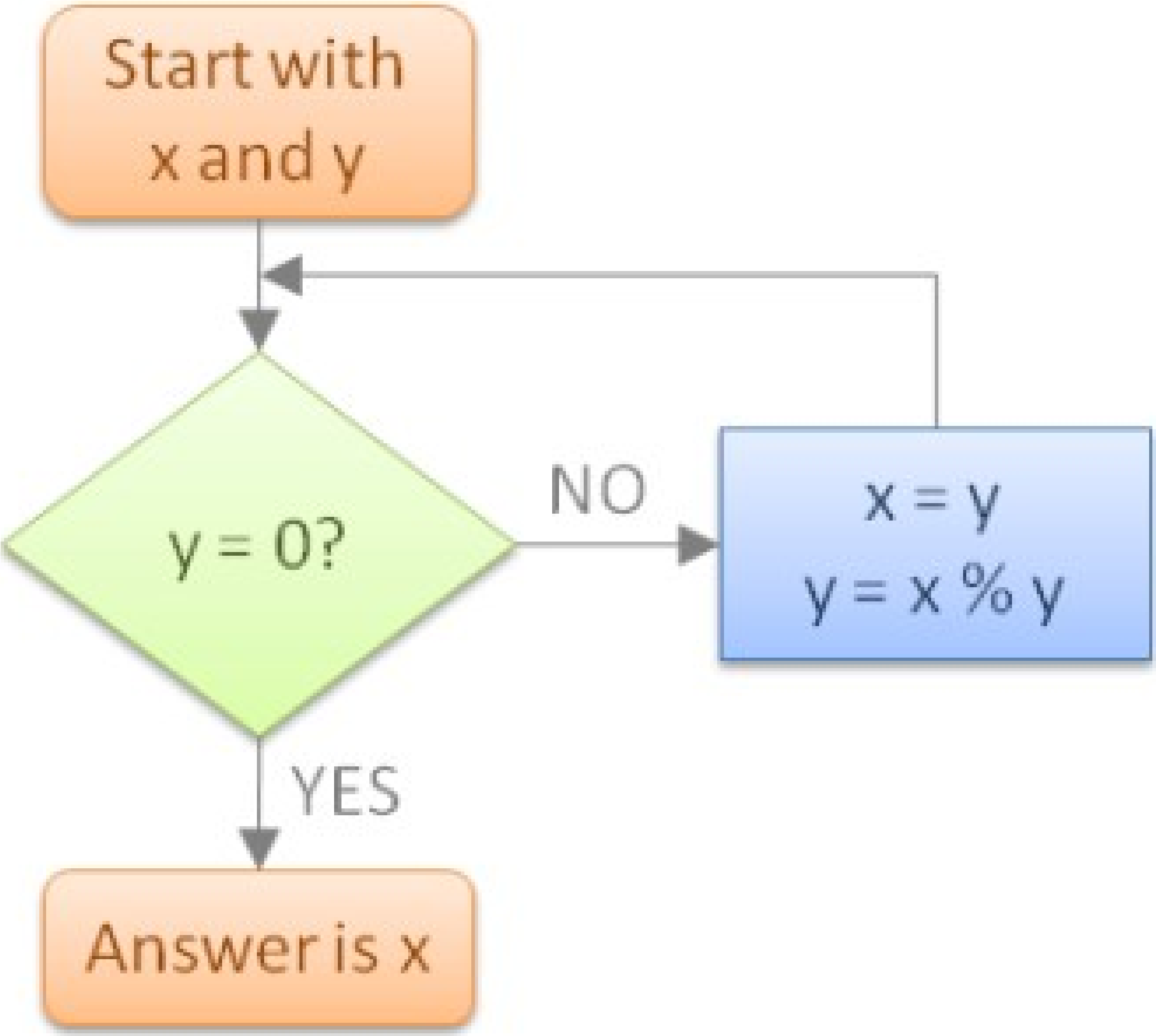
Graphs

A **graph** is a mathematical structure for representing **relationships**.

Each graph is a set of **vertices** (or **nodes**) connected by **edges** (or **arcs**).



http://lysozyme.co.uk/results/lysozyme-ball-stick-stereo.jpg



Start with x and y

y = 0?

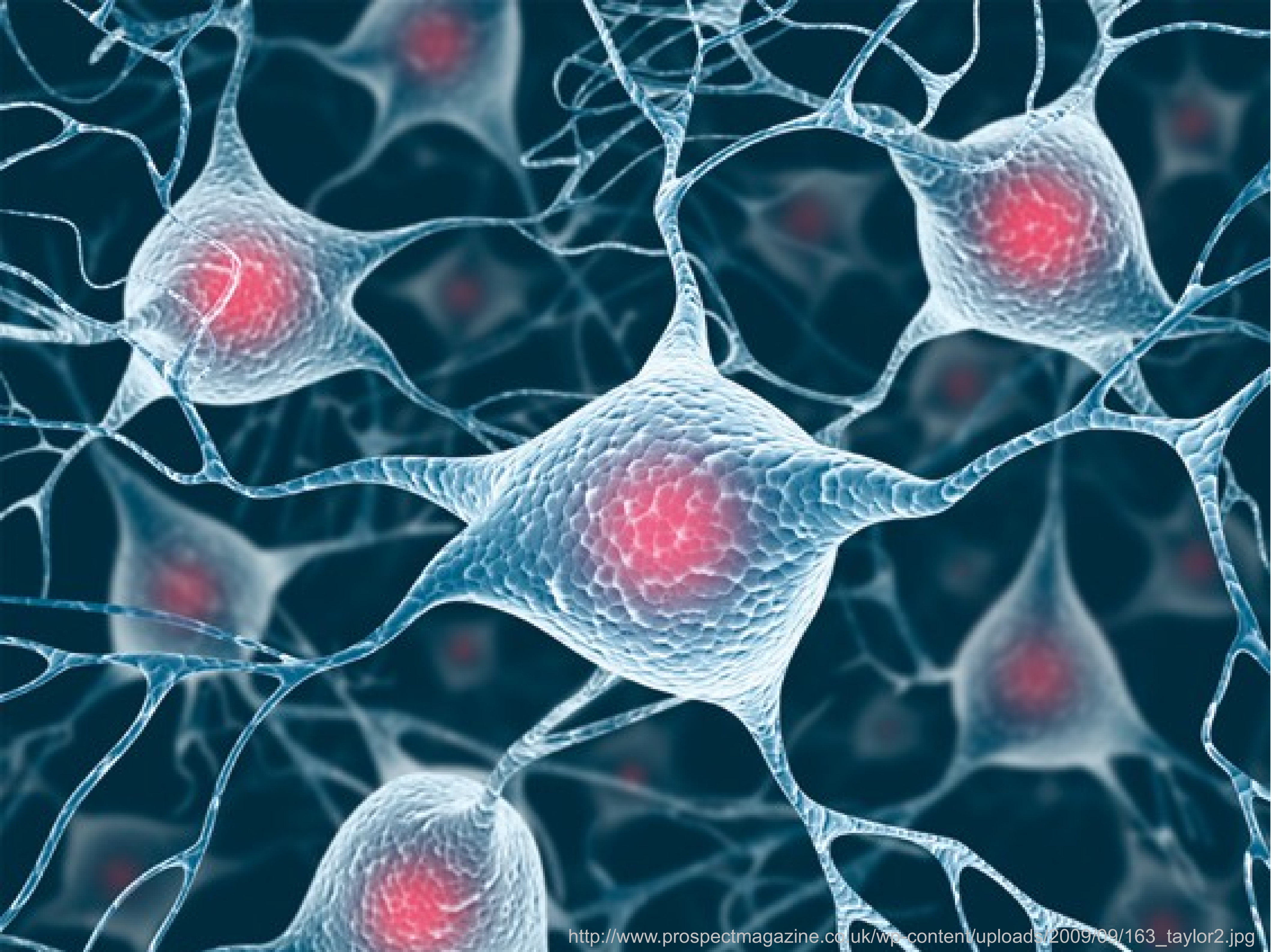
NO

x = y
y = x % y

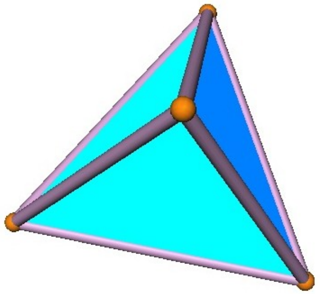
YES

Answer is x

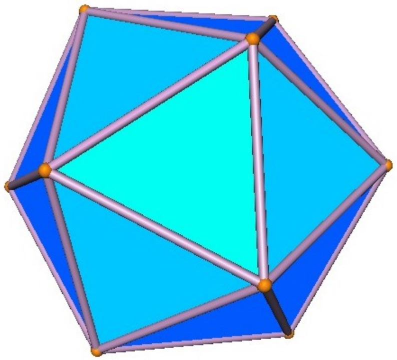
http://diditwith.net/content/binary/WindowsLiveWriter/YAPESProblemFive_C99B/EuclideanAlgorithm_3.png



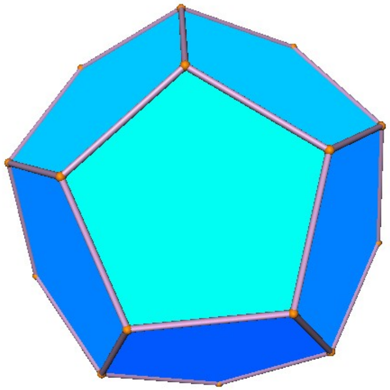
http://www.prospectmagazine.co.uk/wp-content/uploads/2009/09/163_taylor2.jpg



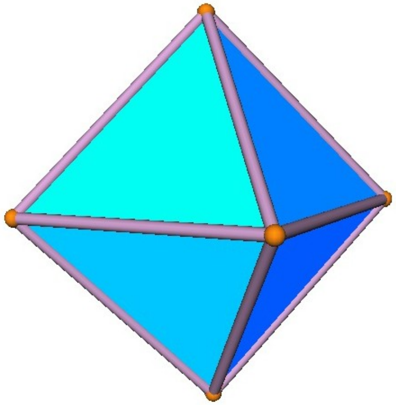
Tetrahedron



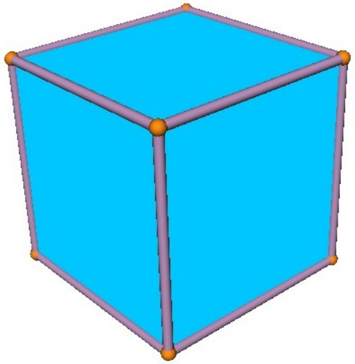
Icosahedron



Dodecahedron



Octahedron

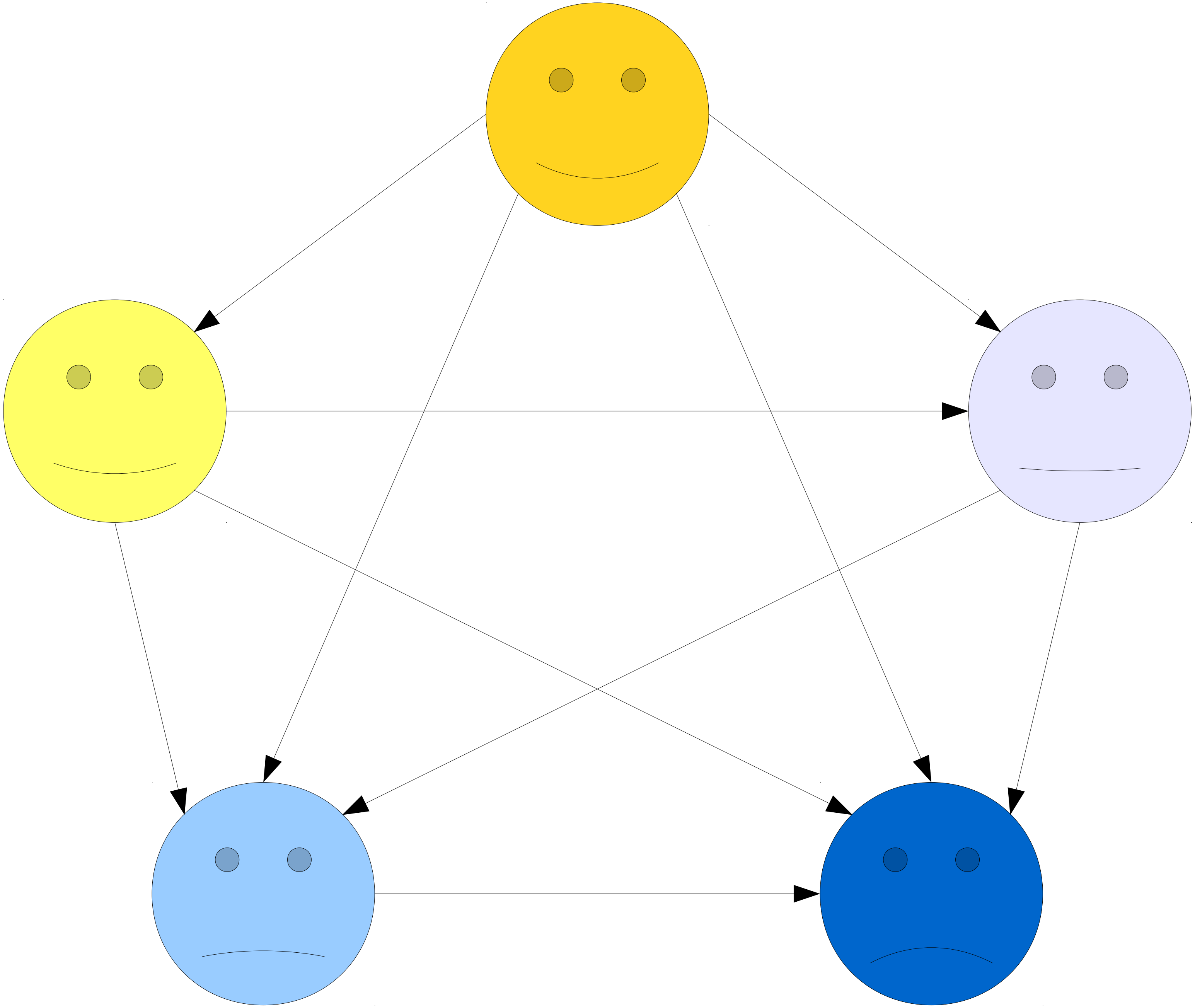


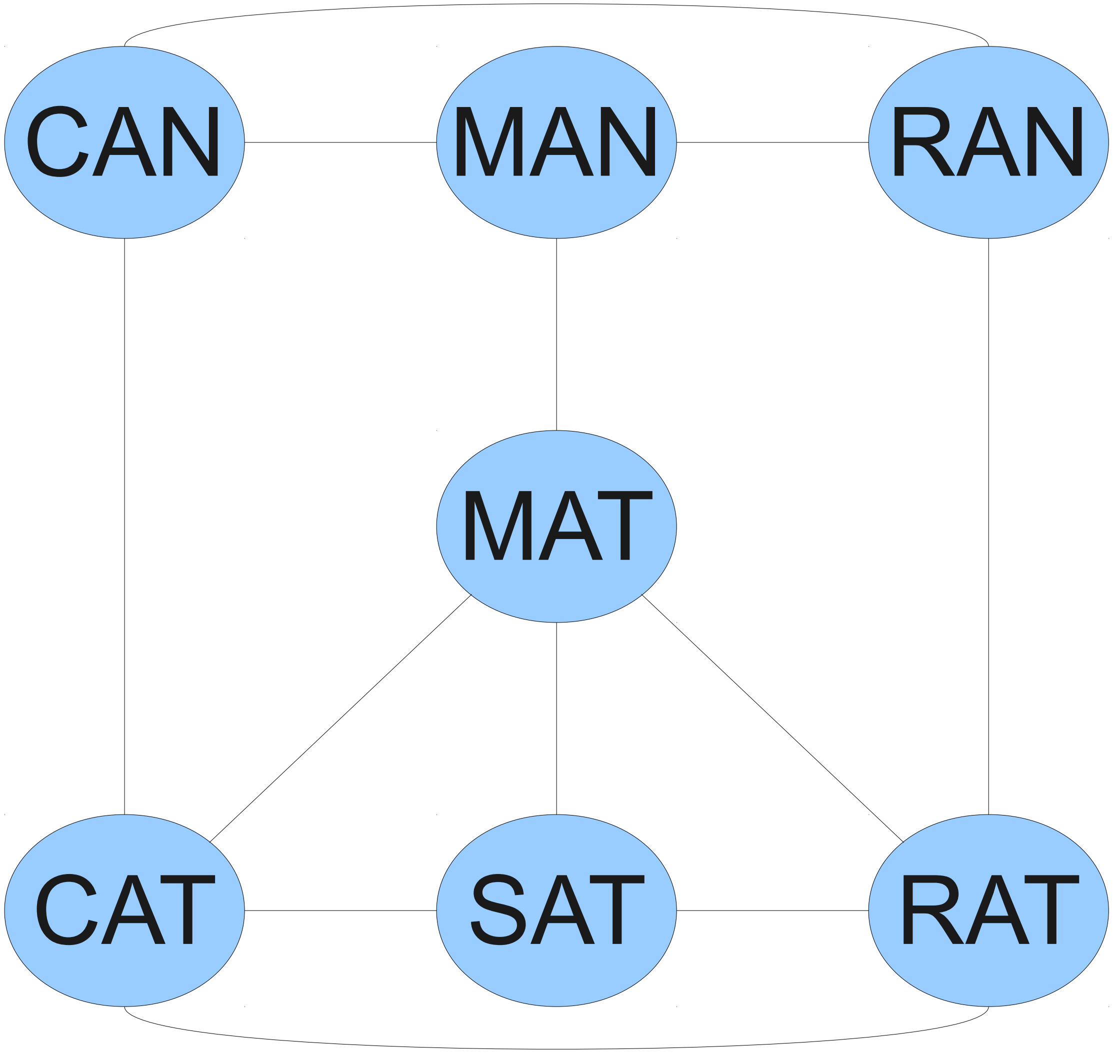
Cube

http://www.princeton.edu/pr/pictures/l-r/packingproblem/pu-platonic-solids.jpg

Formalisms

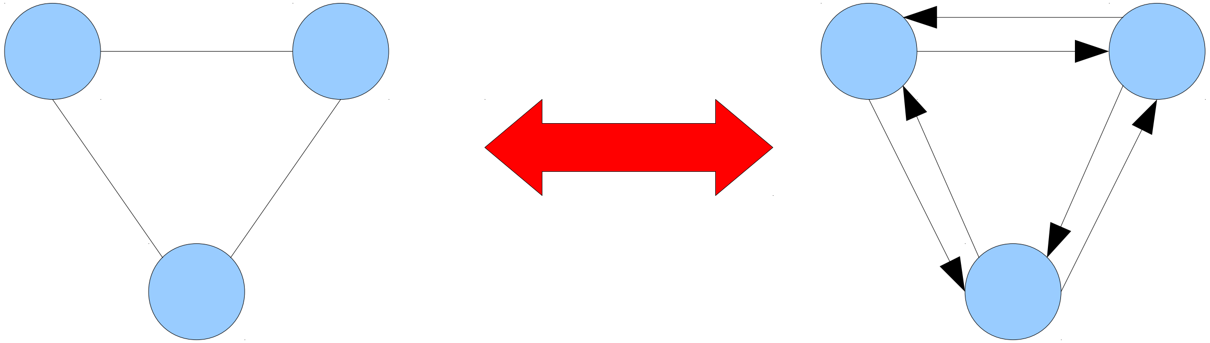
- A **graph** is an ordered pair G = (V, E) where
  - V is a set of the **vertices** (nodes) of the graph.
  - E is a set of the **edges** (arcs) of the graph.
- Each edge is an pair of the **start** and **end** (or **source** and **sink**) of the edge.



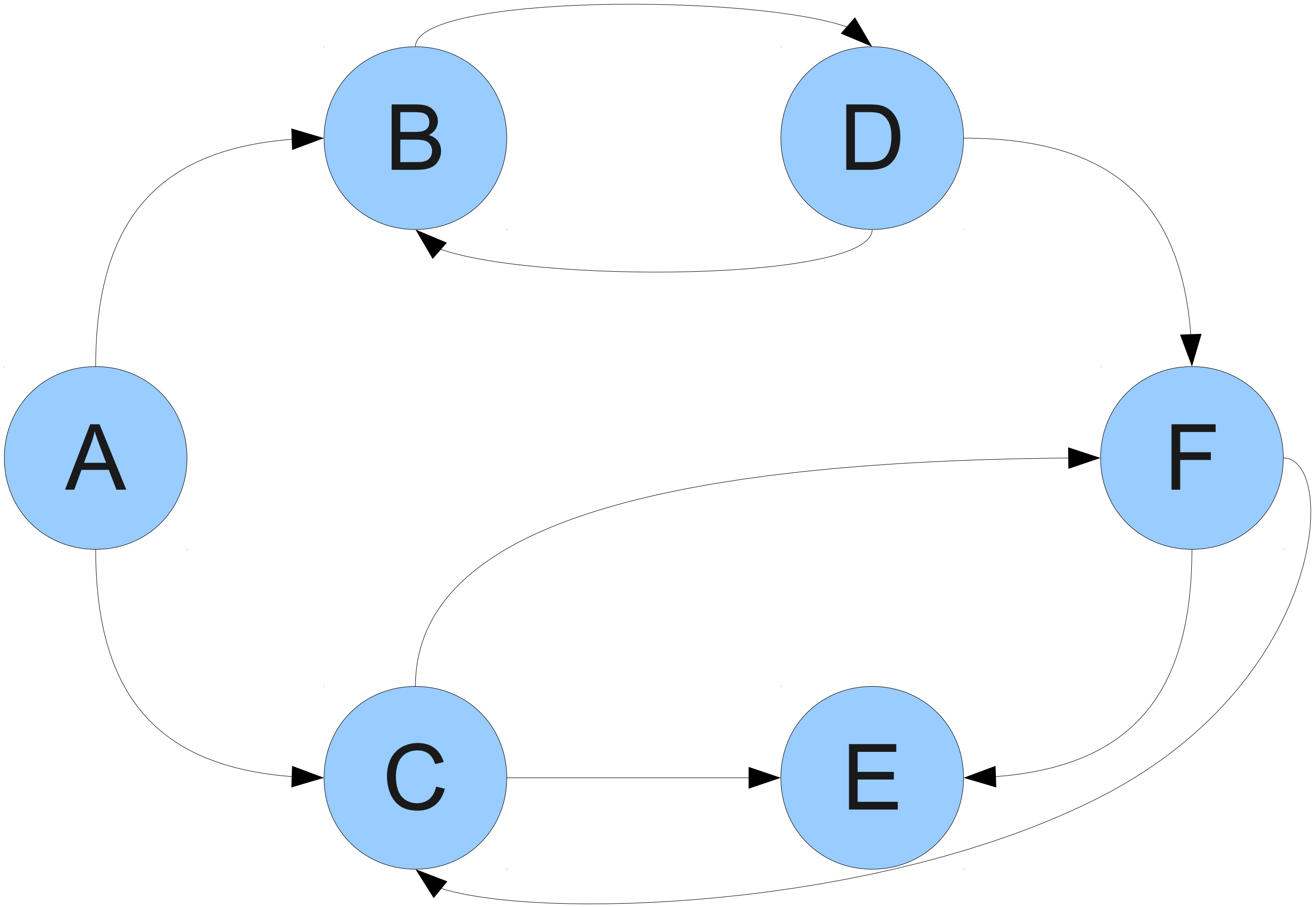

CAN
MAN
RAN
MAT
CAT
SAT
RAT

Directed and Undirected Graphs

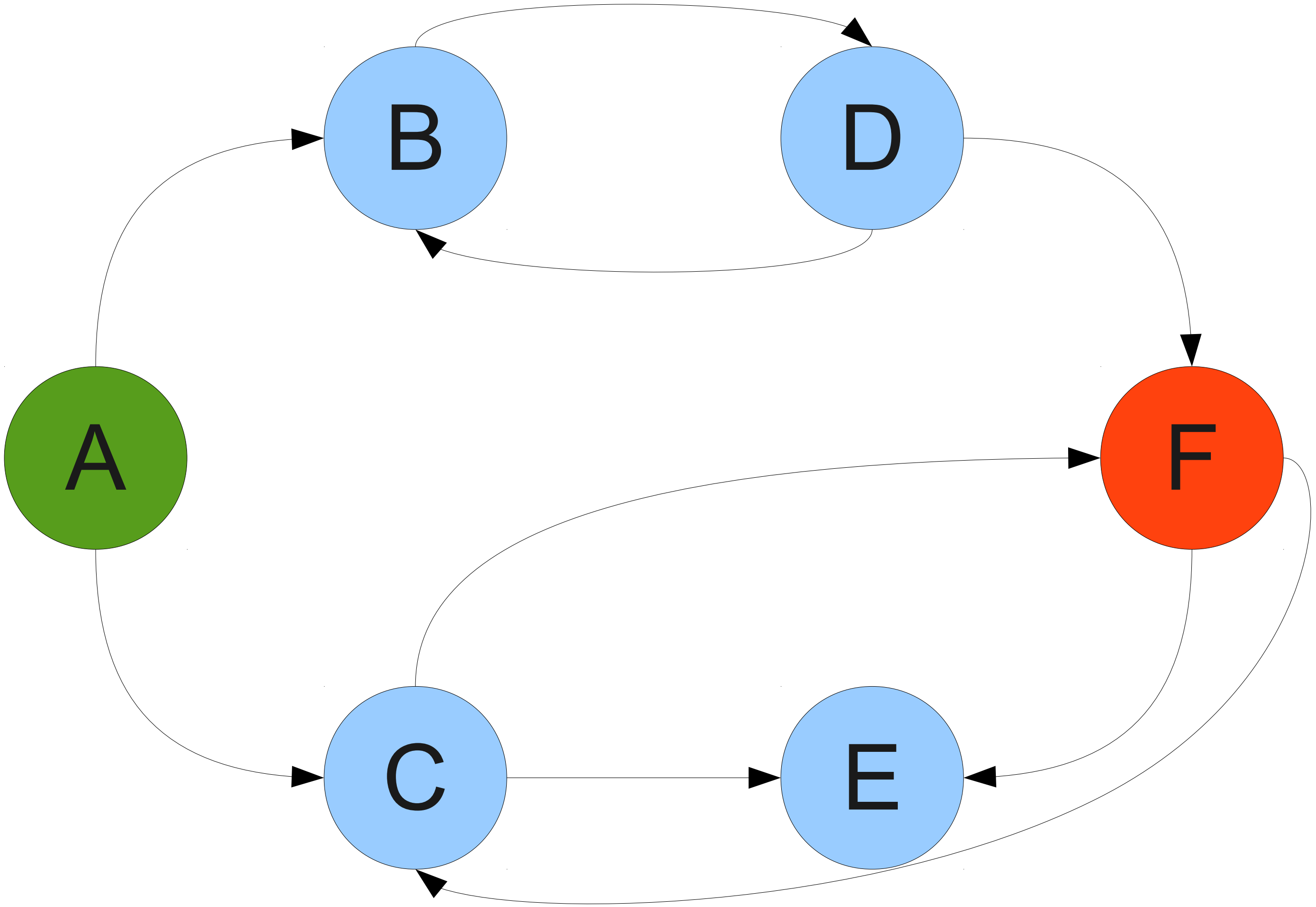
- A graph is **directed** if its edges are ordered pairs.
- A graph is **undirected** if the edges are unordered pairs.
- An undirected graph is a special case of a directed graph (just add edges both ways).



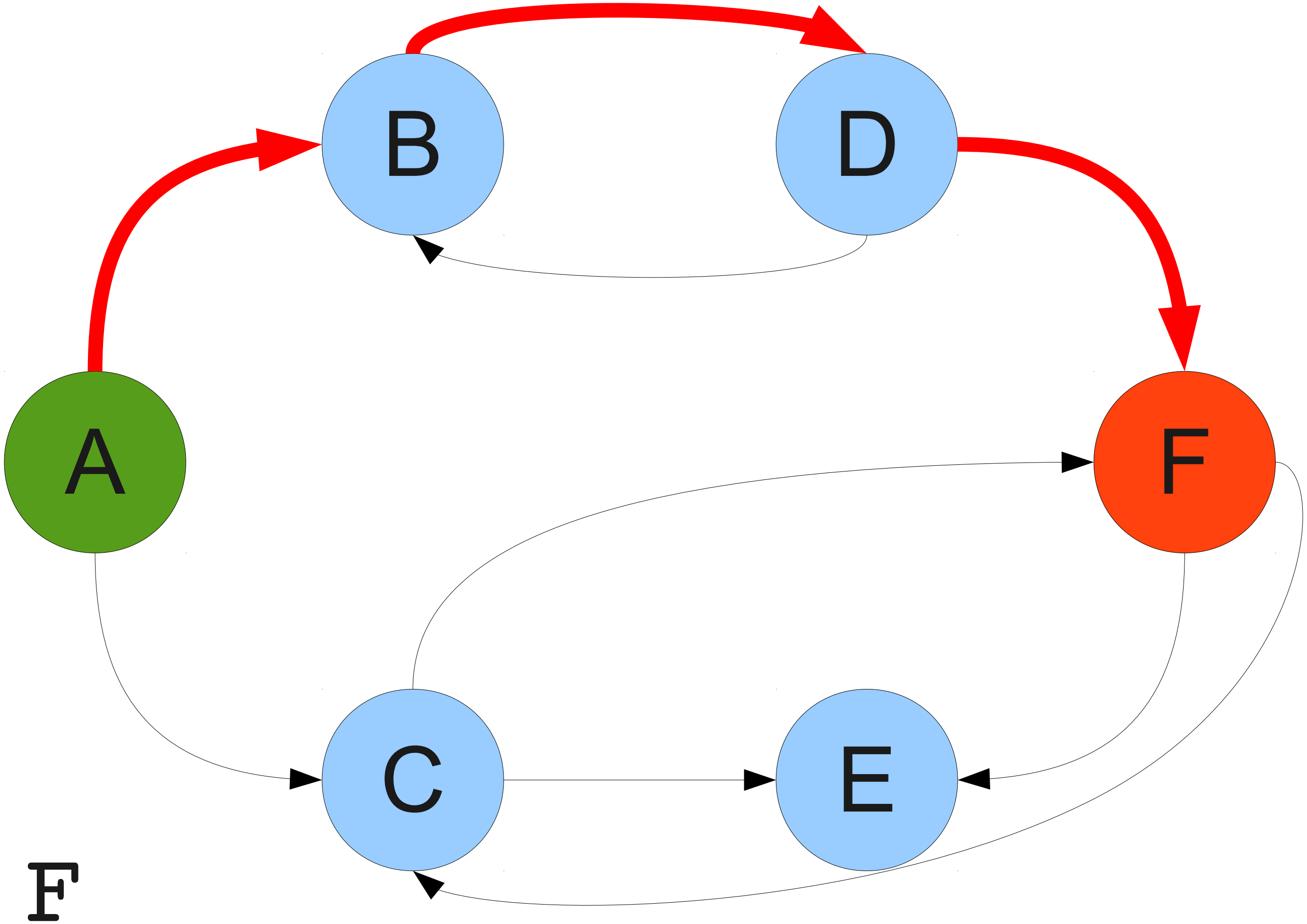
Navigating a Graph



Navigating a Graph

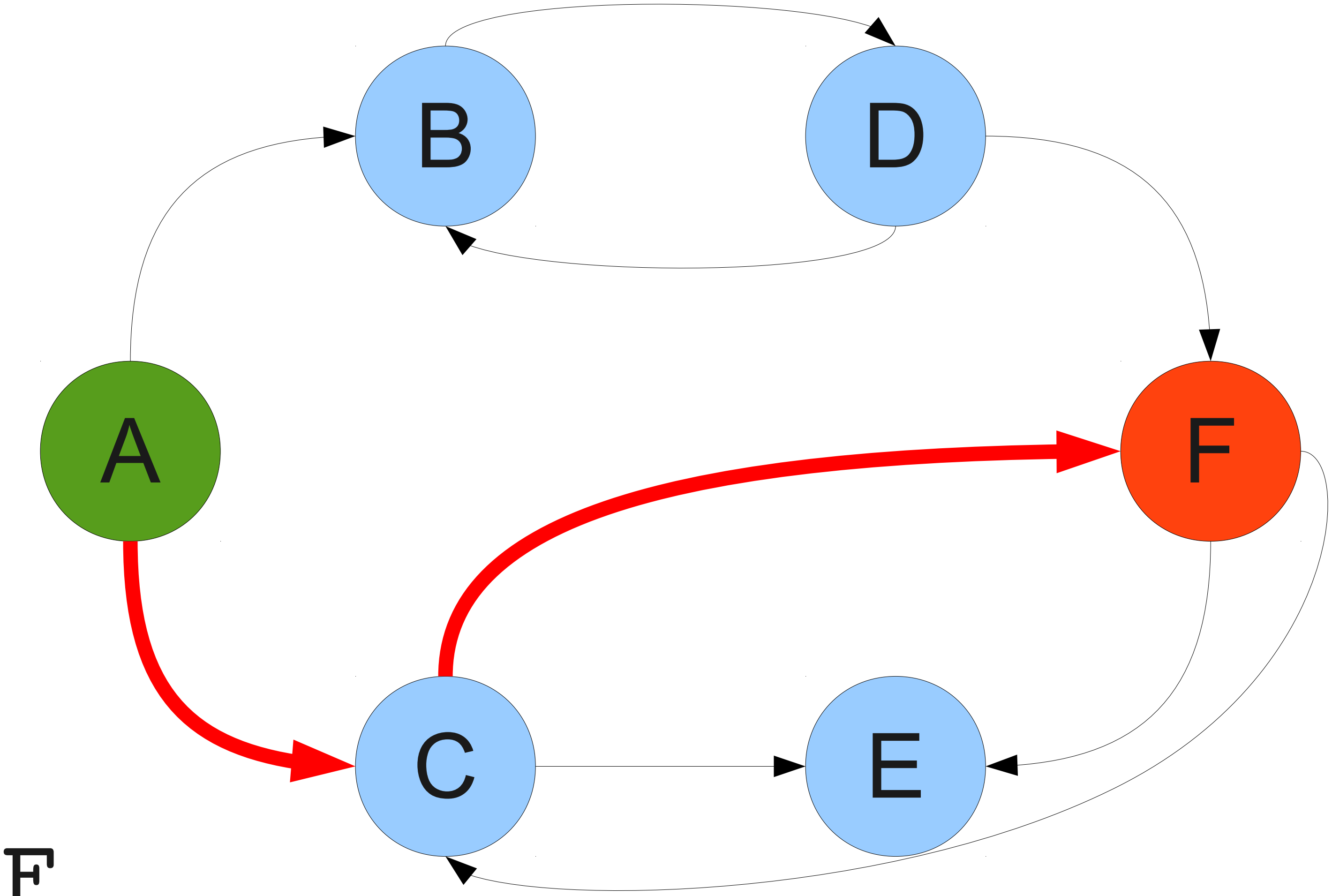


Navigating a Graph



A B D F

Navigating a Graph

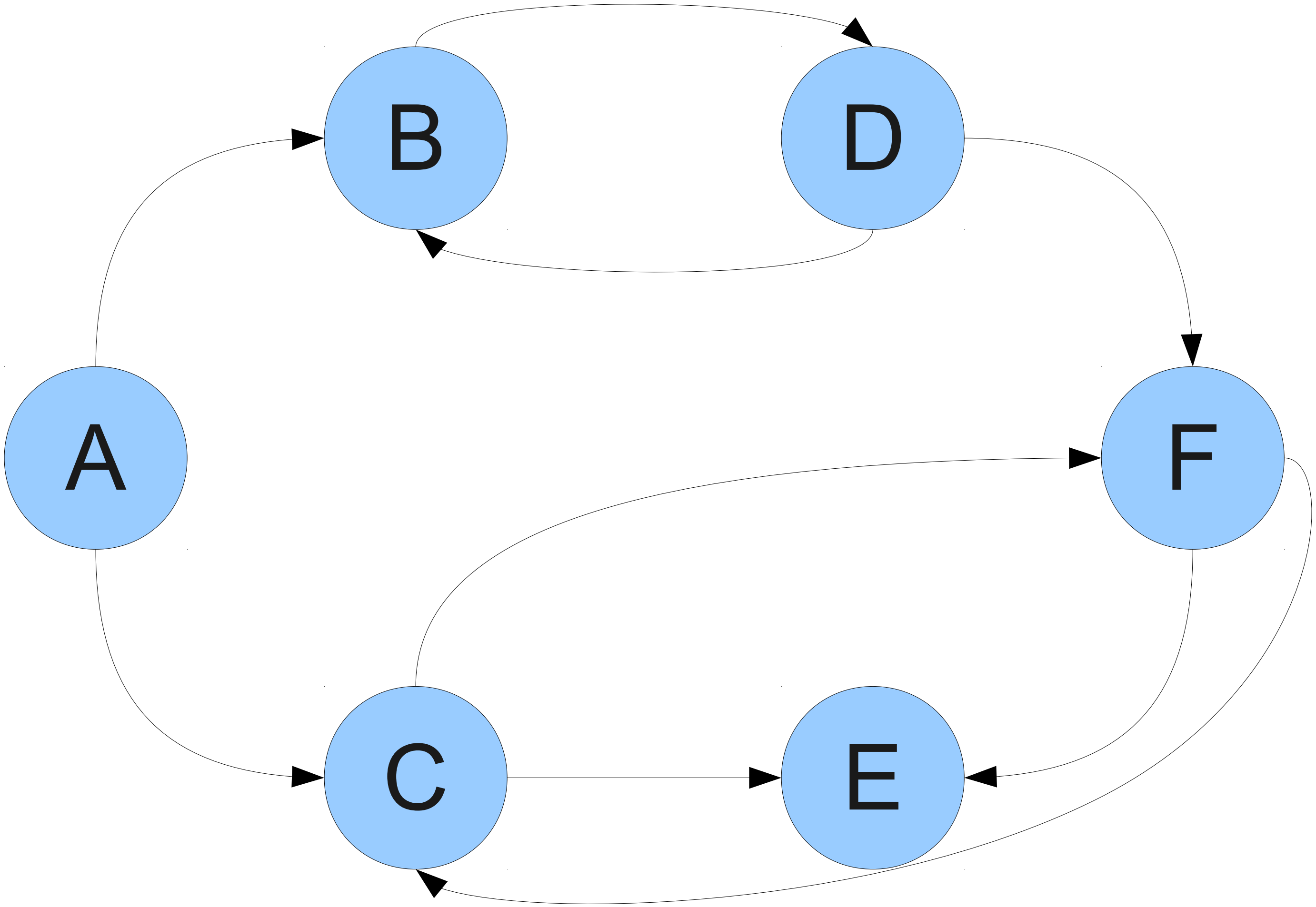


A C F

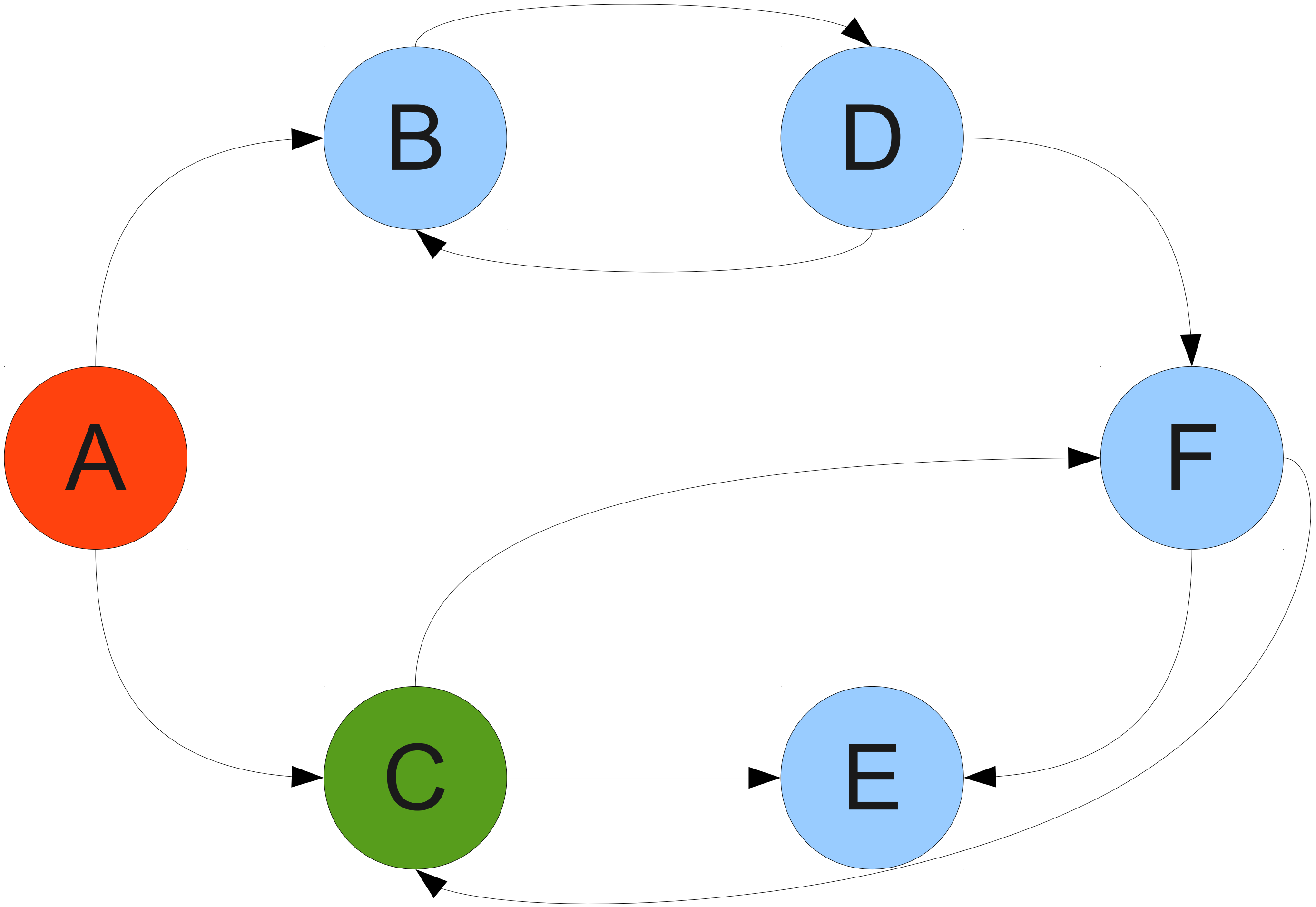
A **path** from v_0 to v_n is a sequence of edges $((v_0, v_1), (v_1, v_2), \ldots, (v_{n-1}, v_n))$.

The **length** of a path is the number of edges it contains.

Navigating a Graph

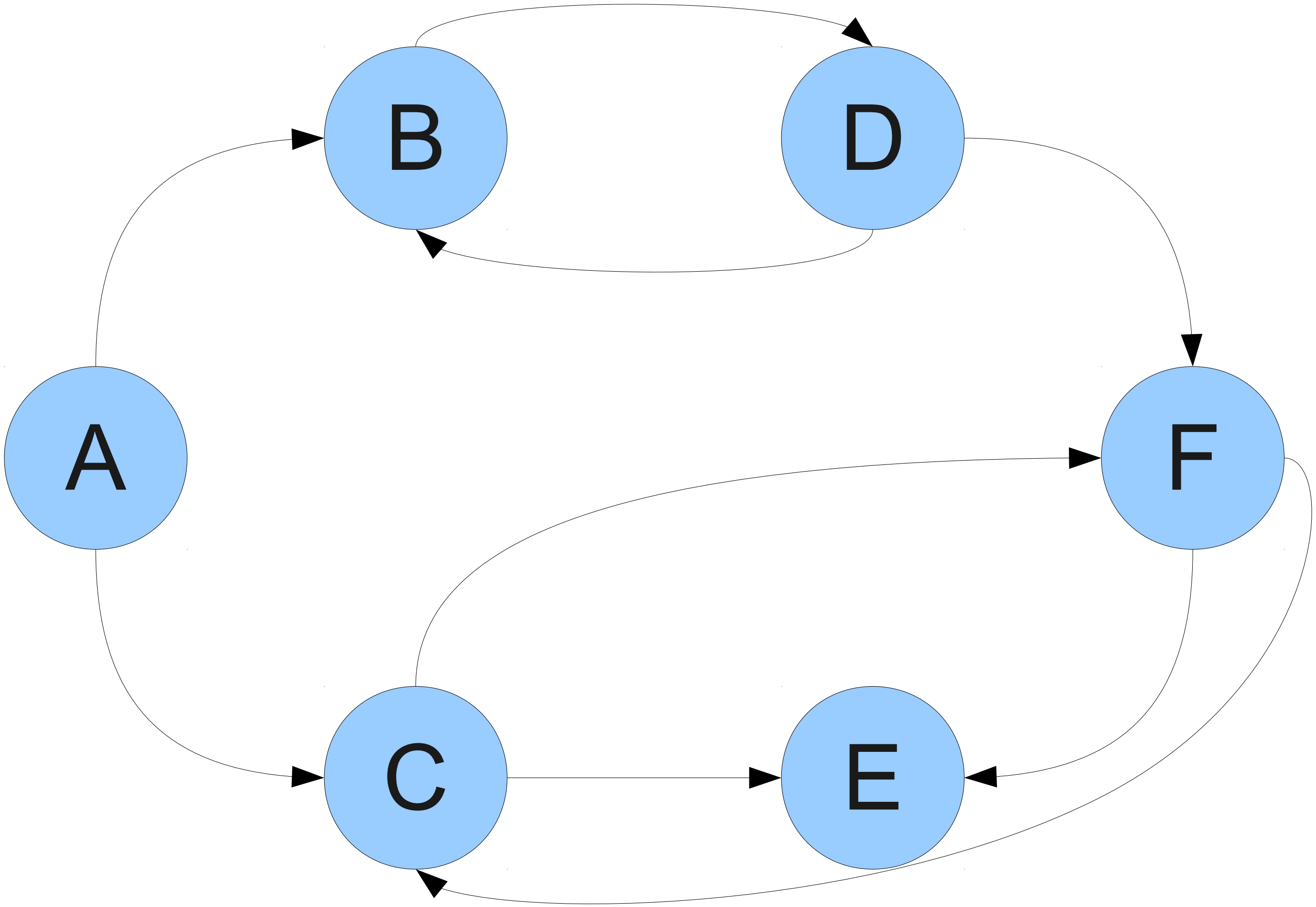


Navigating a Graph

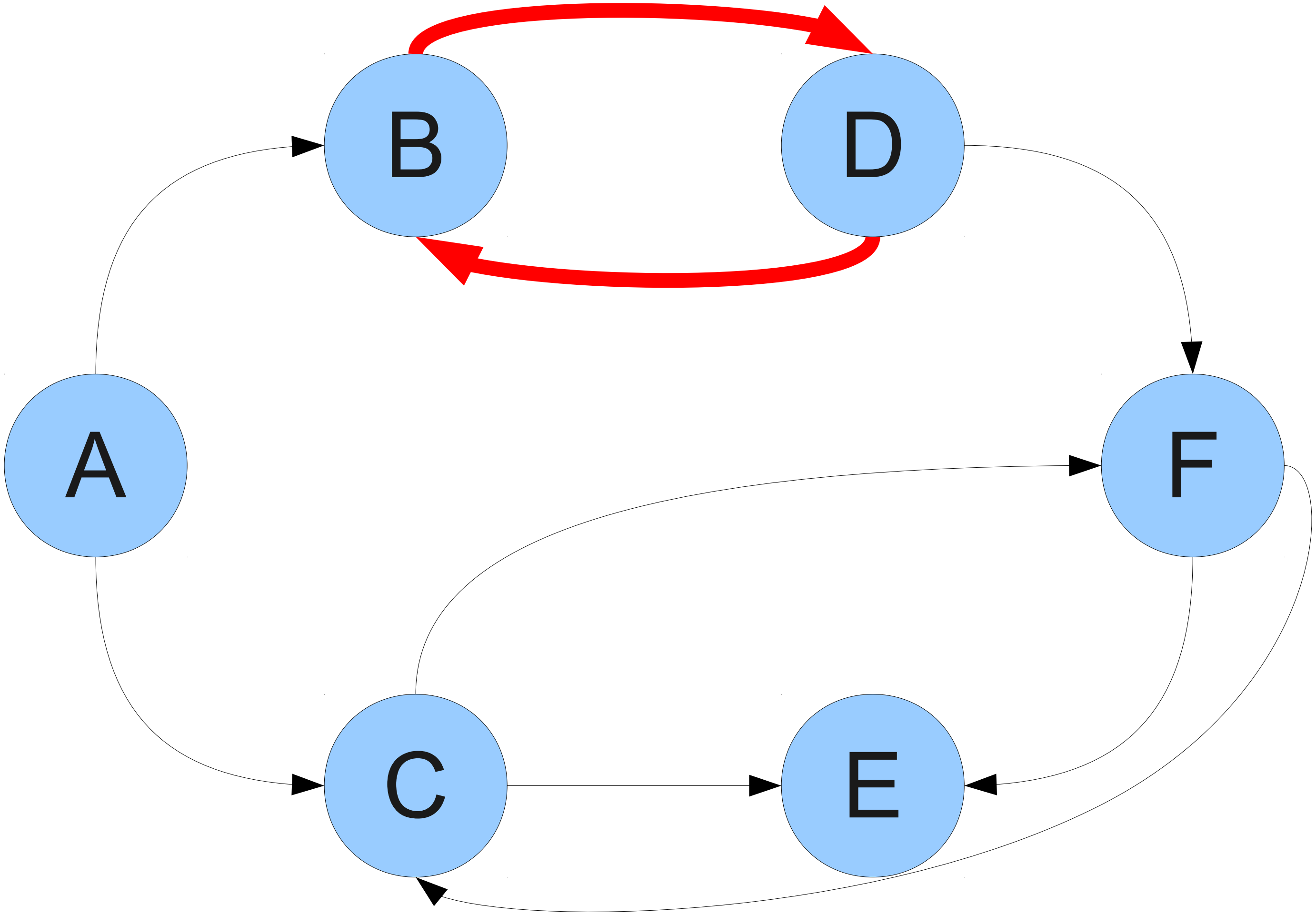


A node v is **reachable** from node u if there is a path from u to v.

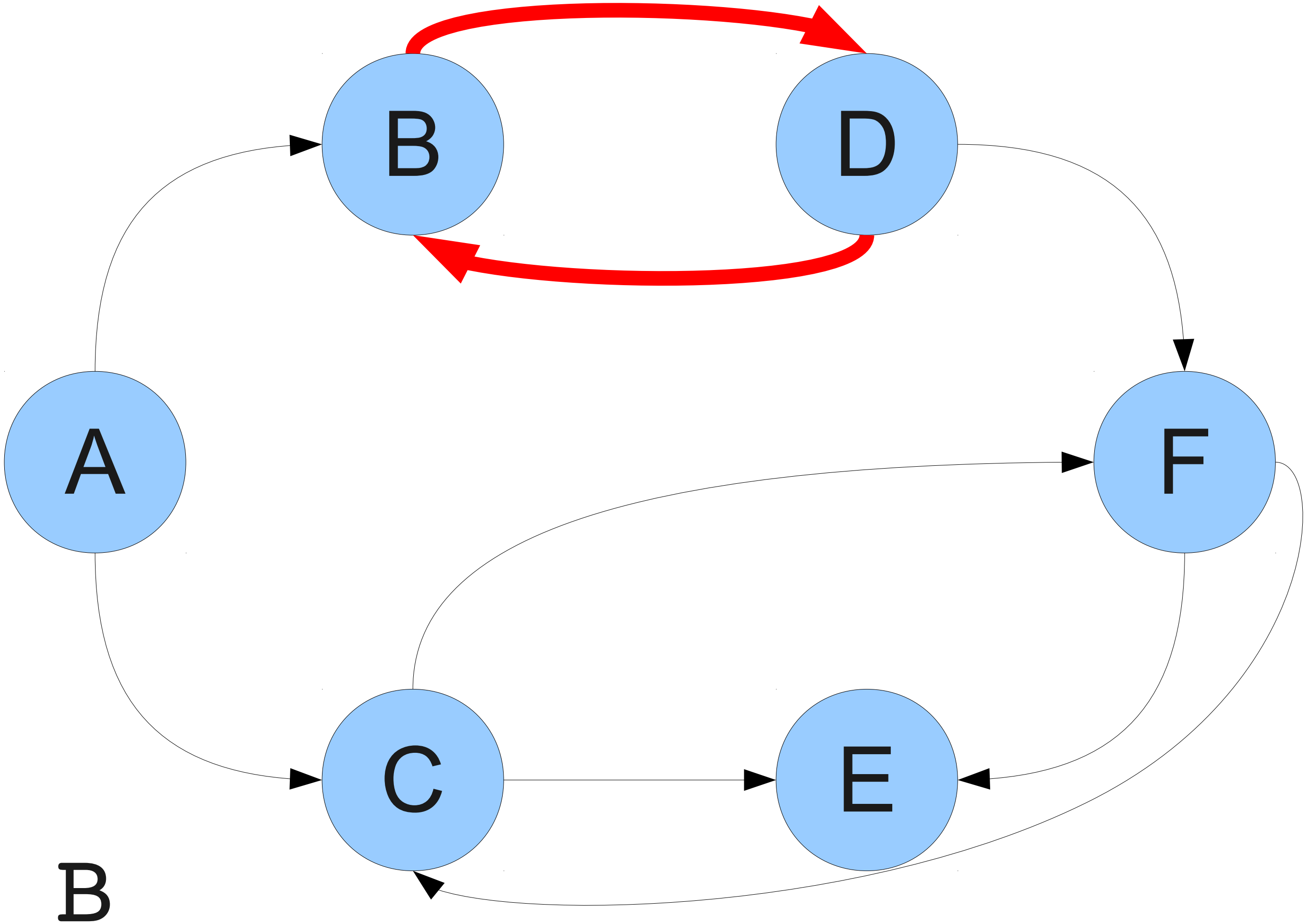
Navigating a Graph



Navigating a Graph

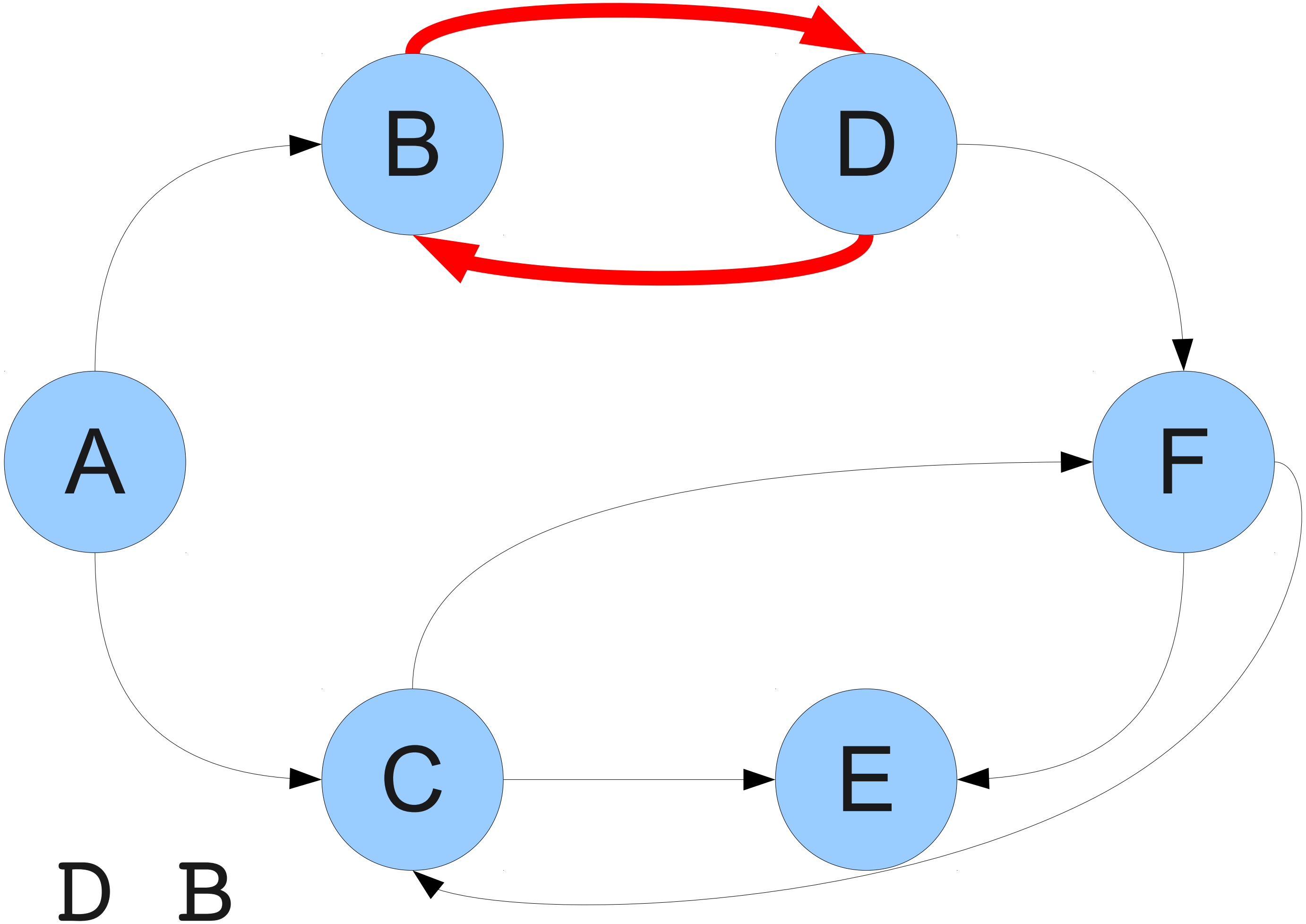


Navigating a Graph



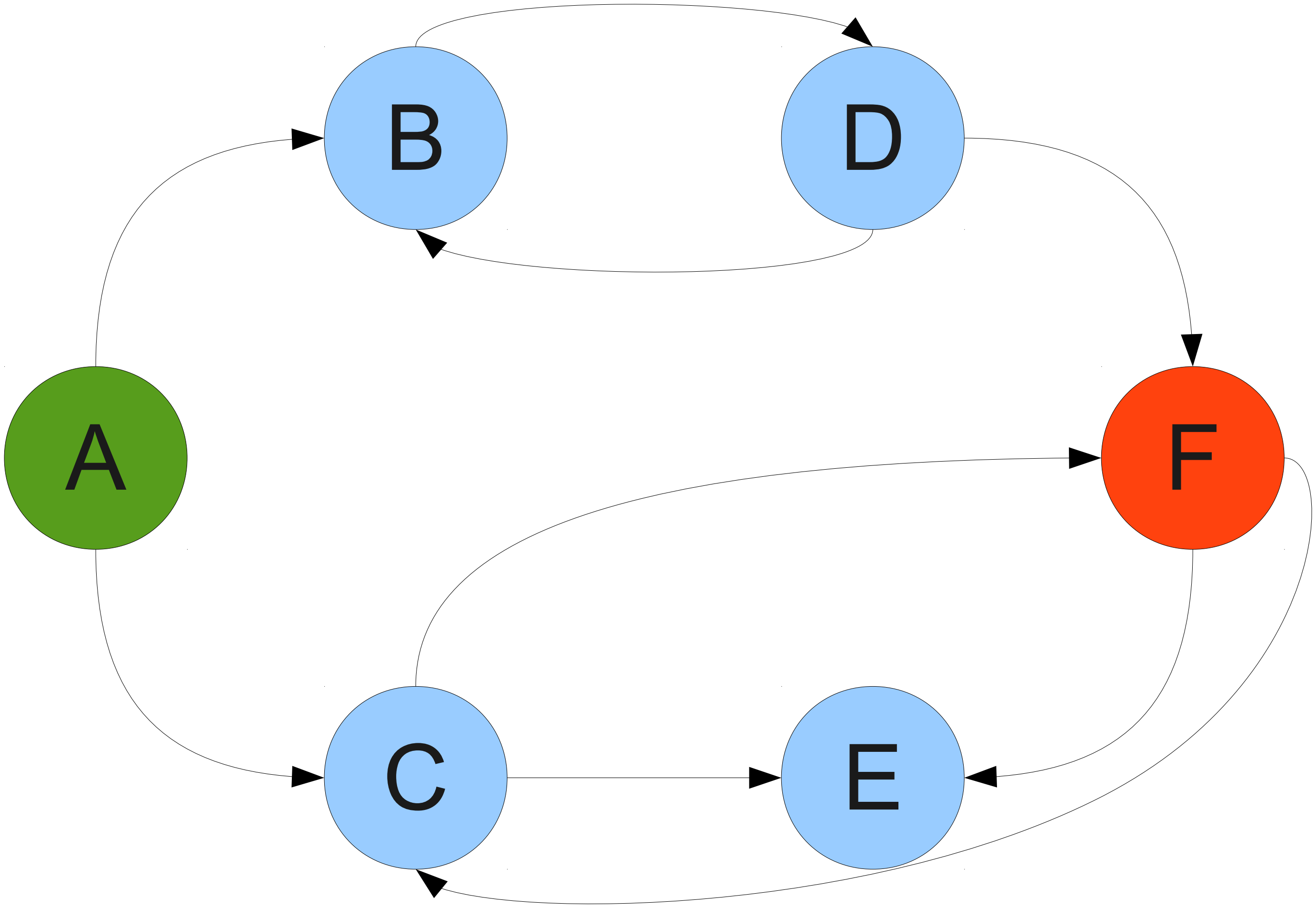
B D B

Navigating a Graph

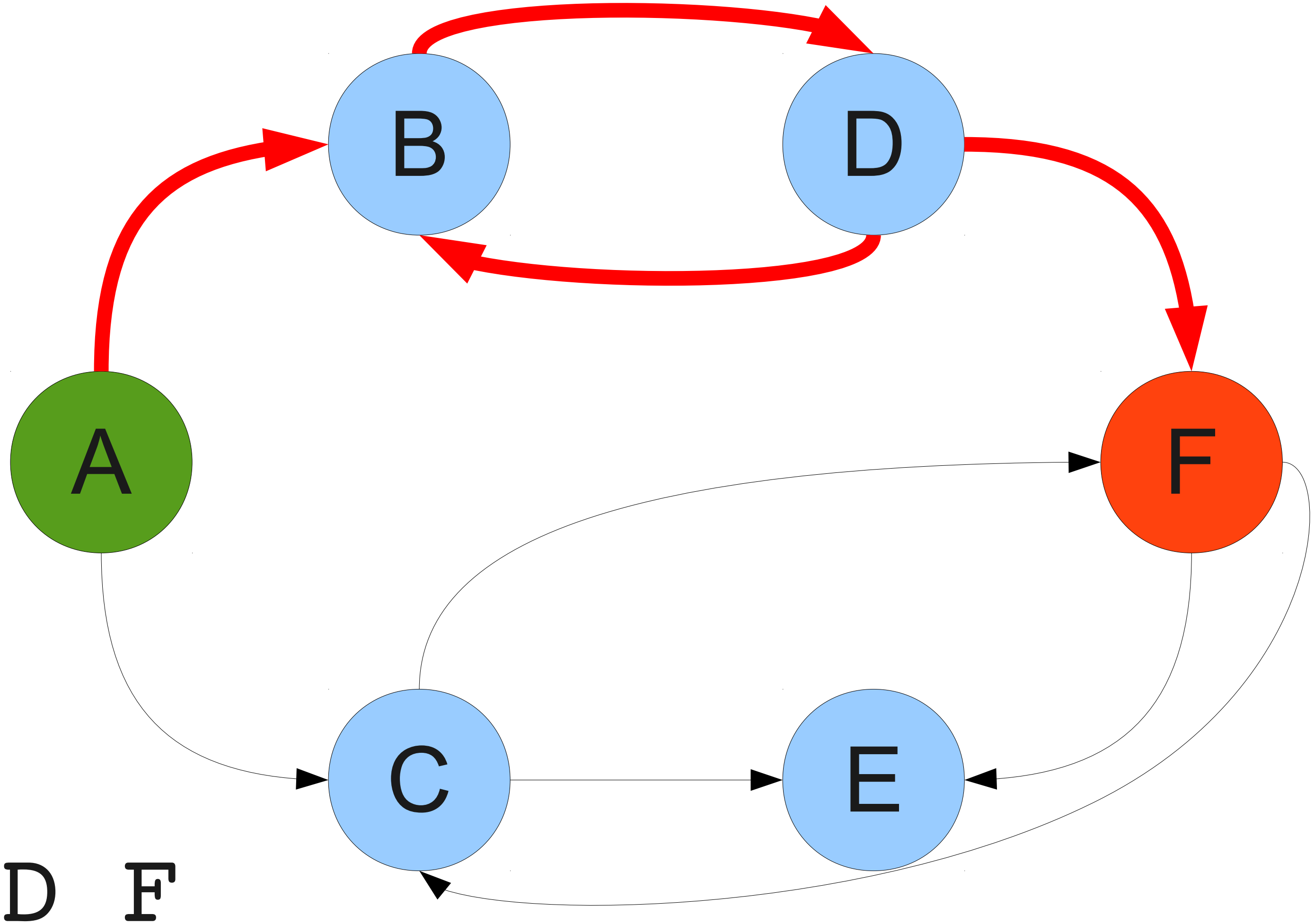


B D B D B

Navigating a Graph

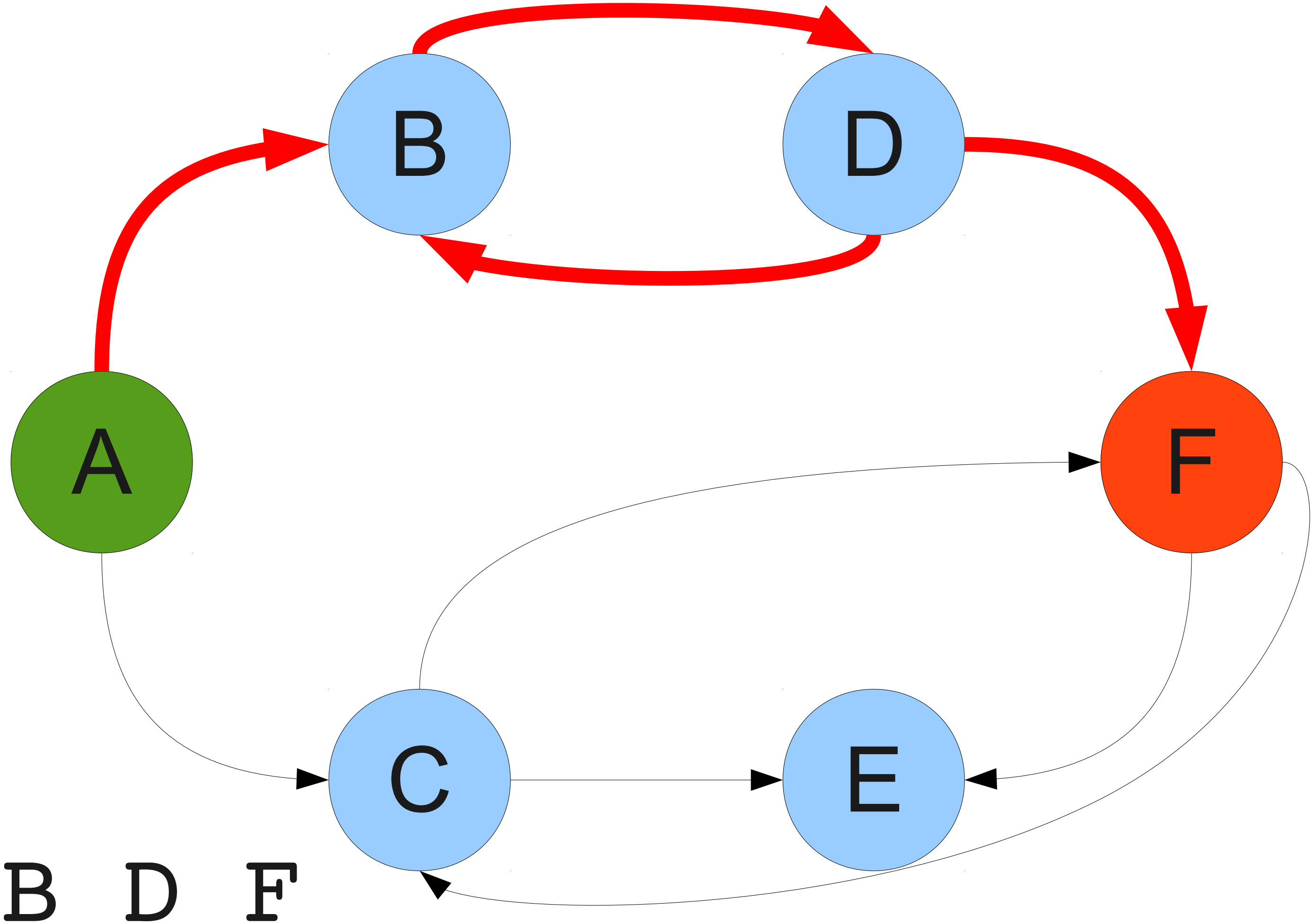


Navigating a Graph



A B D F

Navigating a Graph



A B D B D F

A **cycle** in a graph is a path

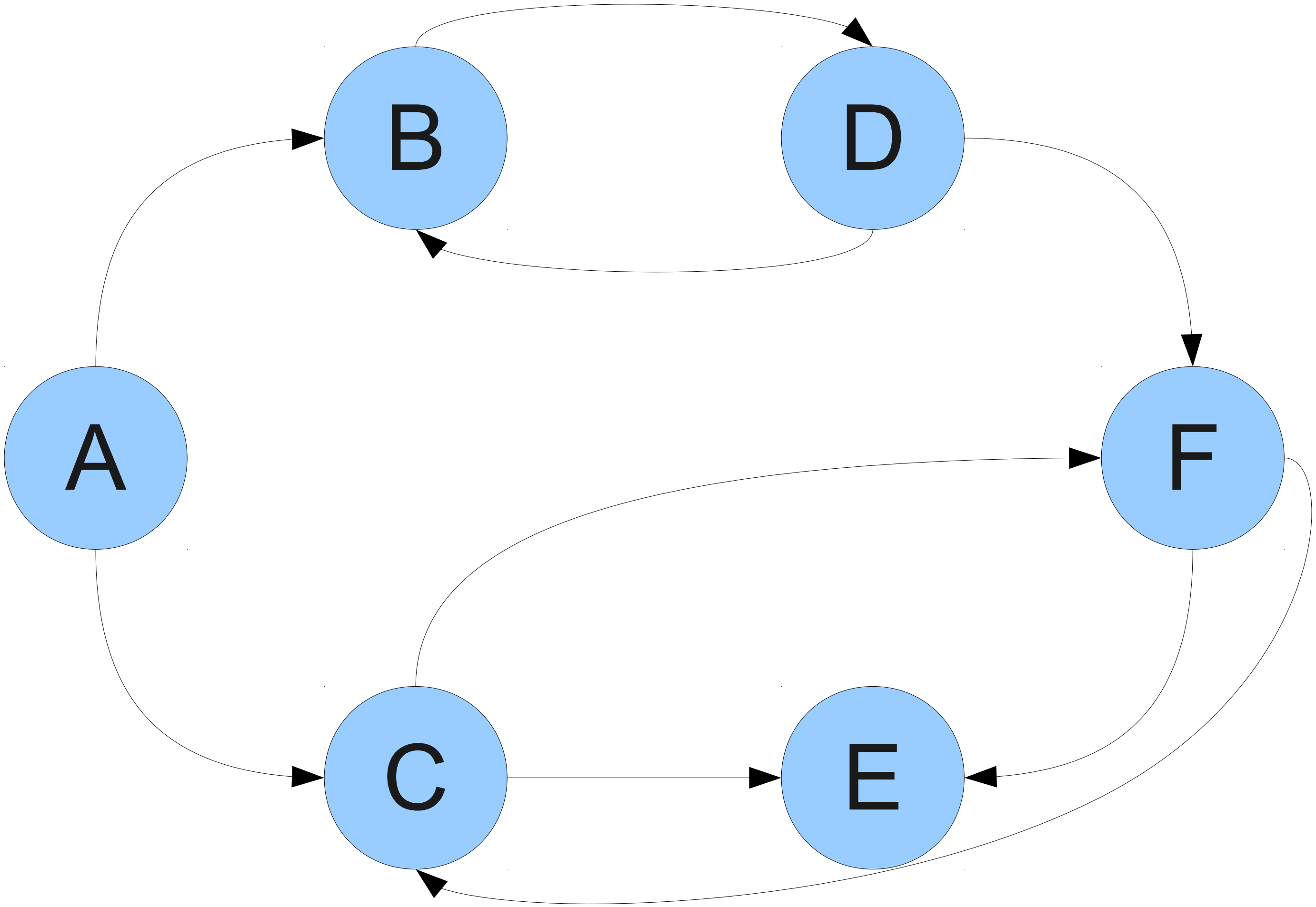
$((v_0, v_1), (v_1, v_2), \ldots, (v_n, v_0))$

that starts and ends at the same node.

A **simple path** is a path that does not contain a cycle.

A **simple cycle** is a cycle that does not contain a smaller cycle

Properties of Nodes



The **indegree** of a node is the number of edges entering that node.

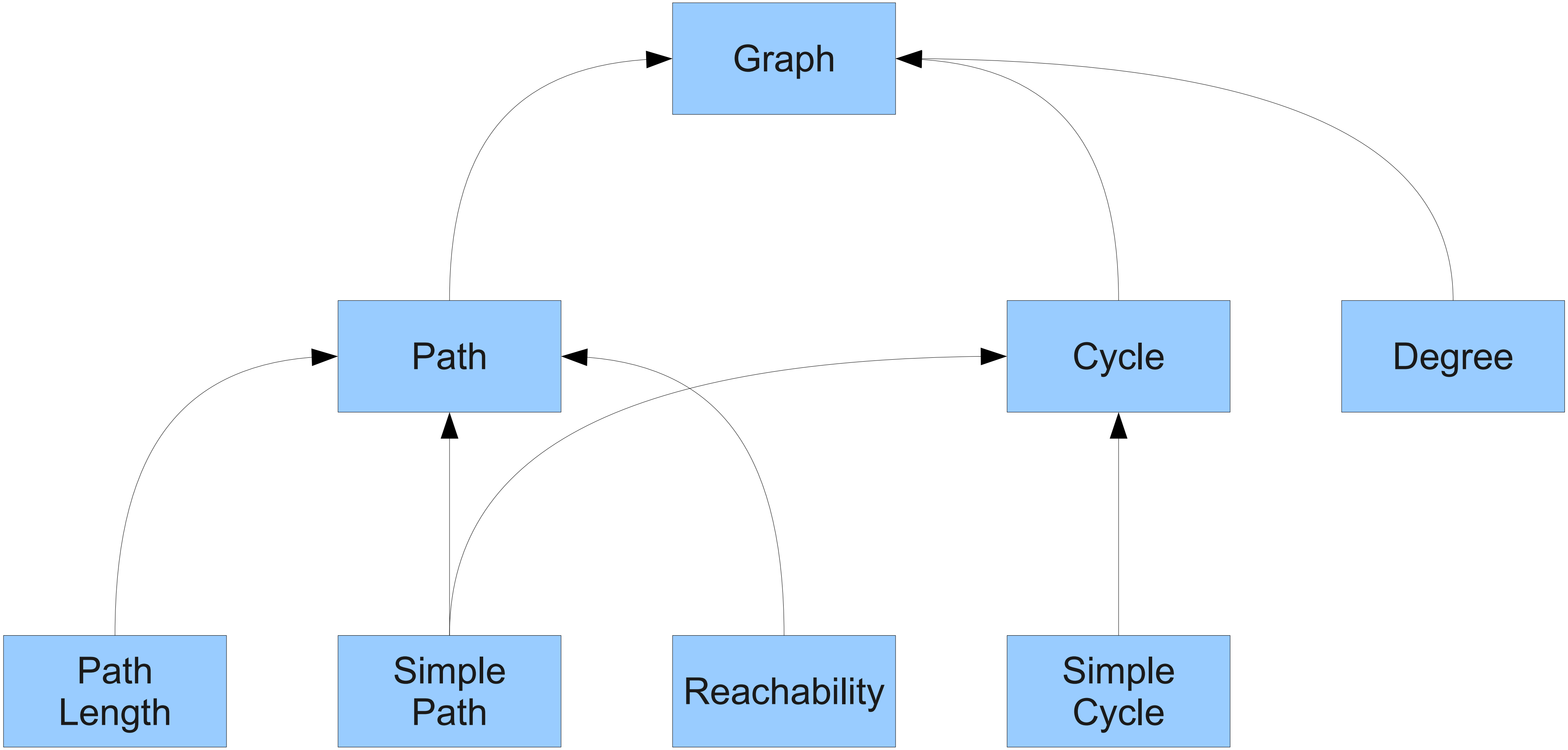
The **outdegree** of a node is the number of edges leaving that node.

In an undirected graph, these are the same and are called the **degree** of the node.

Summary of Terminology

- A **path** is a series of edges connecting two nodes.
  - The **length** of a path is the number of edges in the path.
  - A node v is **reachable** from u if there is a path from u to v.
- A **cycle** is a path from a node to itself.
- A **simple path** is a path without a cycle.
- A **simple cycle** is a cycle that does not contain a nested cycle.
- The **indegree** and **outdegree** of a node are the number of edges entering/leaving it.

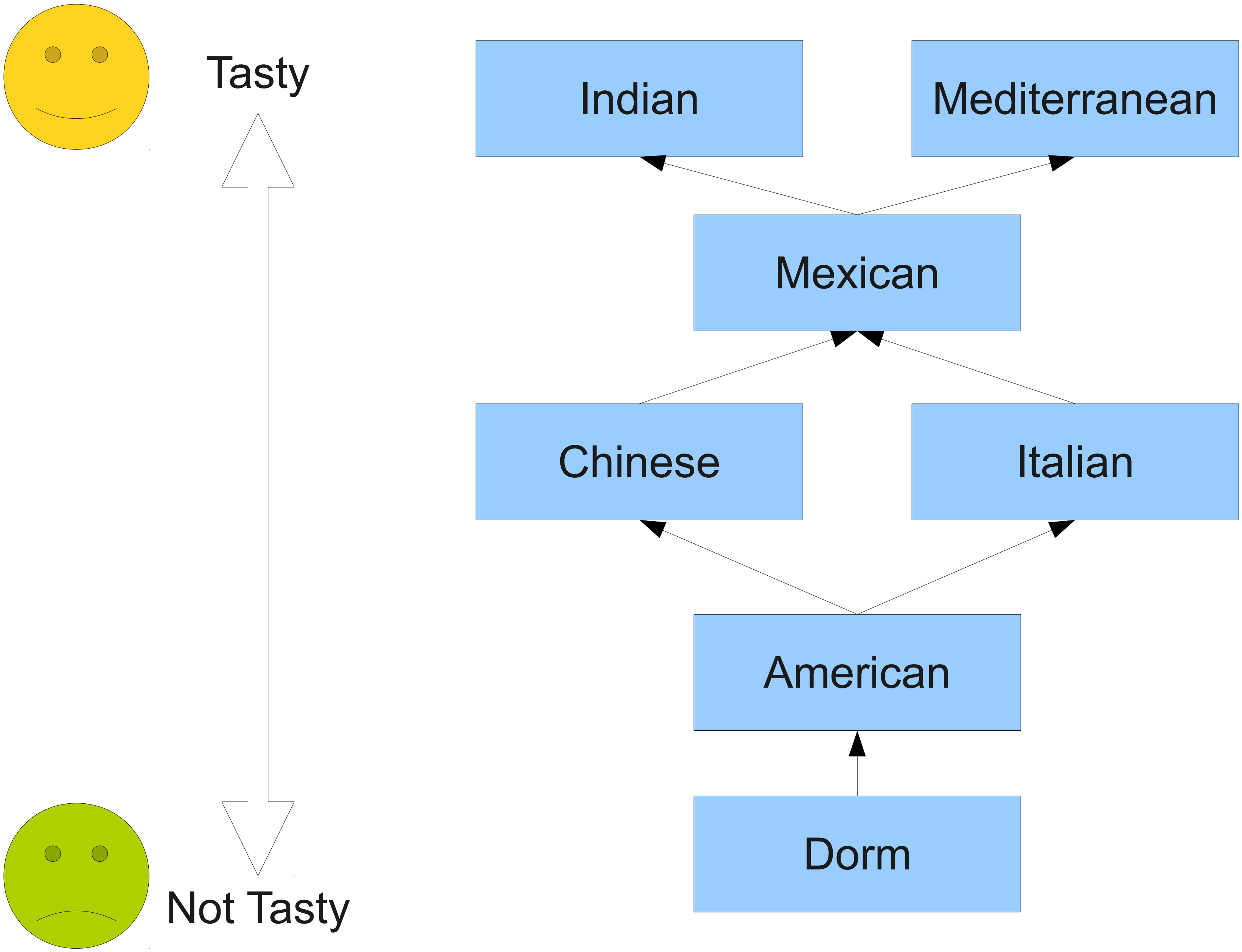
Representing Prerequisites



A **directed acyclic graph** (DAG) is a directed graph with no cycles.

Examples of DAGs

Examples of DAGs



Tasty

Not Tasty

Indian

Mediterranean

Mexican

Chinese

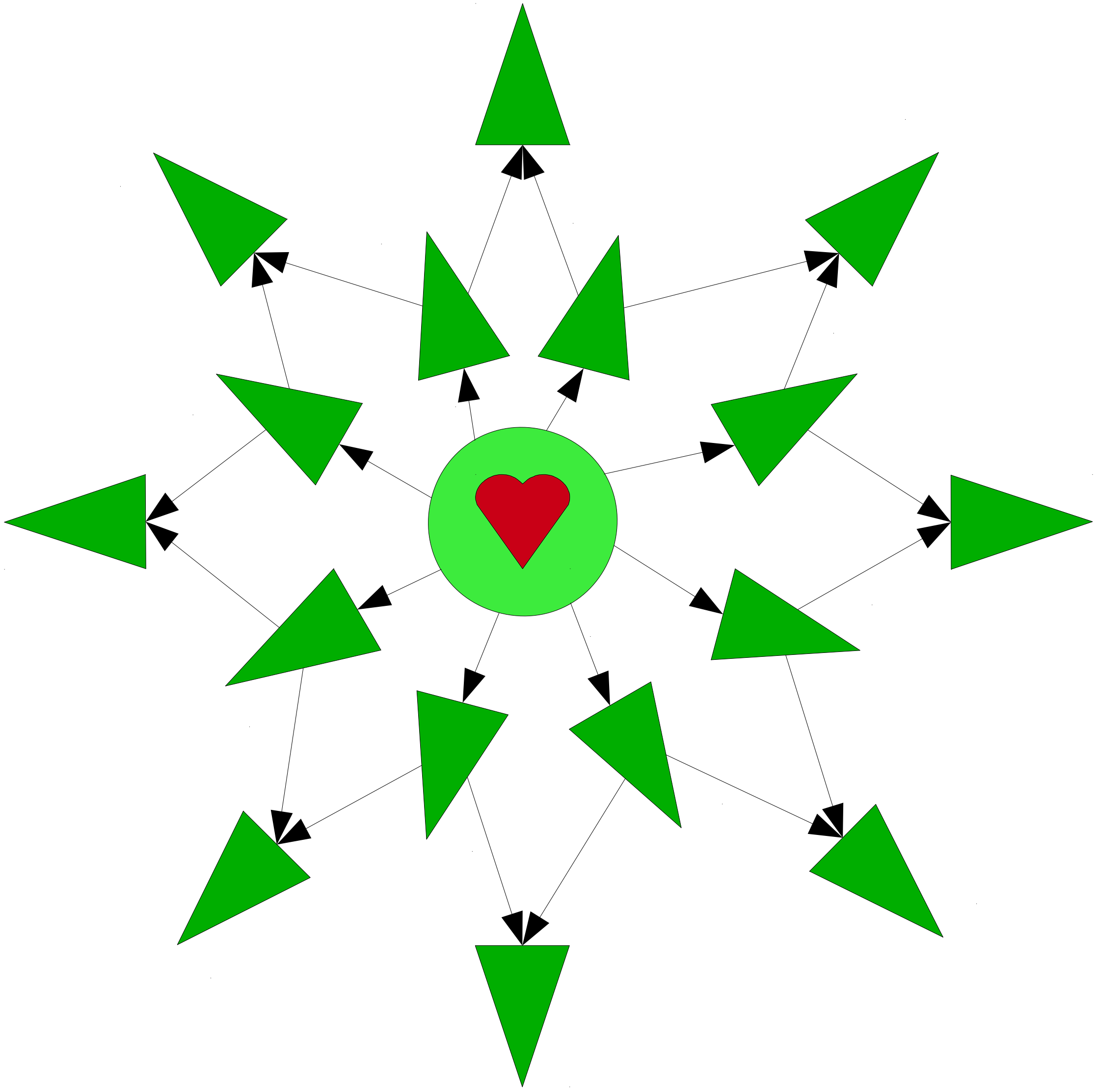
Italian

American

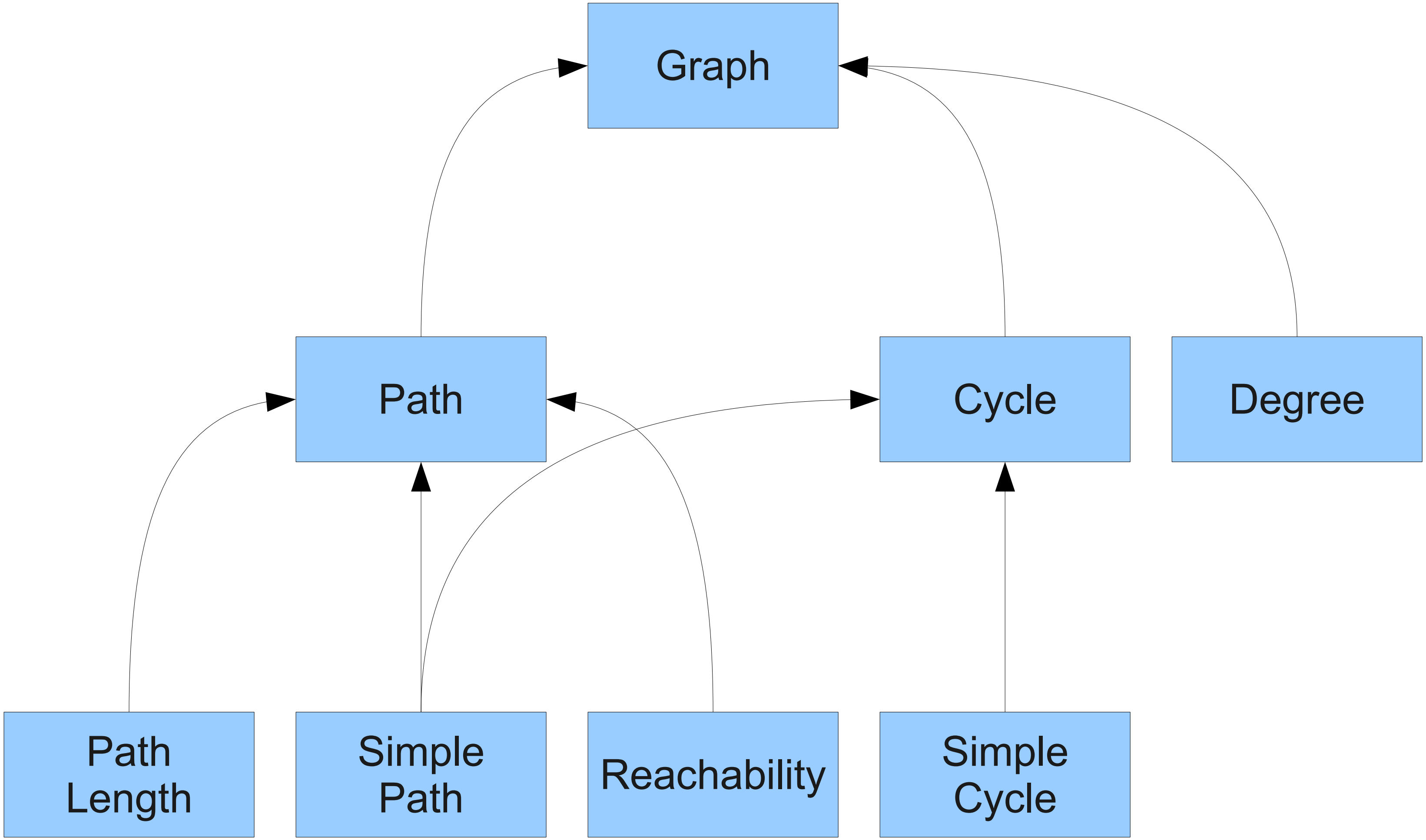
Dorm

Examples of DAGs

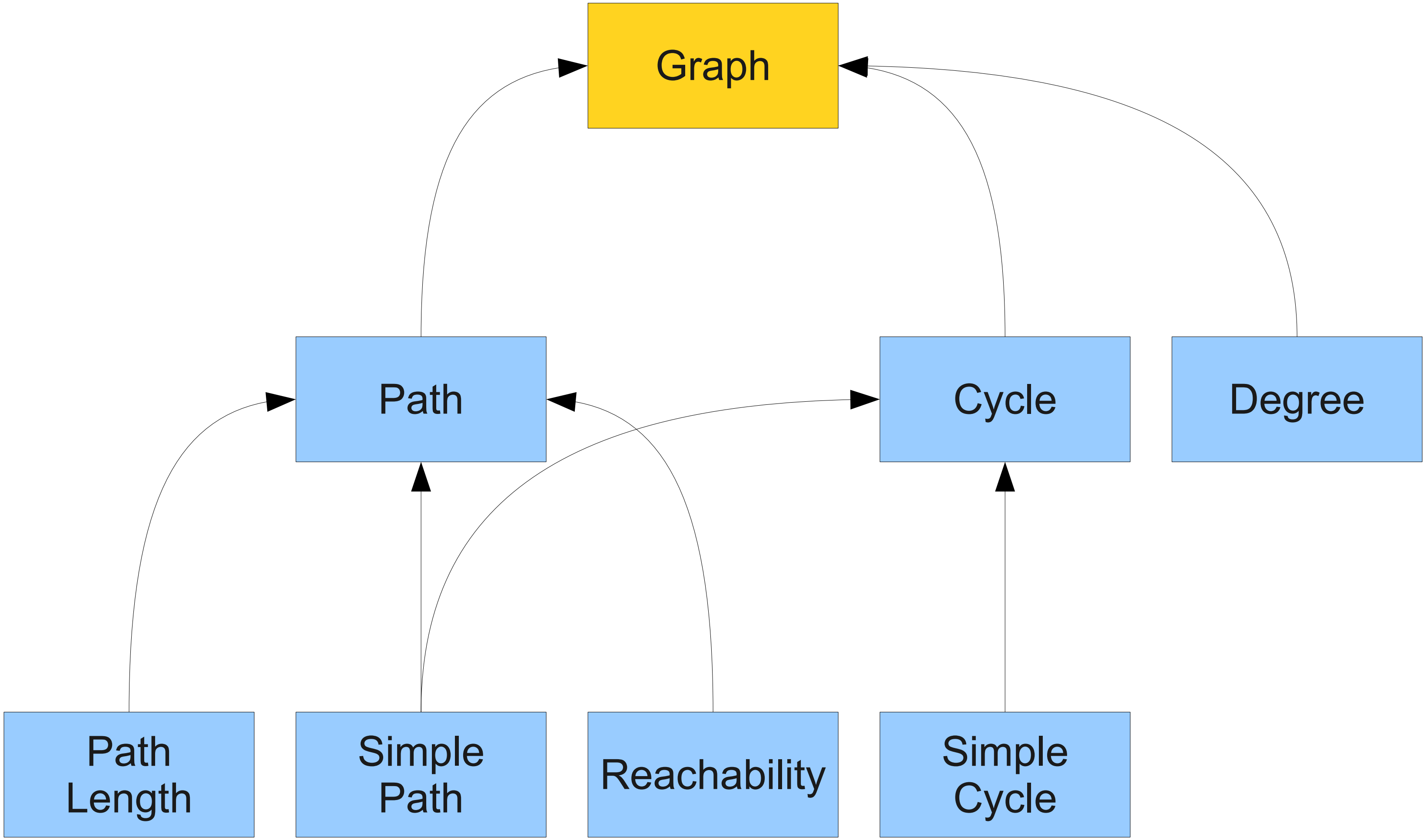
Examples of DAGs



Traversing a DAG

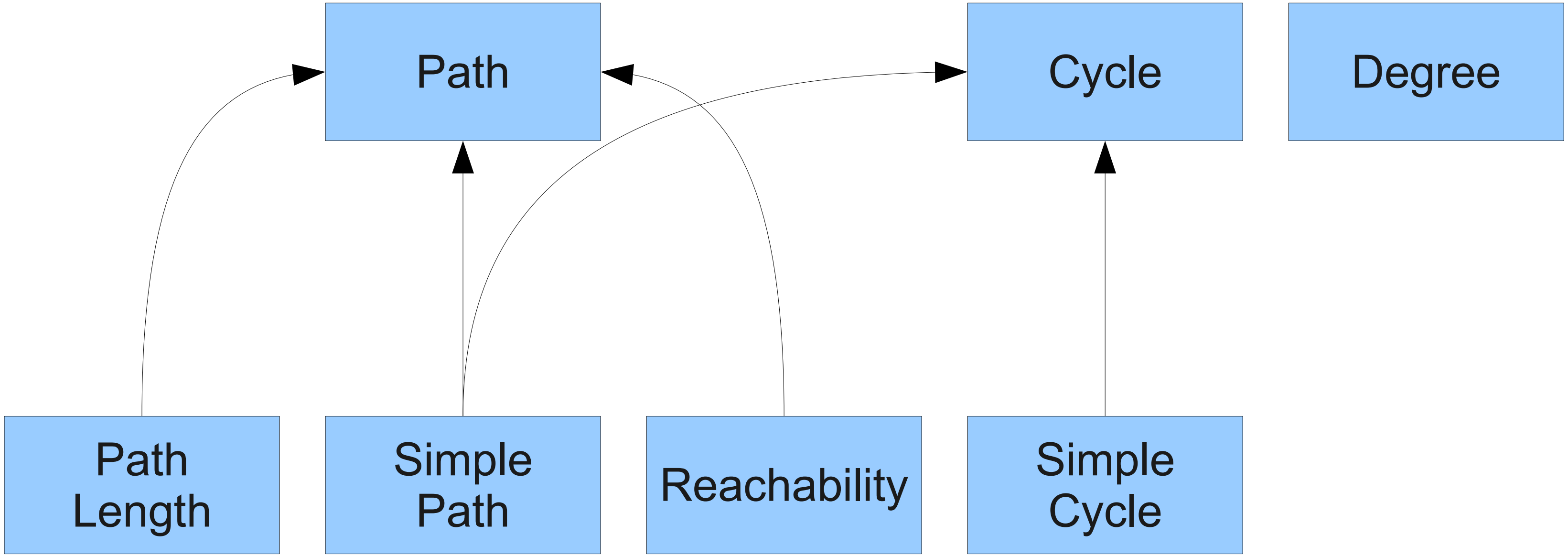


Traversing a DAG

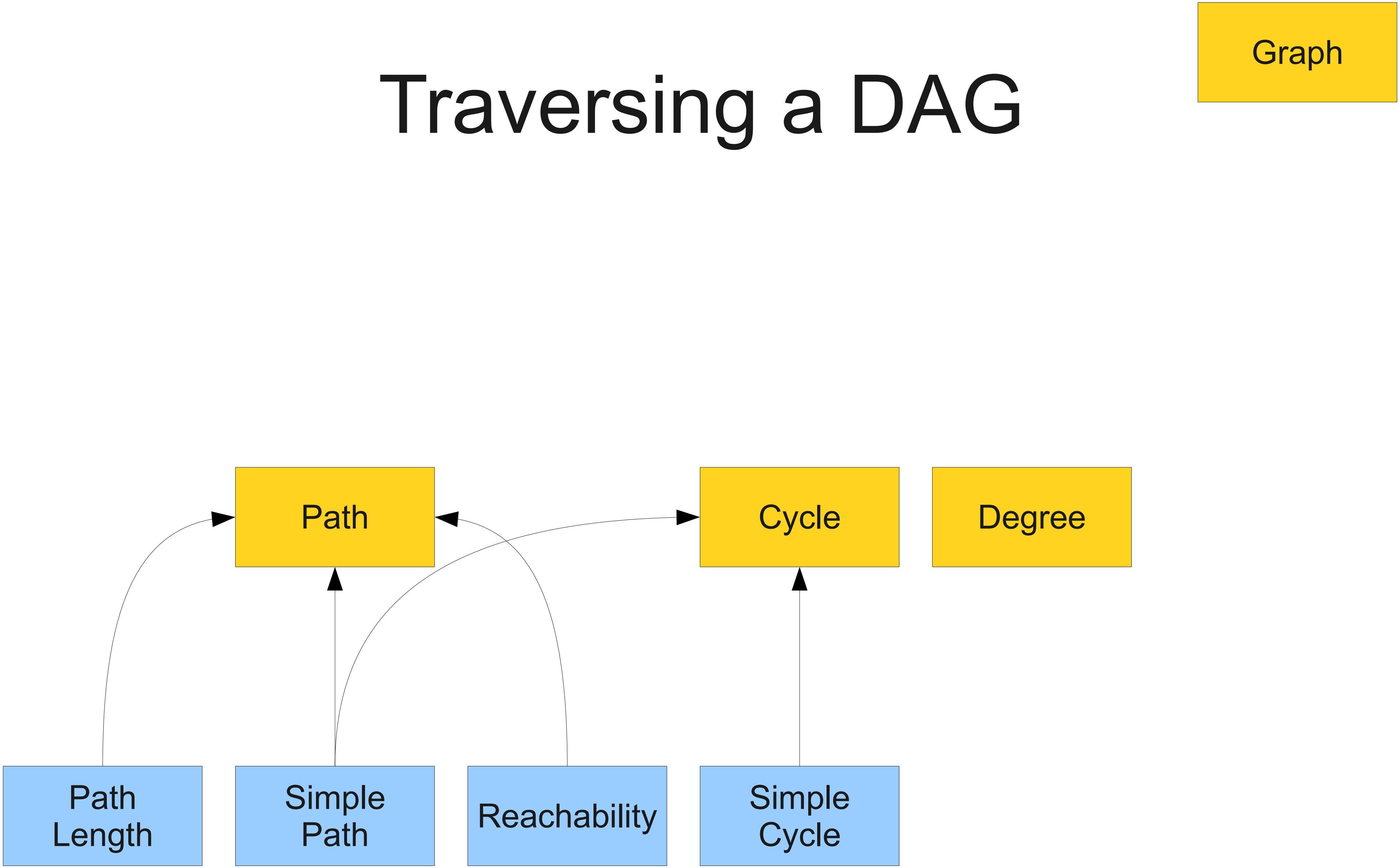


Graph

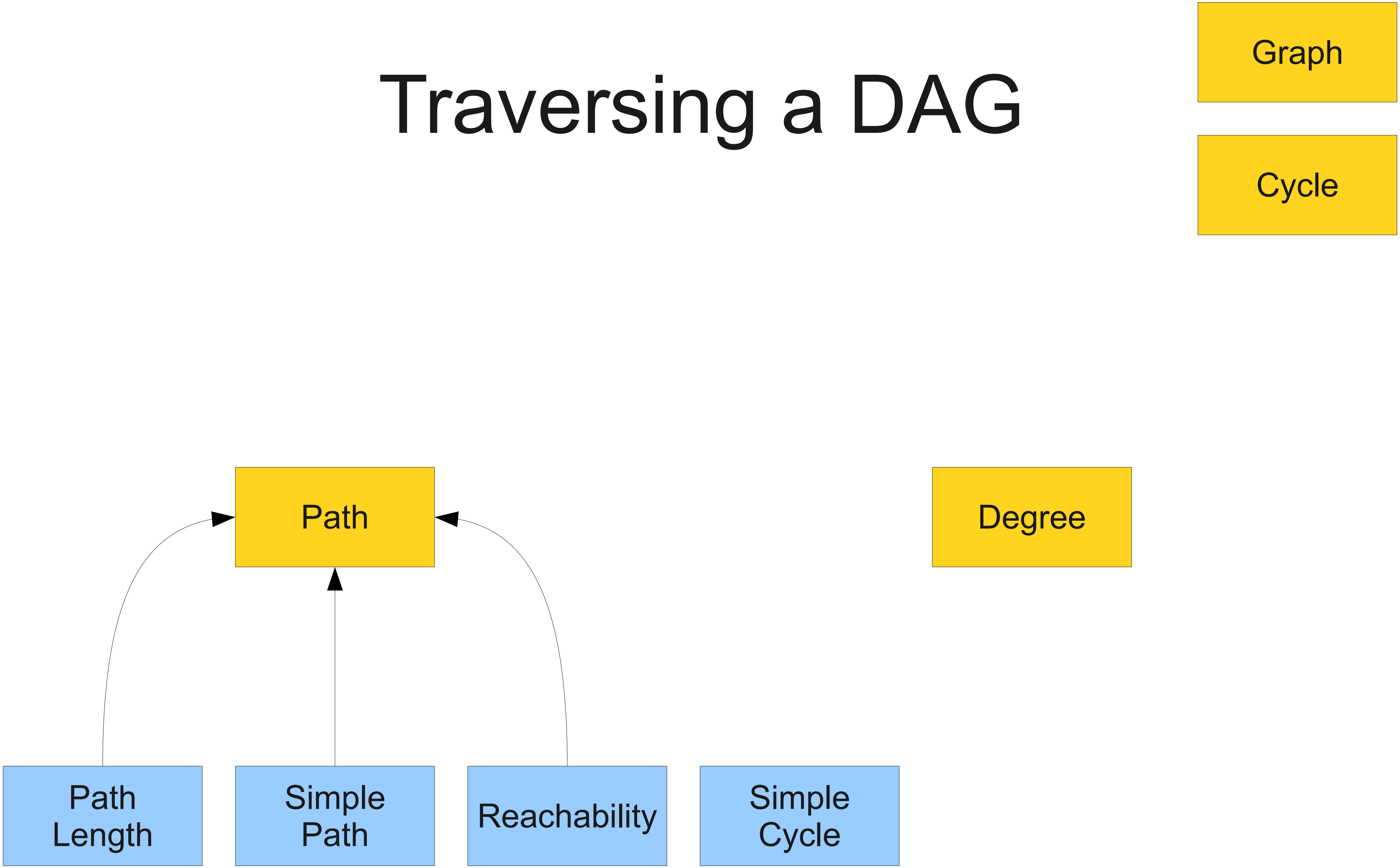
Traversing a DAG



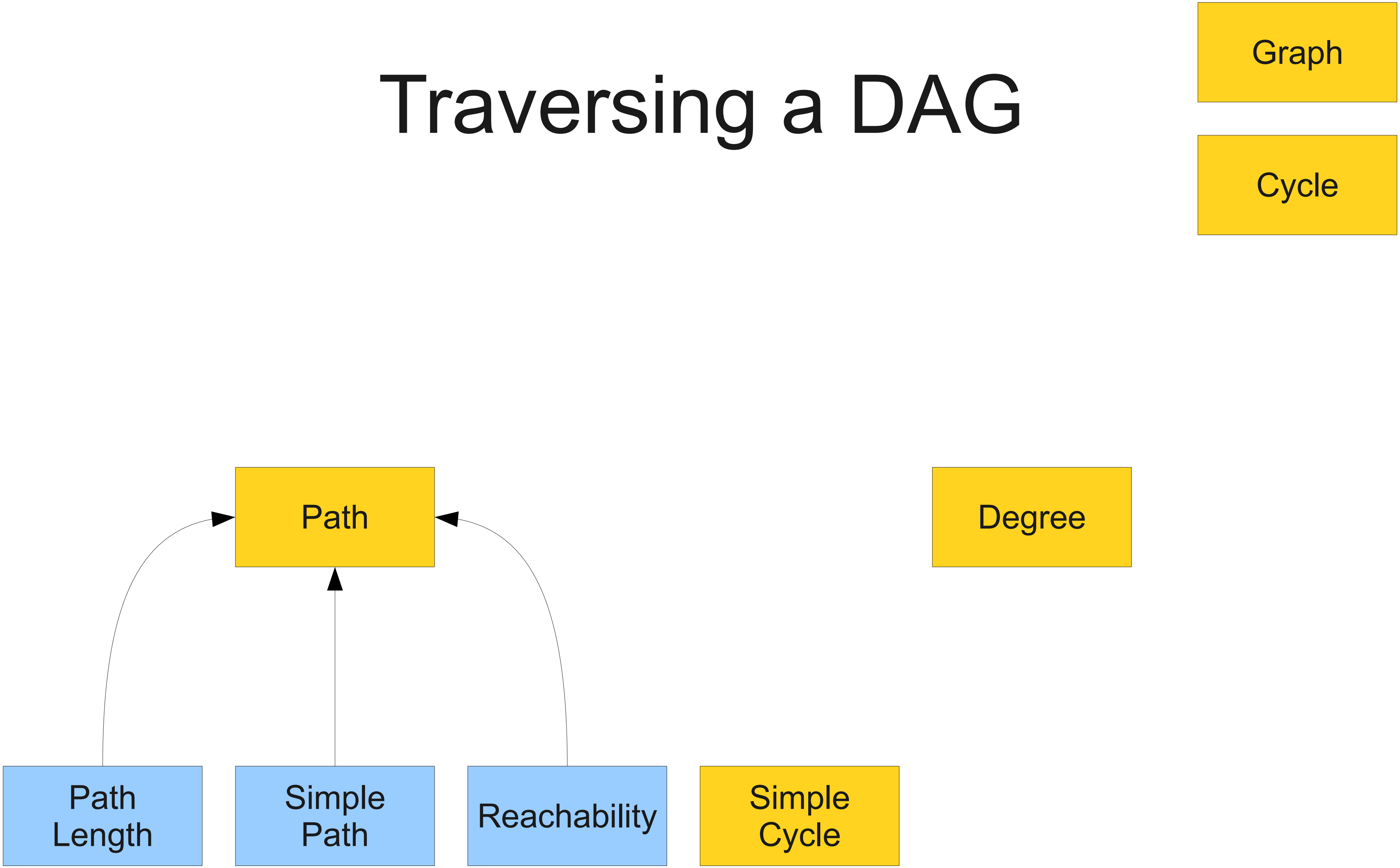
Traversing a DAG



Traversing a DAG



Traversing a DAG

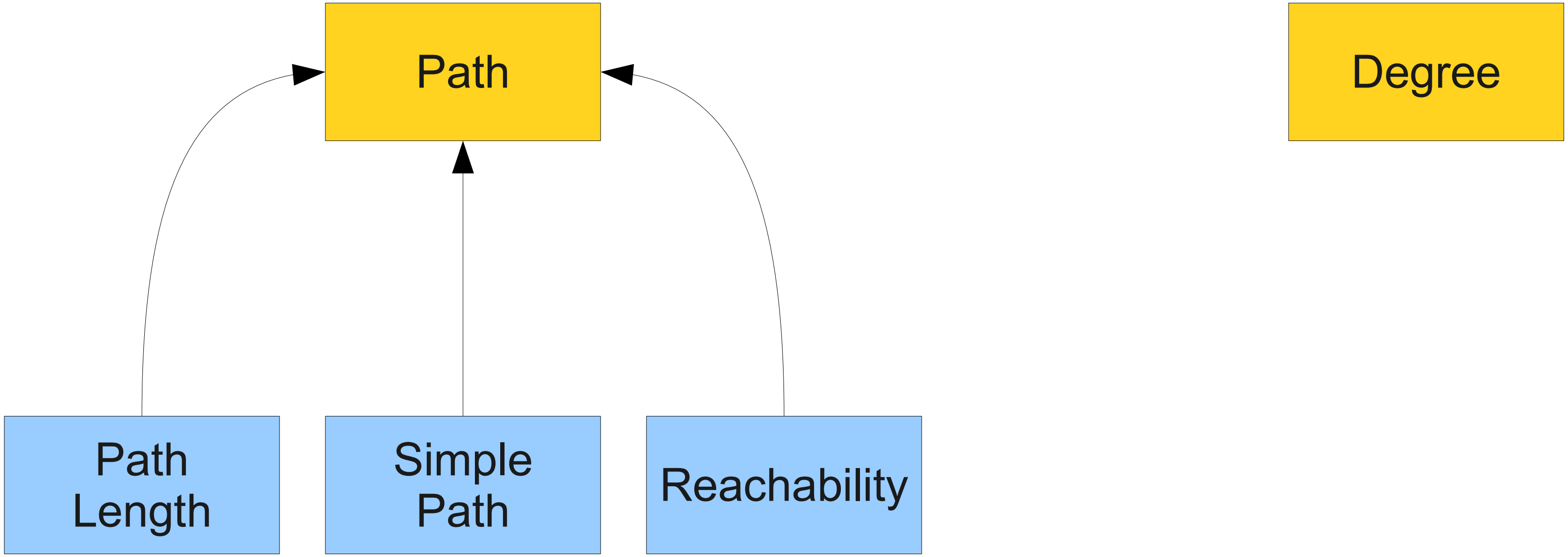


Graph

Cycle

Simple Cycle

Traversing a DAG



Path

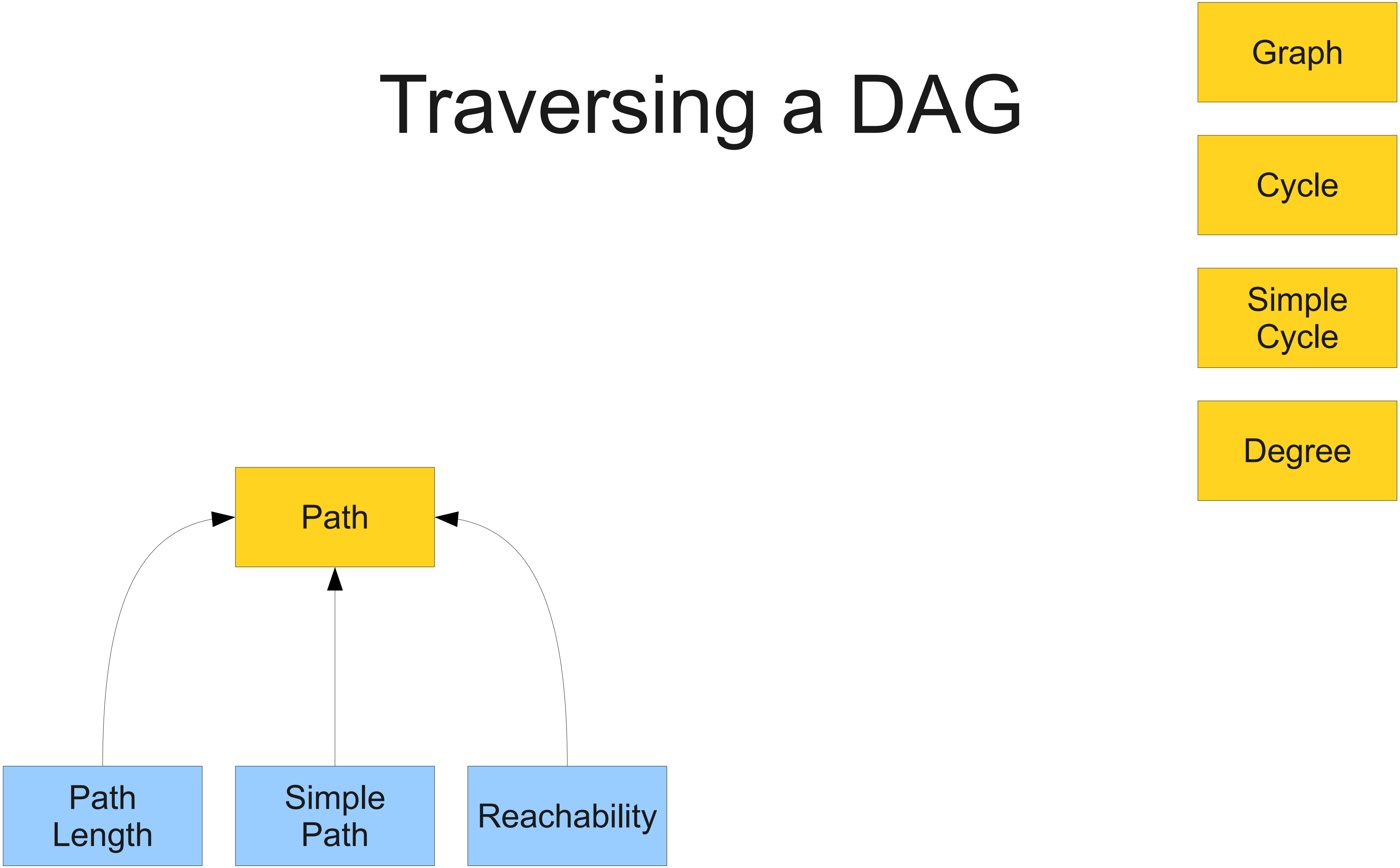
Degree

Path Length

Simple Path

Reachability

Traversing a DAG



Traversing a DAG

Graph

Cycle

Simple Cycle

Degree

Path

Path Length

Simple Path

Reachability

Traversing a DAG

Graph

Cycle

Simple Cycle

Degree

Path

Path Length

Simple Path

Reachability

Traversing a DAG

Graph

Cycle

Simple Cycle

Degree

Path

Path Length

Simple Path

Reachability

Traversing a DAG

Graph

Cycle

Simple Cycle

Degree

Path

Path Length

Simple Path

Reachability

Traversing a DAG

Graph

Cycle

Simple Cycle

Degree

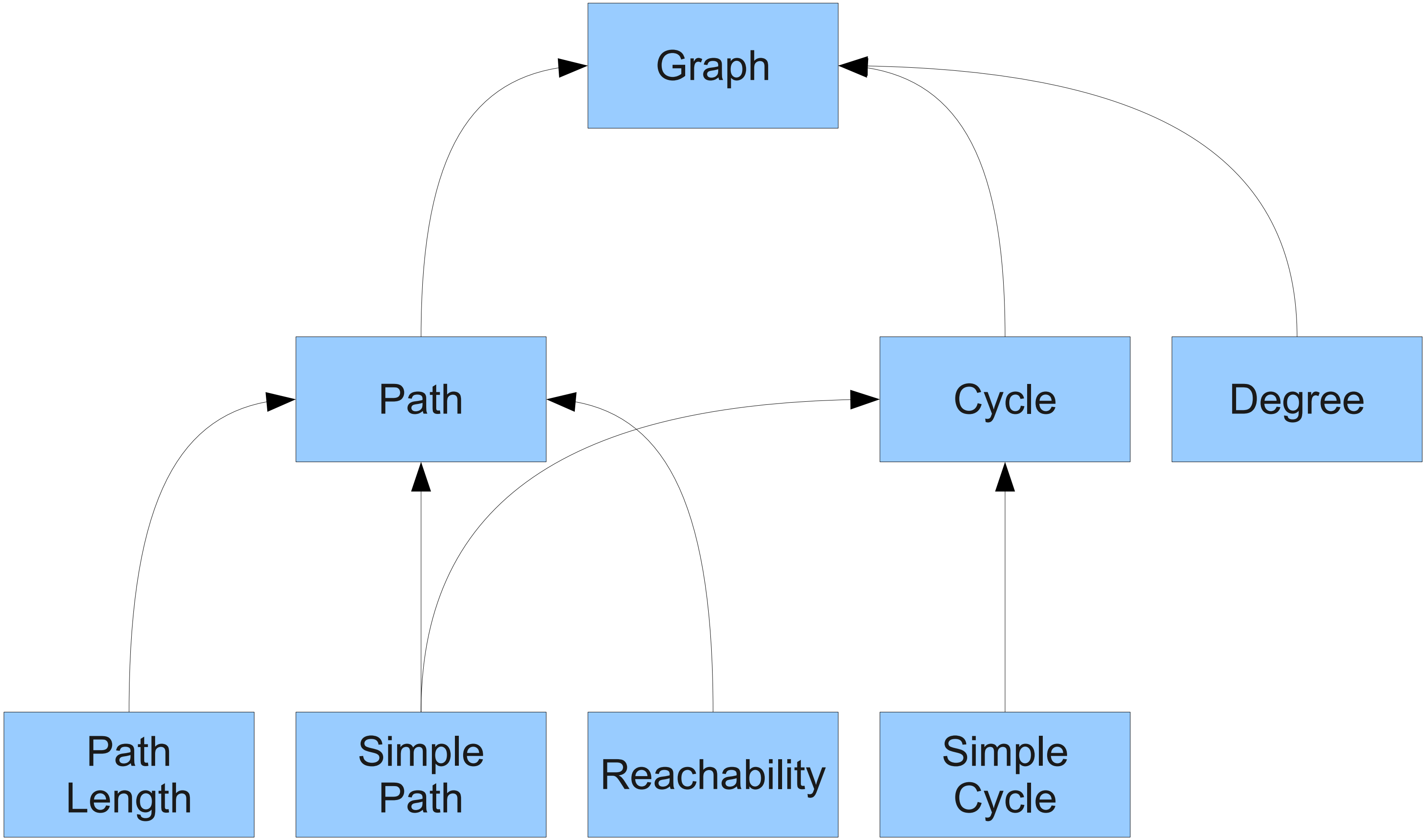
Path

Path Length

Simple Path

Reachability

Traversing a DAG



Graph

Cycle

Simple Cycle

Degree

Path

Path Length

Simple Path

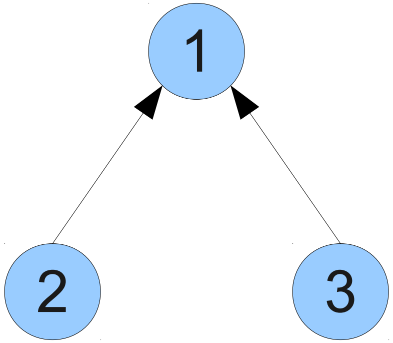
Reachability

Topological Sort

- Order the nodes of a DAG so no node is picked before its predecessors.
- Algorithm:
  - Find a node with no outgoing edges (outdegree 0)
  - Remove it from the graph.
  - Add it to the resulting ordering.
- There may be many valid orderings:

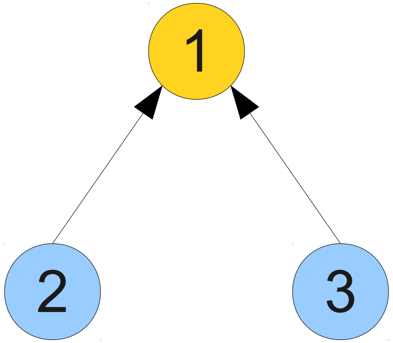
Topological Sort

- Order the nodes of a DAG so no node is picked before its predecessors.
- Algorithm:
    - Find a node with no outgoing edges (outdegree 0)
    - Remove it from the graph.
    - Add it to the resulting ordering.
- There may be many valid orderings:



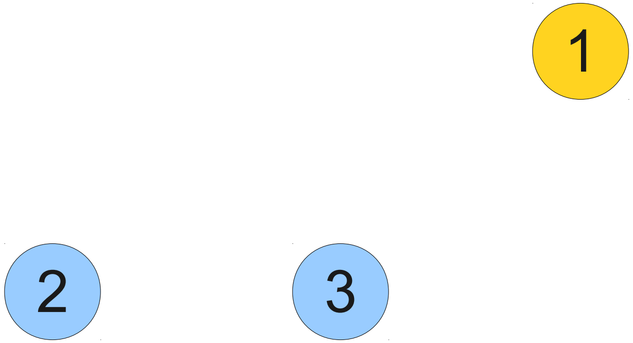
Topological Sort

- Order the nodes of a DAG so no node is picked before its predecessors.
- Algorithm:
  - Find a node with no outgoing edges (outdegree 0)
  - Remove it from the graph.
  - Add it to the resulting ordering.
- There may be many valid orderings:



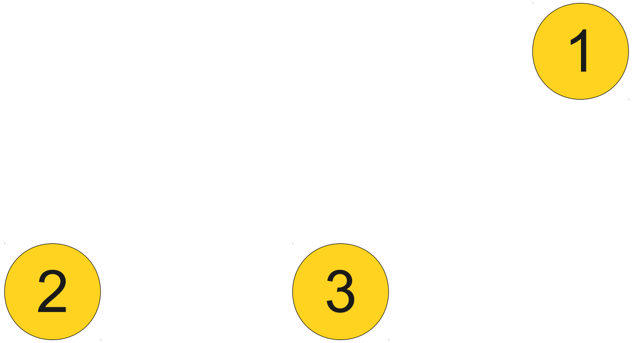
Topological Sort

- Order the nodes of a DAG so no node is picked before its predecessors.
- Algorithm:
  - Find a node with no outgoing edges (outdegree 0)
  - Remove it from the graph.
  - Add it to the resulting ordering.
- There may be many valid orderings:



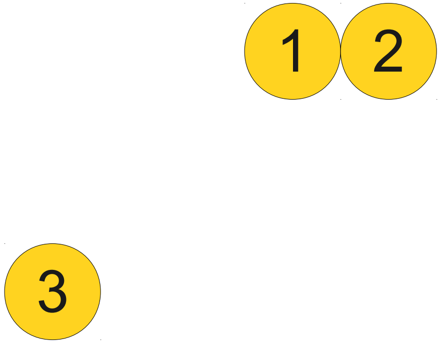
Topological Sort

- Order the nodes of a DAG so no node is picked before its predecessors.
- Algorithm:
  - Find a node with no outgoing edges (outdegree 0)
  - Remove it from the graph.
  - Add it to the resulting ordering.
- There may be many valid orderings:



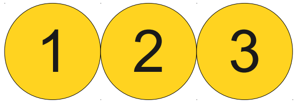
Topological Sort

- Order the nodes of a DAG so no node is picked before its predecessors.
- Algorithm:
  - Find a node with no outgoing edges (outdegree 0)
  - Remove it from the graph.
  - Add it to the resulting ordering.
- There may be many valid orderings:



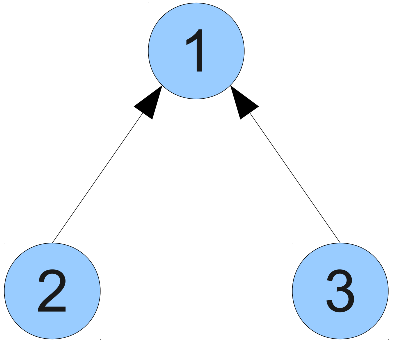
Topological Sort

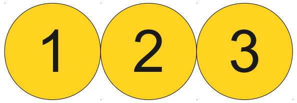
- Order the nodes of a DAG so no node is picked before its predecessors.
- Algorithm:
  - Find a node with no outgoing edges (outdegree 0)
  - Remove it from the graph.
  - Add it to the resulting ordering.
- There may be many valid orderings:



Topological Sort

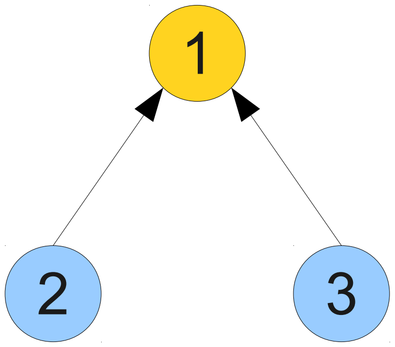
- Order the nodes of a DAG so no node is picked before its predecessors.
- Algorithm:
  - Find a node with no outgoing edges (outdegree 0)
  - Remove it from the graph.
  - Add it to the resulting ordering.
- There may be many valid orderings:

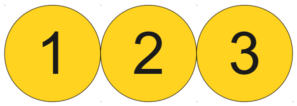




Topological Sort

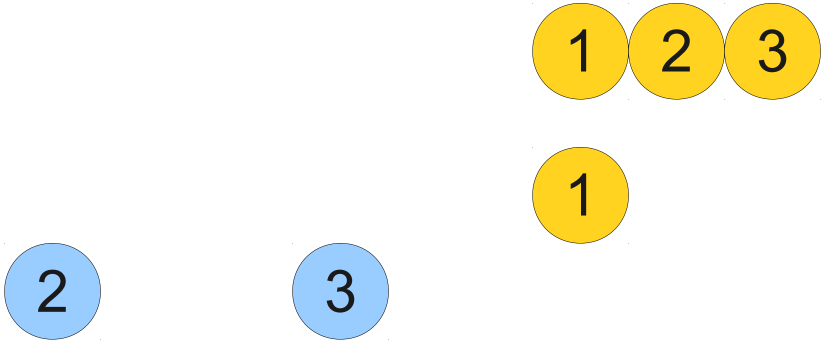
- Order the nodes of a DAG so no node is picked before its predecessors.
- Algorithm:
  - Find a node with no outgoing edges (outdegree 0)
  - Remove it from the graph.
  - Add it to the resulting ordering.
- There may be many valid orderings:





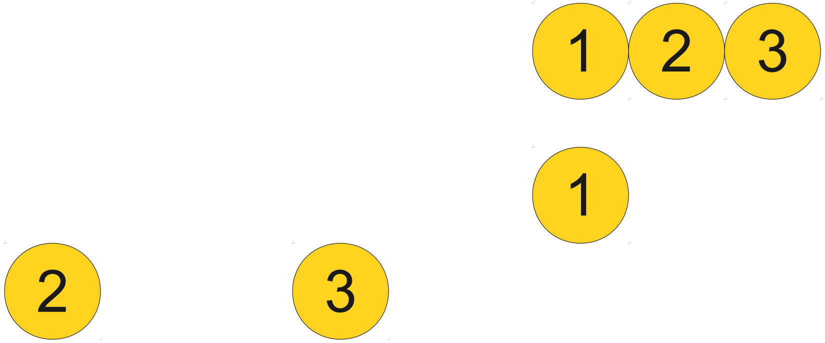
Topological Sort

- Order the nodes of a DAG so no node is picked before its predecessors.
- Algorithm:
  - Find a node with no outgoing edges (outdegree 0)
  - Remove it from the graph.
  - Add it to the resulting ordering.
- There may be many valid orderings:



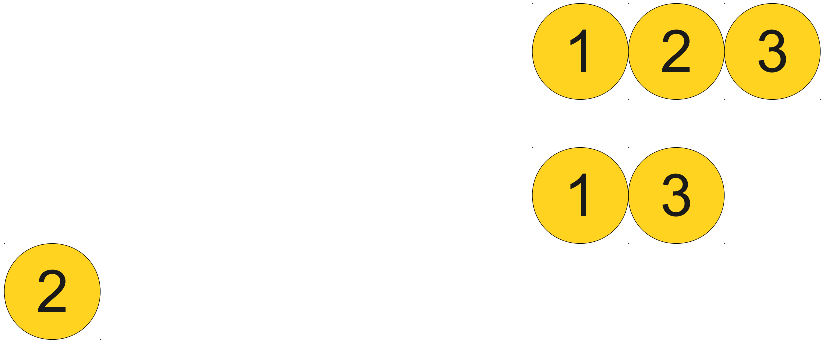
Topological Sort

- Order the nodes of a DAG so no node is picked before its predecessors.
- Algorithm:
  - Find a node with no outgoing edges (outdegree 0)
  - Remove it from the graph.
  - Add it to the resulting ordering.
- There may be many valid orderings:



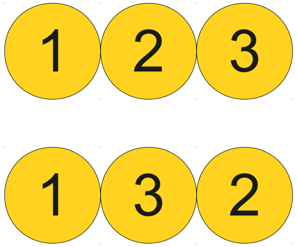
Topological Sort

- Order the nodes of a DAG so no node is picked before its predecessors.
- Algorithm:
  - Find a node with no outgoing edges (outdegree 0)
  - Remove it from the graph.
  - Add it to the resulting ordering.
- There may be many valid orderings:



Topological Sort

- Order the nodes of a DAG so no node is picked before its predecessors.
- Algorithm:
  - Find a node with no outgoing edges (outdegree 0)
  - Remove it from the graph.
  - Add it to the resulting ordering.
- There may be many valid orderings:

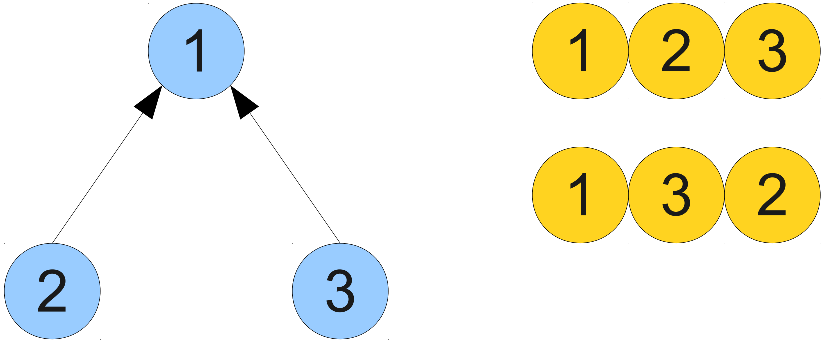


1 2 3

1 3 2

Topological Sort

- Order the nodes of a DAG so no node is picked before its predecessors.
- Algorithm:
  - Find a node with no outgoing edges (outdegree 0)
  - Remove it from the graph.
  - Add it to the resulting ordering.
- There may be many valid orderings:



Relations

Relations

- A **binary relation** is a property that describes whether two objects are related in some way.
- Examples:
  - Less-than: $x < y$
  - Divisibility: *x* divides *y* evenly
  - Friendship: *x* is a friend of *y*
  - Tastiness: *x* is tastier than *y*
- If we have a binary relation *R*, we write *aRb* if *a* is **related** to *b*.
  - $a = b$
  - $a < b$
  - *a* "is tastier than" *b*

Relations as Sets

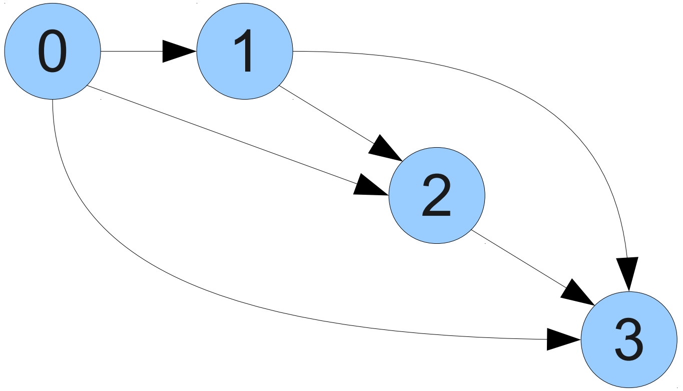
- Formally, a relation is a set of ordered pairs representing the pairs for which the relation is true.
  - Equality: { (0, 0), (1, 1), (2, 2), … }
  - Less-than: { (0, 1), (0, 2), …, (1, 2), (1, 3), … }
- Formally, we have that

$$\boldsymbol{aRb \equiv (a, b) \in R}$$

- The binary relations we'll discuss today will be binary relations over a set *A*.
  - Each relation is a subset of A^2.

Binary Relations and Graphs

- Each (directed) graph defines a binary relation:
  - *aRb* iff (*a*, *b*) is an edge.
- Each binary relation defines a graph:
  - (*a*, *b*) is an edge iff *aRb*.
- Example: Less-than



An Important Question

- Why study binary relations and graphs separately?
- **Simplicity**:
    - Certain operations feel more “natural” on binary relations than on graphs and vice-versa.
    - Converting a relation to a graph might result in an overly complex graph.
- **Terminology**:
    - Vocabulary for graphs often different from that for relations.

Equivalence Relations

“x and y have the same color”

“x and y have the same shape”

“x = y”

“x and y have the same area”

“x and y are programs that produce the same output”

Informally

An **equivalence relation** is a relation that indicates when objects have some trait in common.

Do not use this definition in proofs!
It's just an intuition!

Properties of Equivalence Relations

xRy ≡ x and y have the same shape.

Properties of Equivalence Relations

xRy ≡ x and y have the same shape.

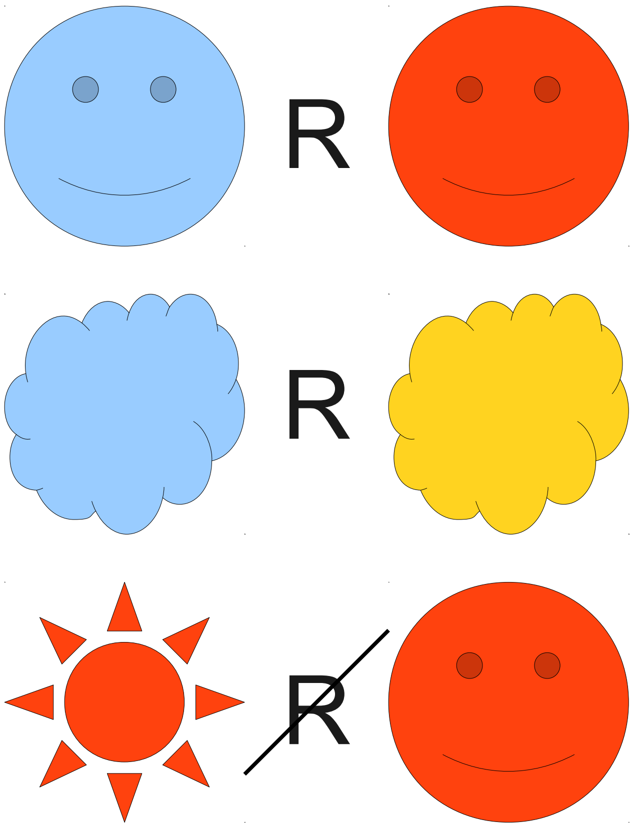
Recall: This symbol means
"is defined as"

Properties of Equivalence Relations

xRy ≡ x and y have the same shape.

Properties of Equivalence Relations

xRy ≡ x and y have the same shape.

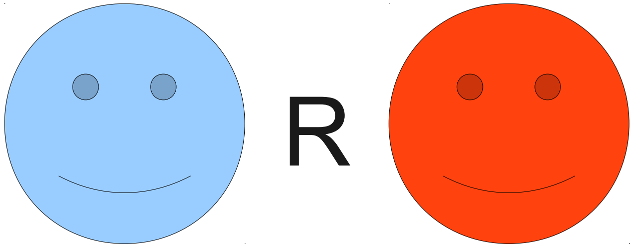


Properties of Equivalence Relations

xRy ≡ x and y have the same shape.

Properties of Equivalence Relations

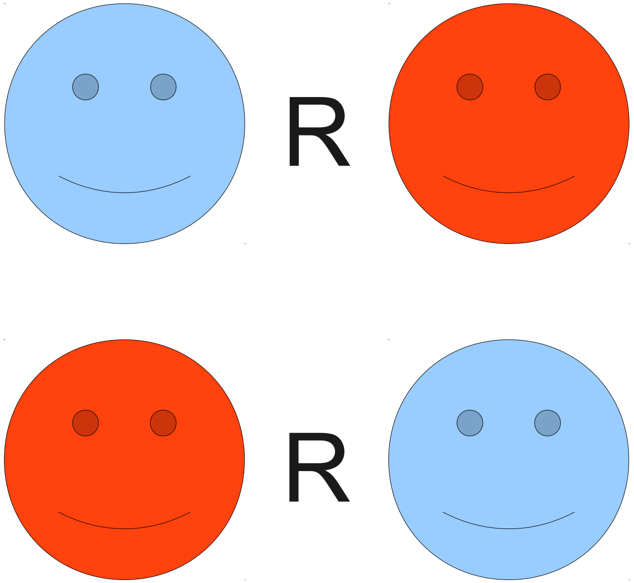
xRy ≡ x and y have the same shape.



R

Properties of Equivalence Relations

xRy ≡ x and y have the same shape.

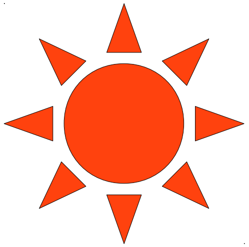
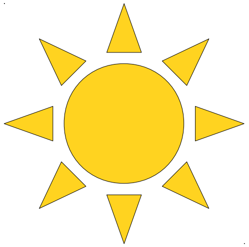


R

R

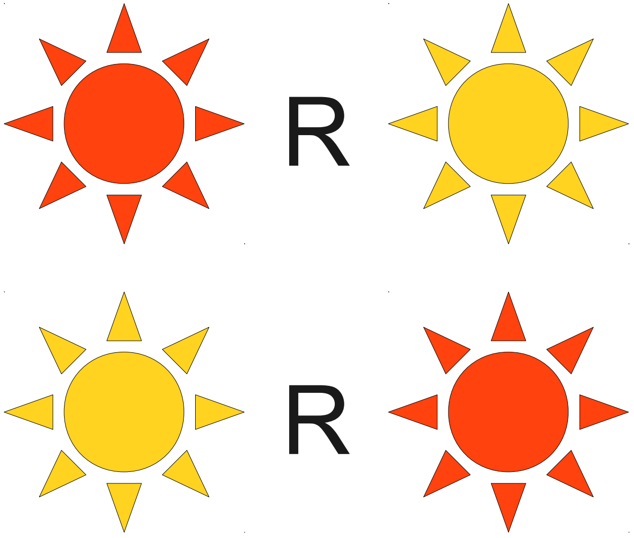
Properties of Equivalence Relations

xRy ≡ x and y have the same shape.

 R 

Properties of Equivalence Relations

xRy ≡ x and y have the same shape.



R

R

Properties of Equivalence Relations

xRy ≡ x and y have the same shape.

xRy

Properties of Equivalence Relations

xRy ≡ x and y have the same shape.

xRy

yRx

Symmetry

A binary relation *R* over a set *A*
is called **symmetric** iff

for any $x \in A$ and $y \in A$, if $x\text{R}y$, then $y\text{R}x$.

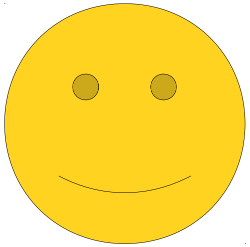
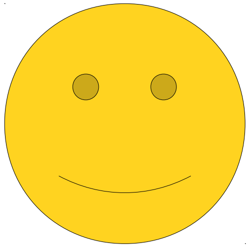
This definition (and others like it) can be used in formal proofs.

Properties of Equivalence Relations

xRy ≡ x and y have the same shape.

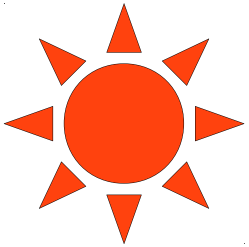
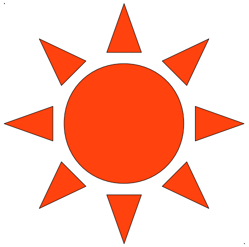
Properties of Equivalence Relations

xRy ≡ x and y have the same shape.

 R 

Properties of Equivalence Relations

xRy ≡ x and y have the same shape.

 R 

Properties of Equivalence Relations

$xRy \equiv x$ and y have the same shape.

$$xRx$$

Reflexivity

A binary relation *R* over a set *A* is called **reflexive** iff

For any $x \in A$, $x\mathrm{R}x$.

Some Reflexive Relations

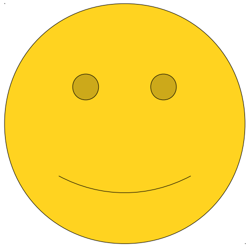
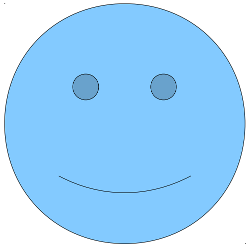
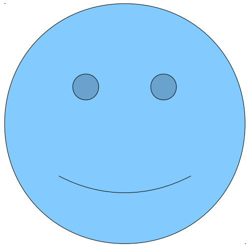
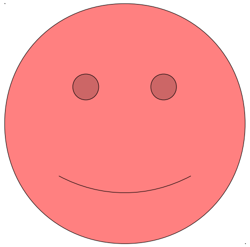
- Equality:
  - For any x, x = x.
- Not greater than:
  - For any integer x, x ≤ x.
- Subset:
  - For any set S, S ⊆ S.

Properties of Equivalence Relations

xRy ≡ x and y have the same shape.

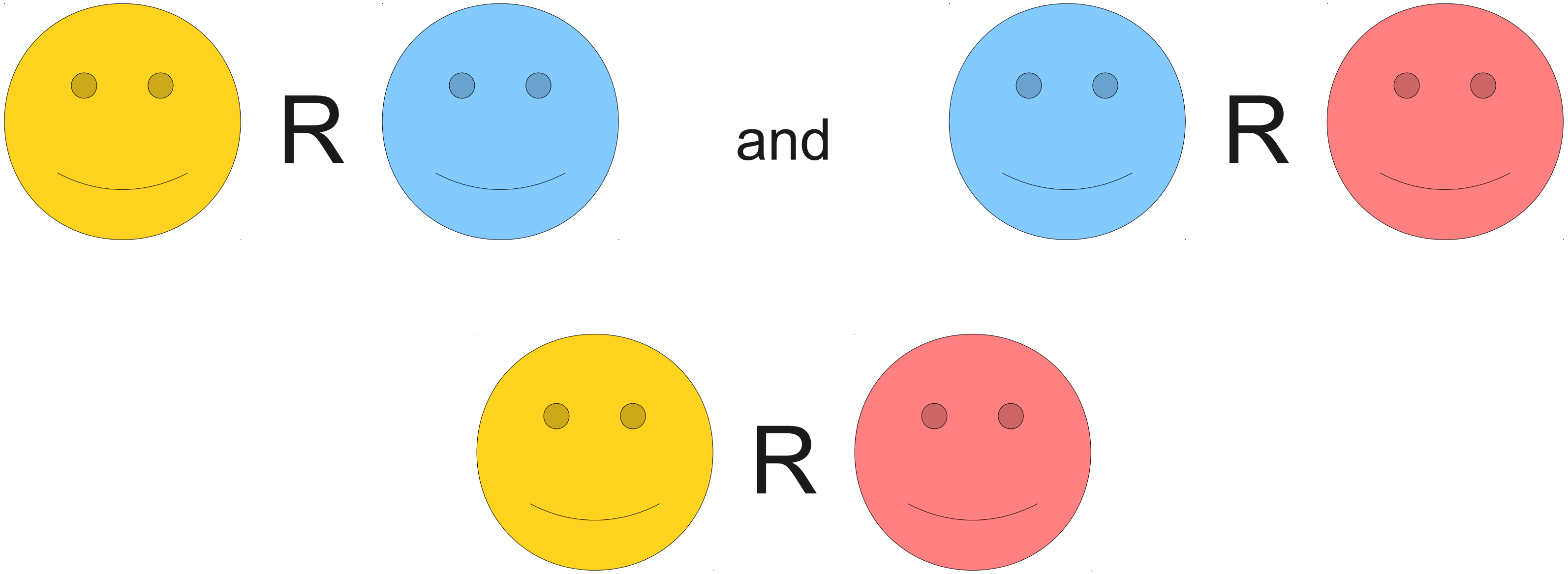
Properties of Equivalence Relations

xRy ≡ x and y have the same shape.

 R  and  R 

Properties of Equivalence Relations

xRy ≡ x and y have the same shape.

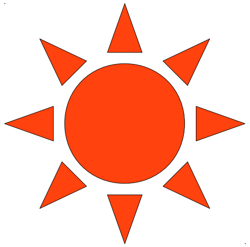
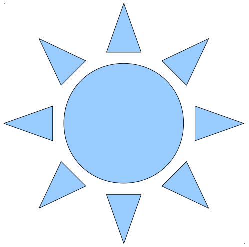
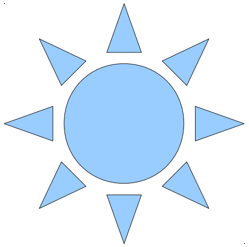
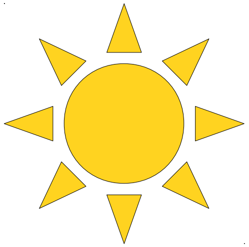


R and R

R

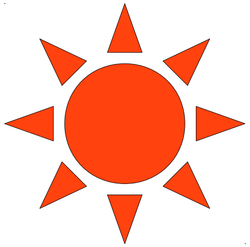
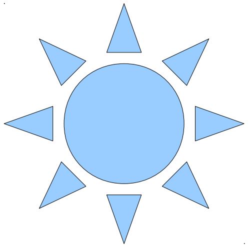
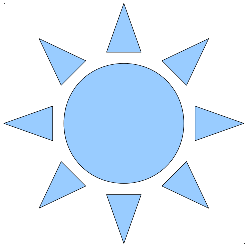
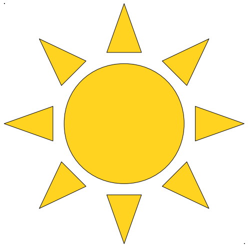
Properties of Equivalence Relations

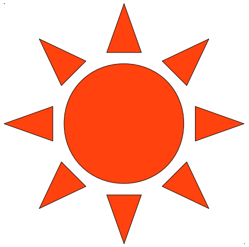
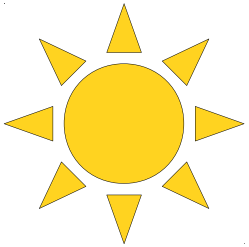
xRy ≡ x and y have the same shape.

 R  and  R 

Properties of Equivalence Relations

xRy ≡ x and y have the same shape.

 R  and  R 

 R 

Properties of Equivalence Relations

xRy ≡ x and y have the same shape.

xRy and yRz

Properties of Equivalence Relations

xRy ≡ x and y have the same shape.

xRy and yRz

xRz

Transitivity

A binary relation R over a set A
is called **transitive** iff

For any *x*, *y*, and *z*, if *x*R*y* and *y*R*z*, then *x*R*z*.

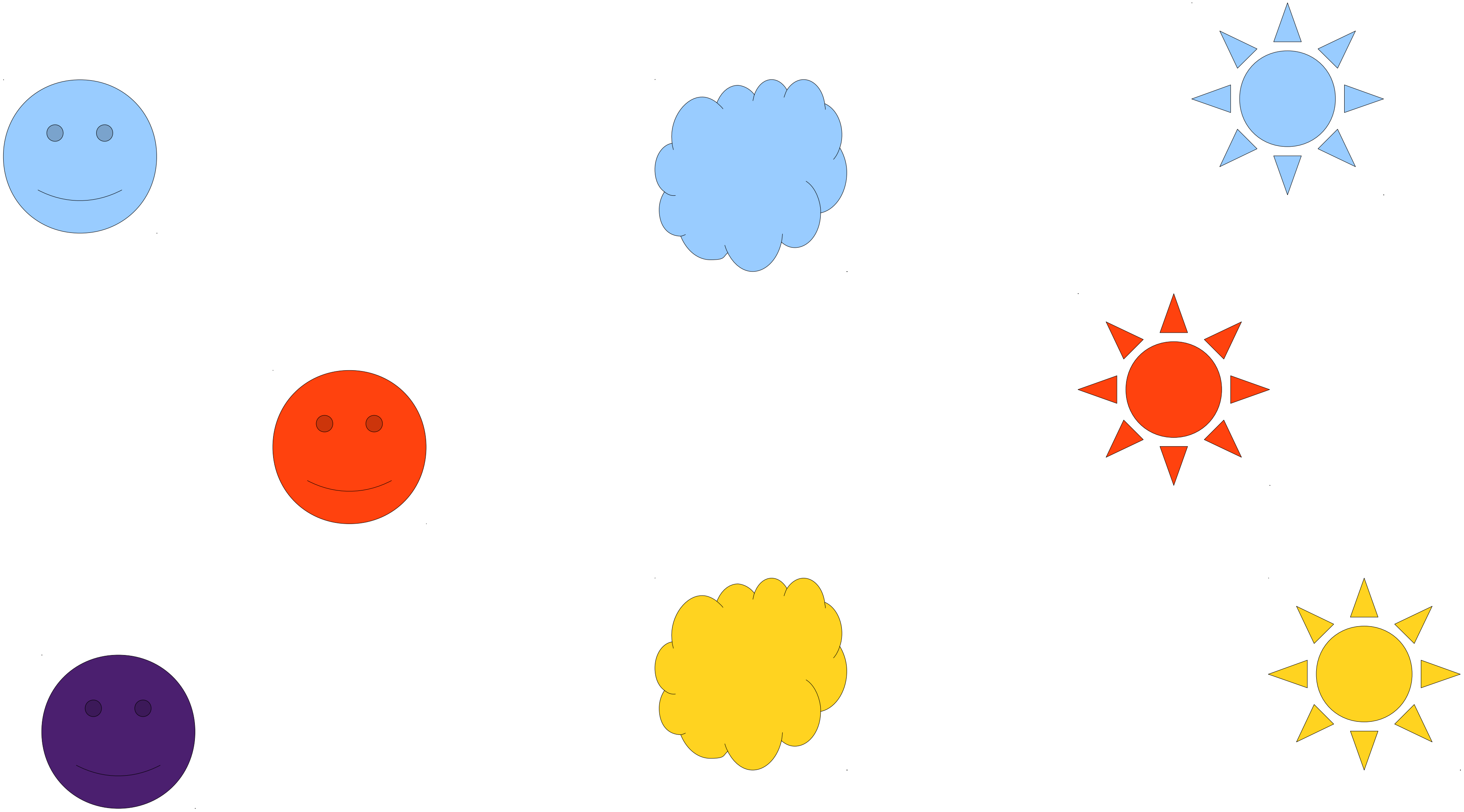
Some Transitive Relations

- Equality:
  - $x = y$ and $y = z$ implies $x = z$.
- Less-than:
  - $x < y$ and $y < z$ implies $x < z$.
- Subset:
  - $S \subseteq T$ and $T \subseteq U$ implies $S \subseteq U$.

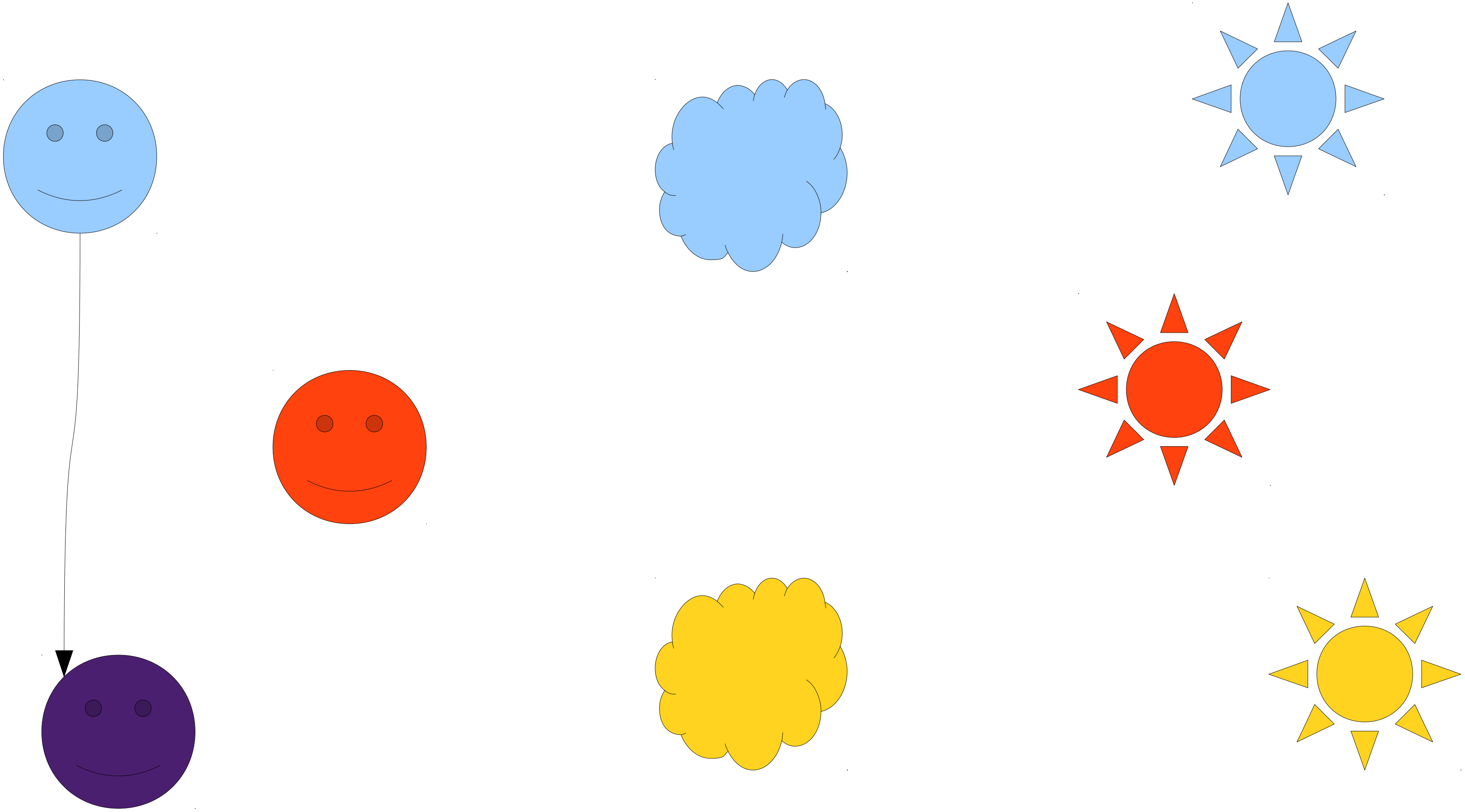
Equivalence Relations

A binary relation *R* over a set *A* is called an **equivalence relation** if it is

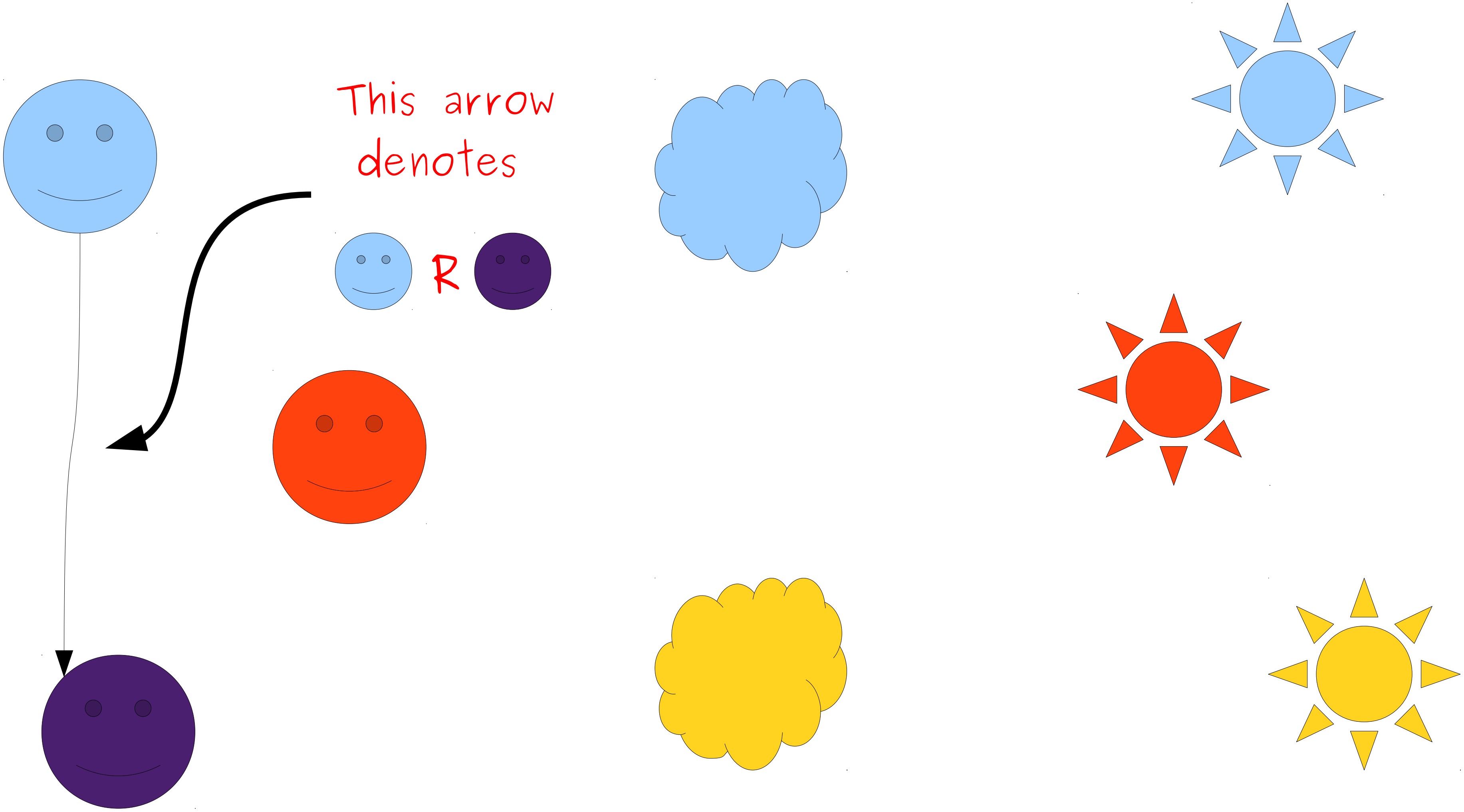
- **reflexive**,
- **symmetric**, and
- **transitive**.



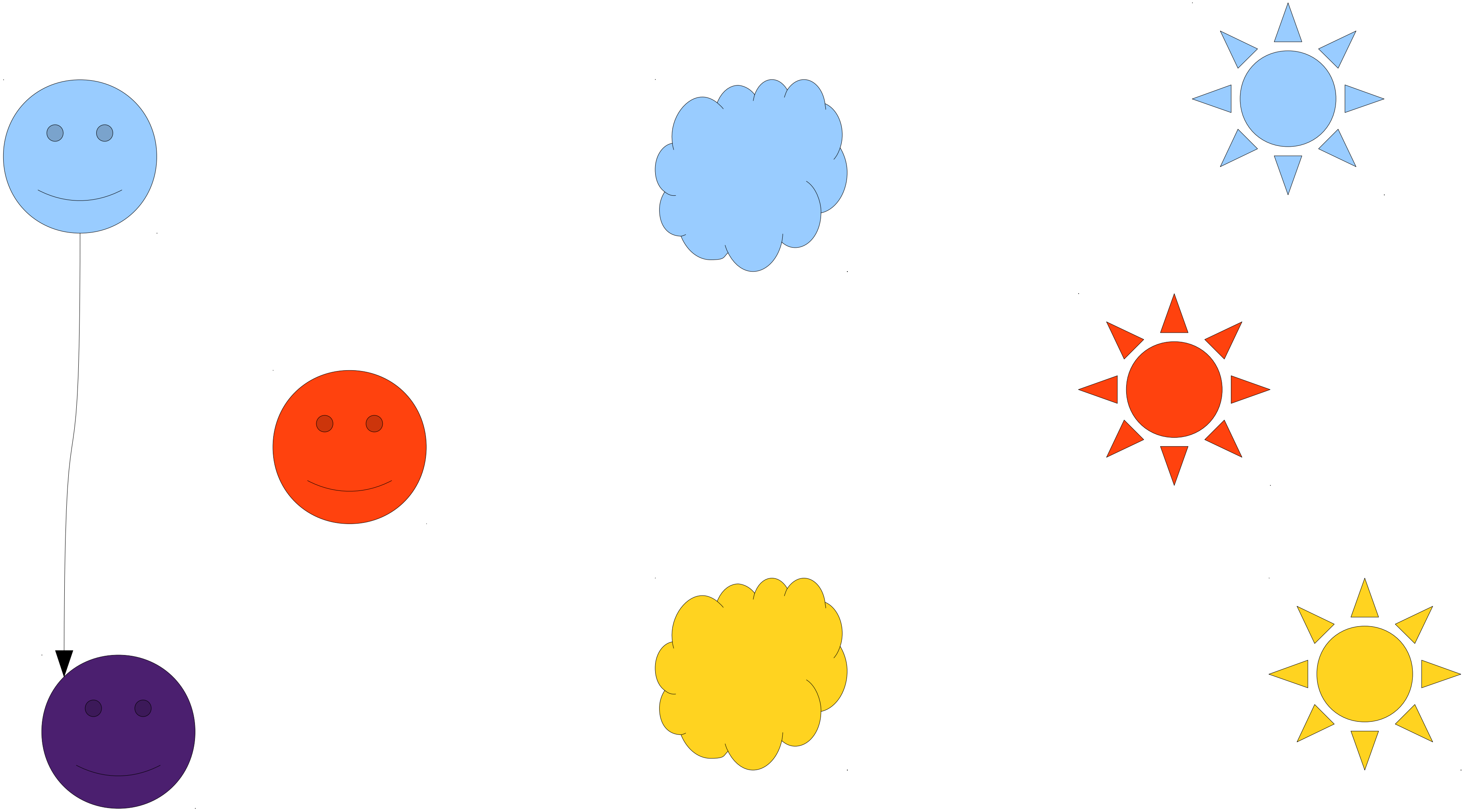
xRy ≡ x and y have the same shape.



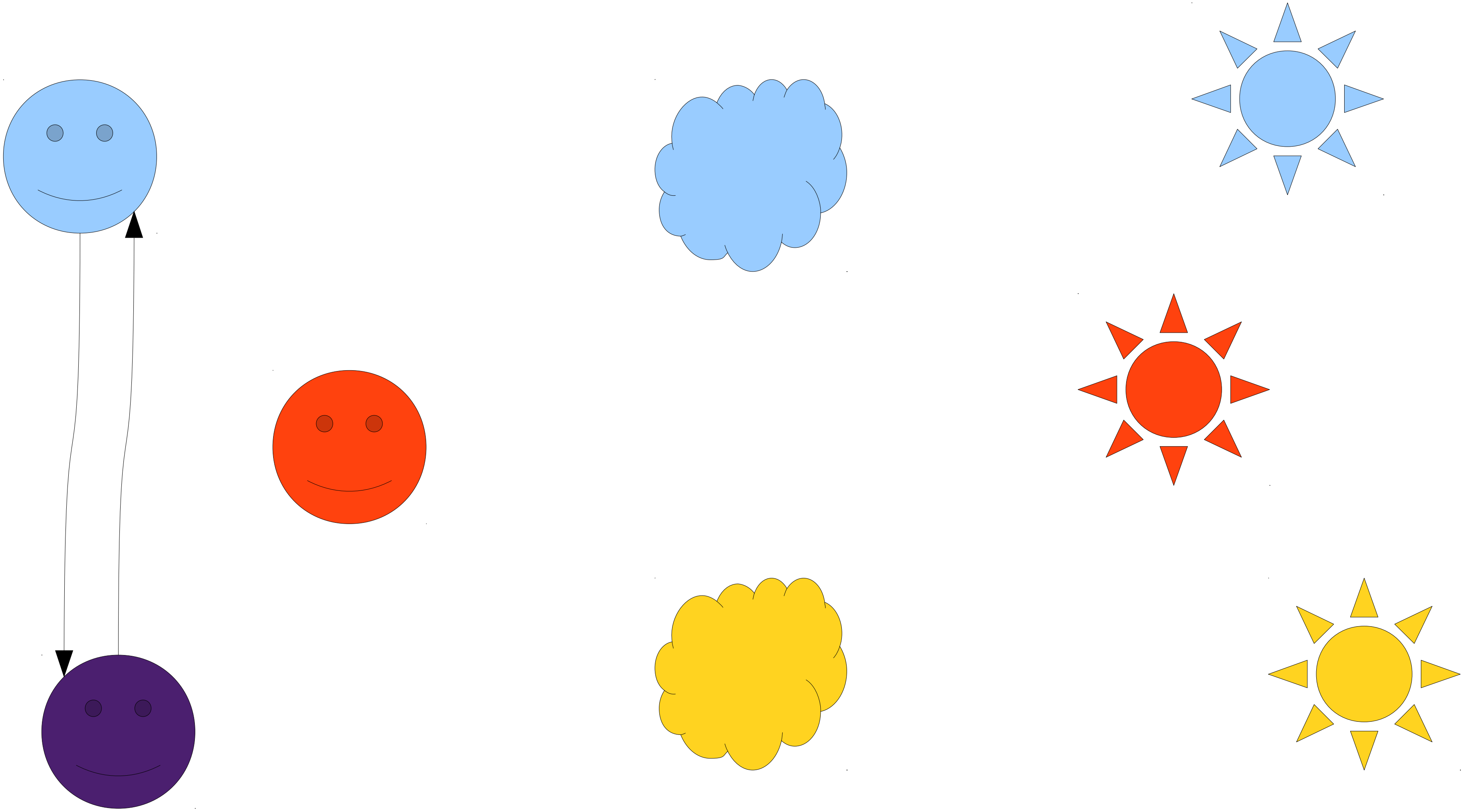
xRy ≡ x and y have the same shape.



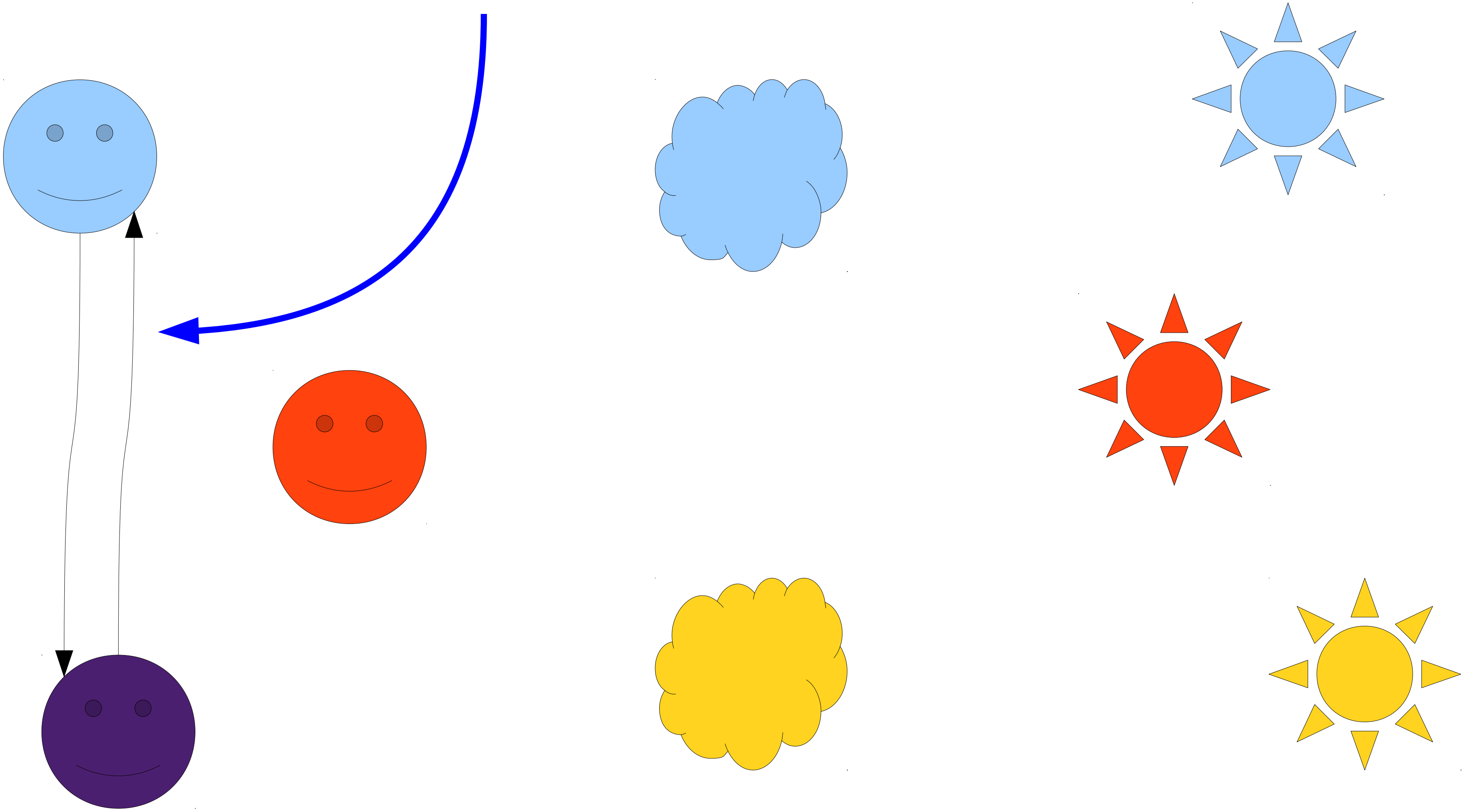
xRy ≡ x and y have the same shape.



xRy ≡ x and y have the same shape.

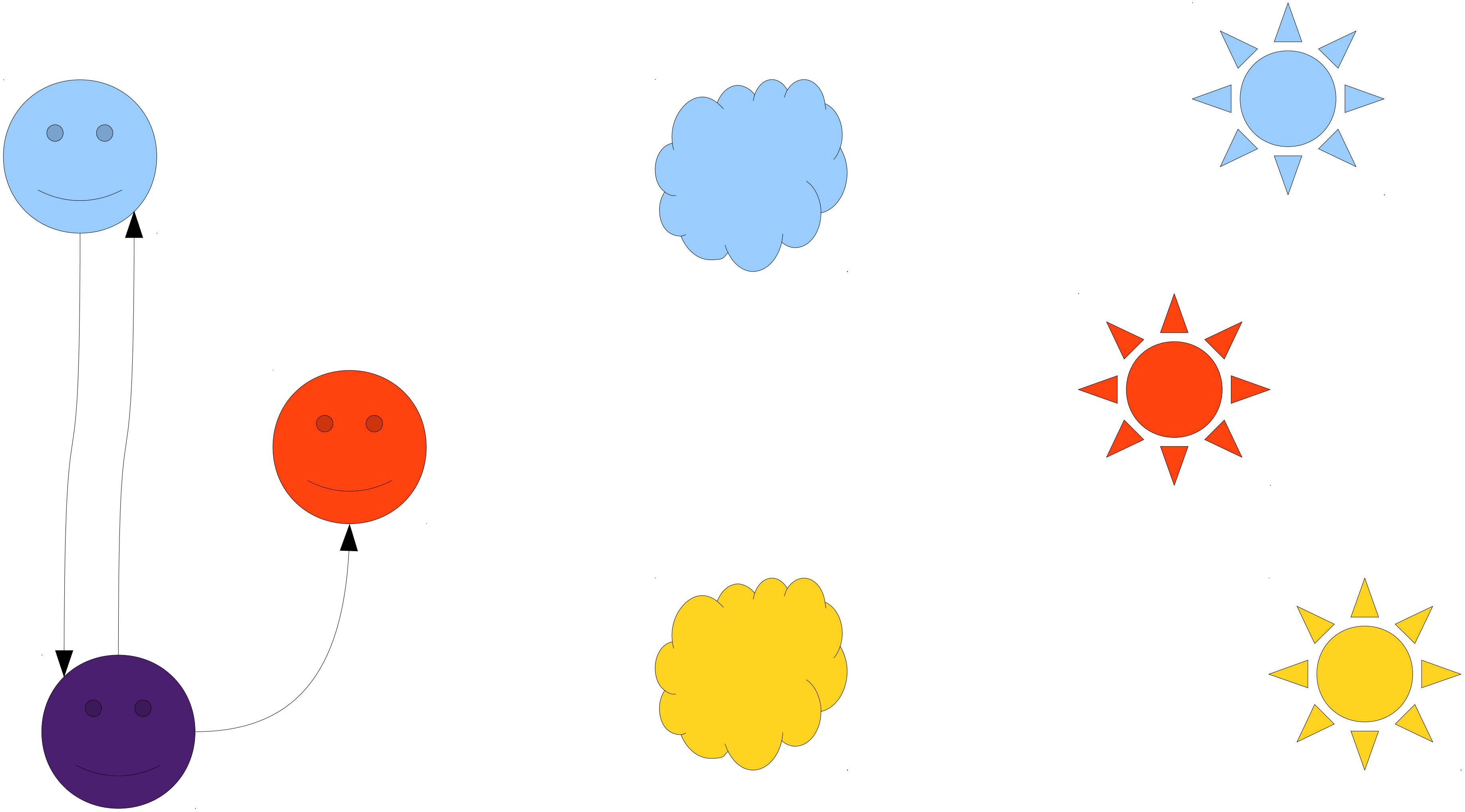


xRy ≡ x and y have the same shape.

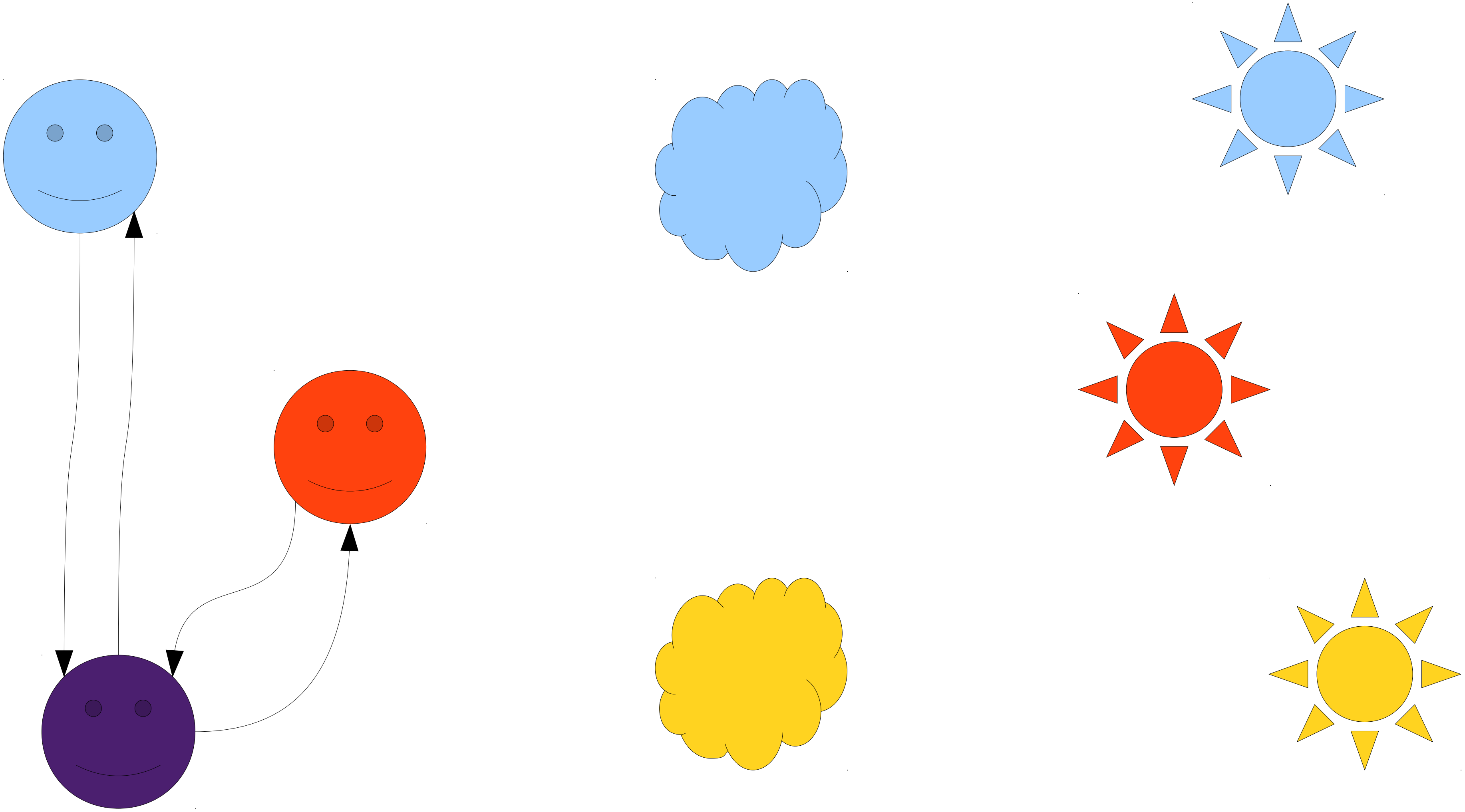


What property says this edge must be here?

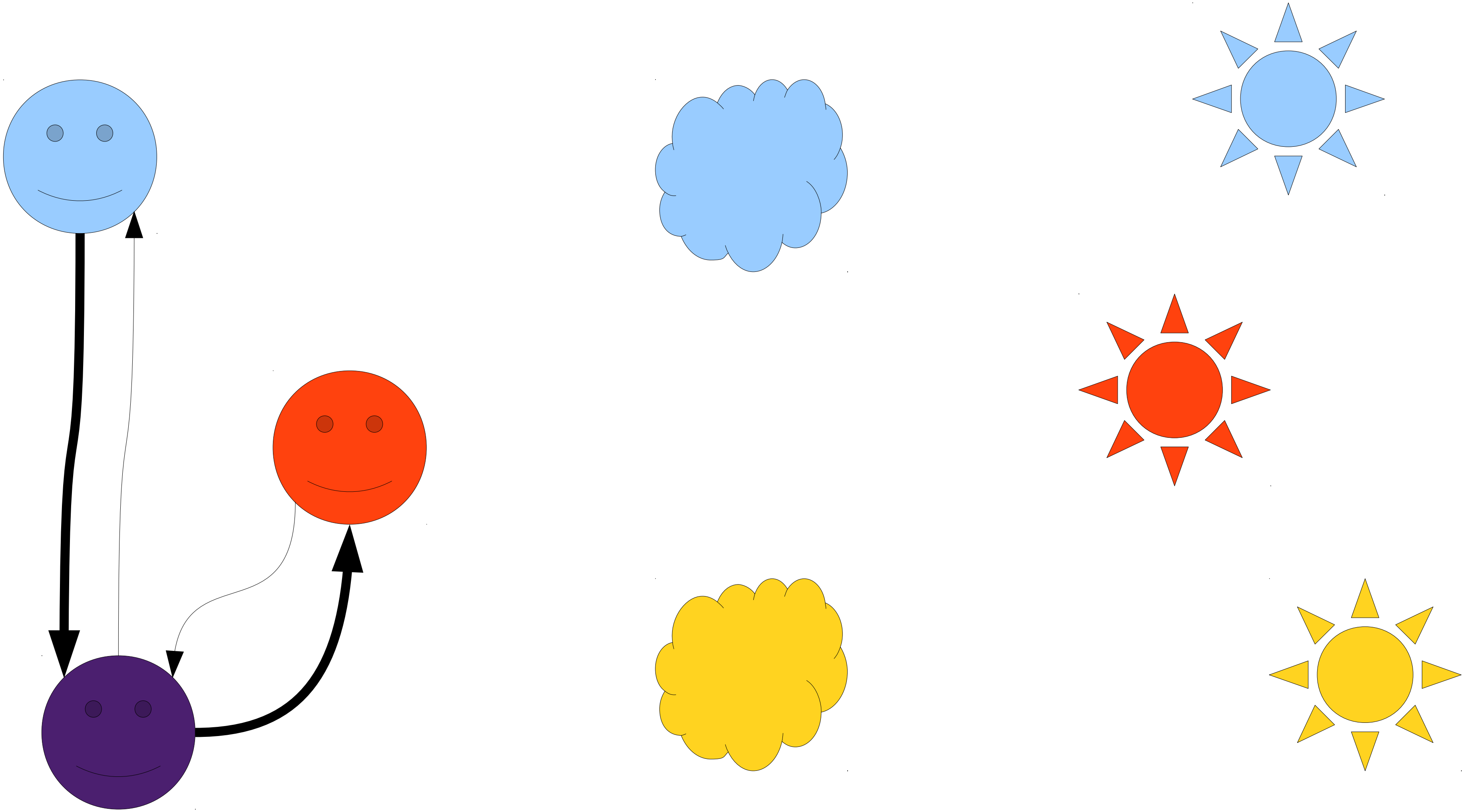
xRy ≡ x and y have the same shape.



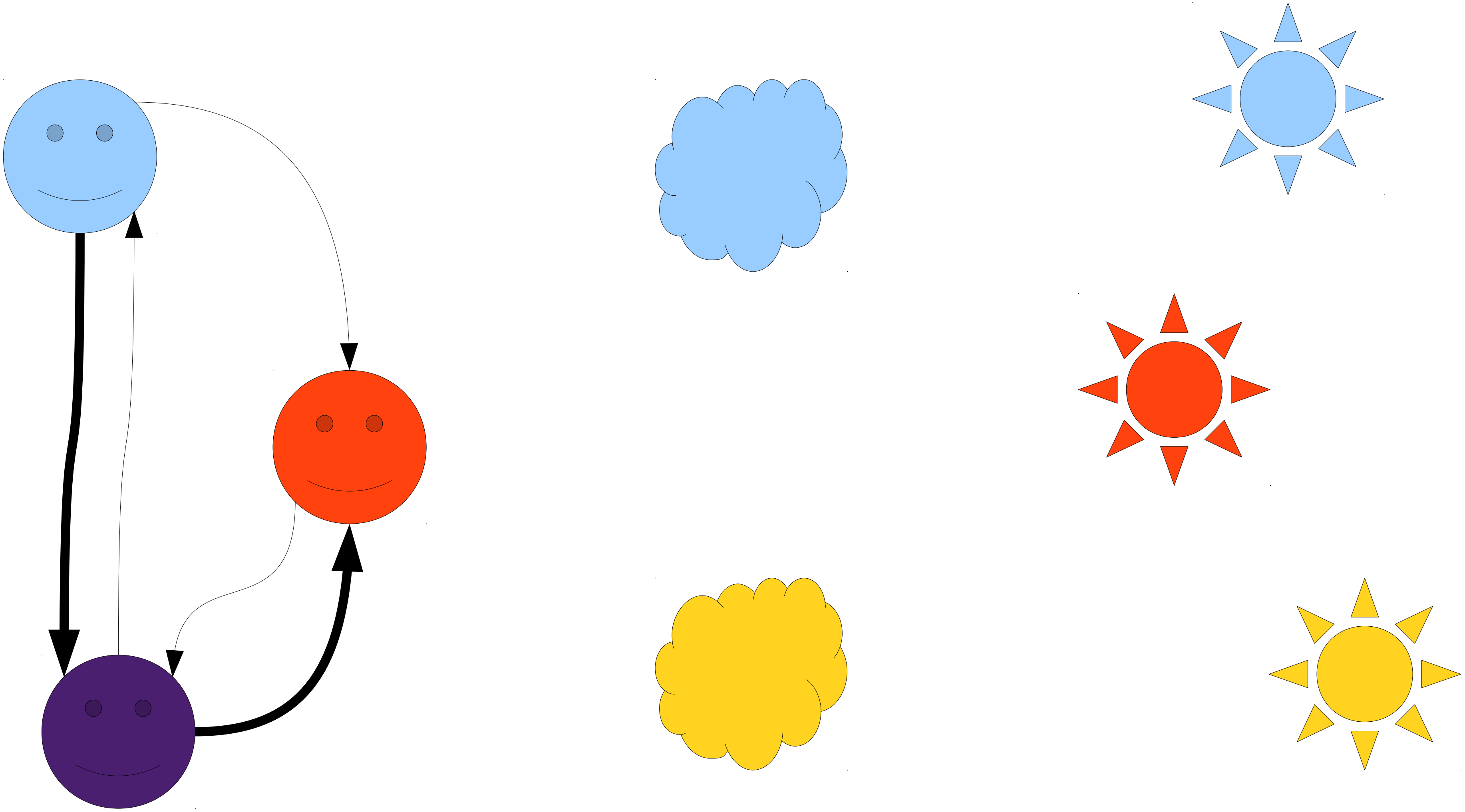
$xRy \equiv x$ and y have the same shape.



xRy ≡ x and y have the same shape.

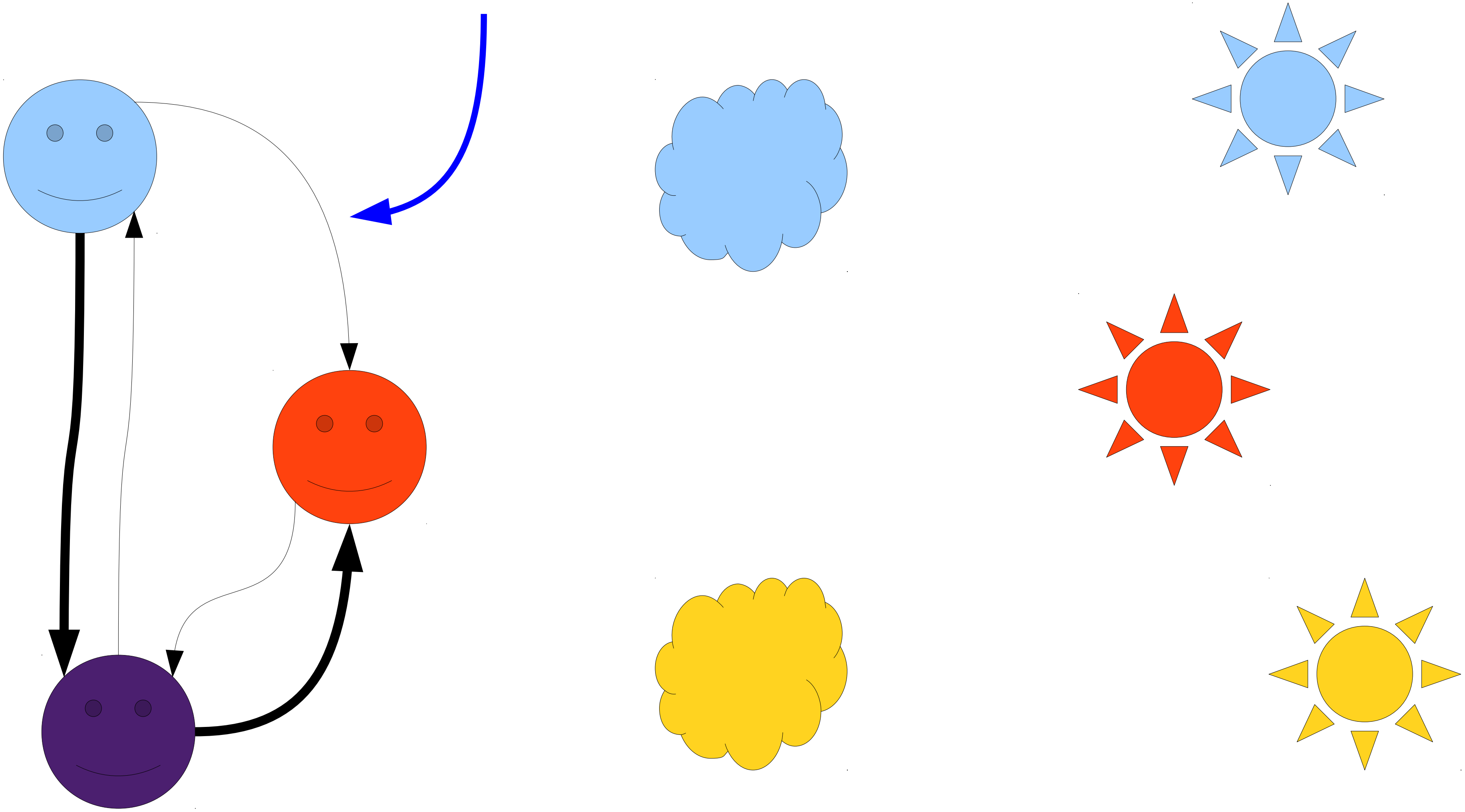


xRy ≡ x and y have the same shape.

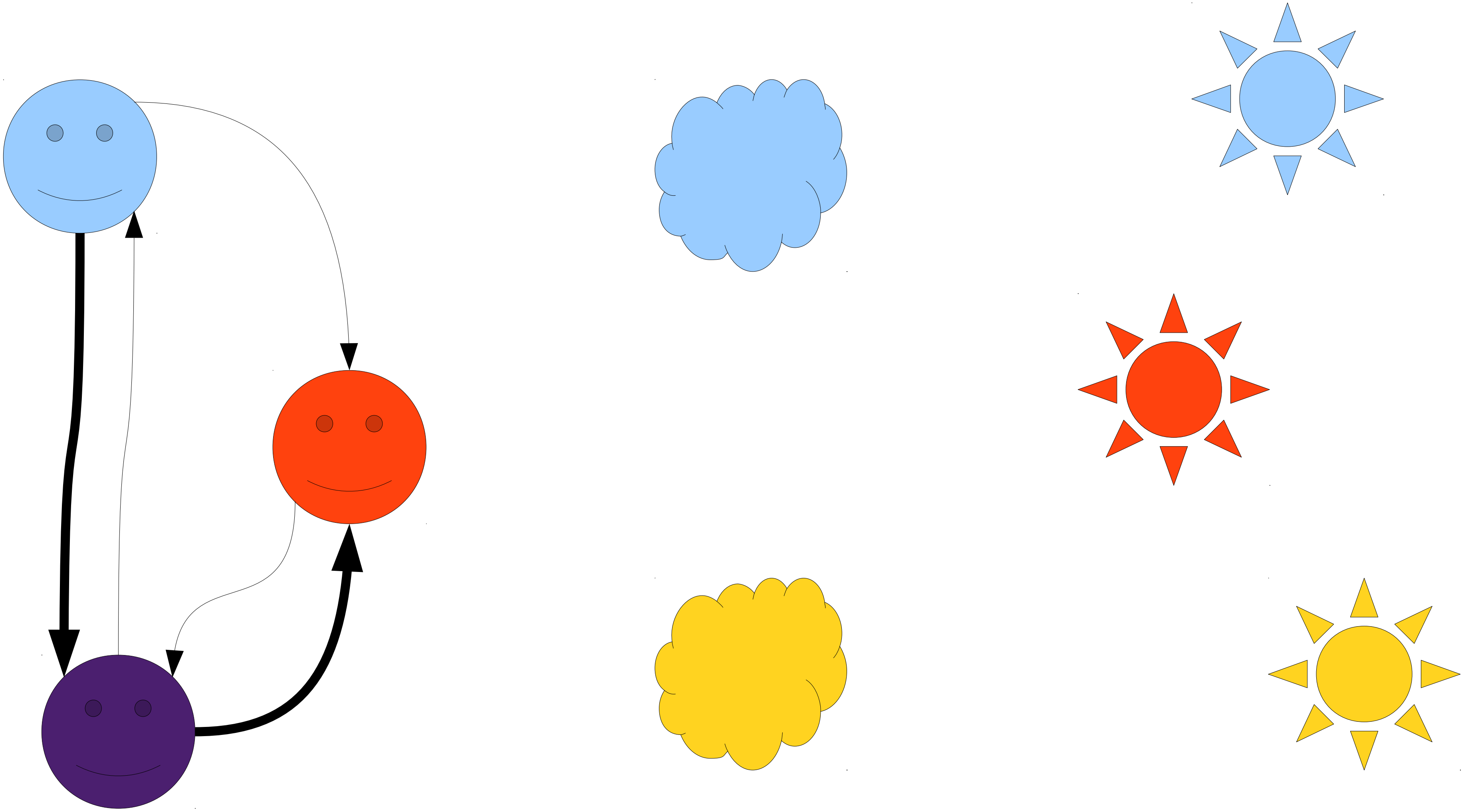


xRy ≡ x and y have the same shape.

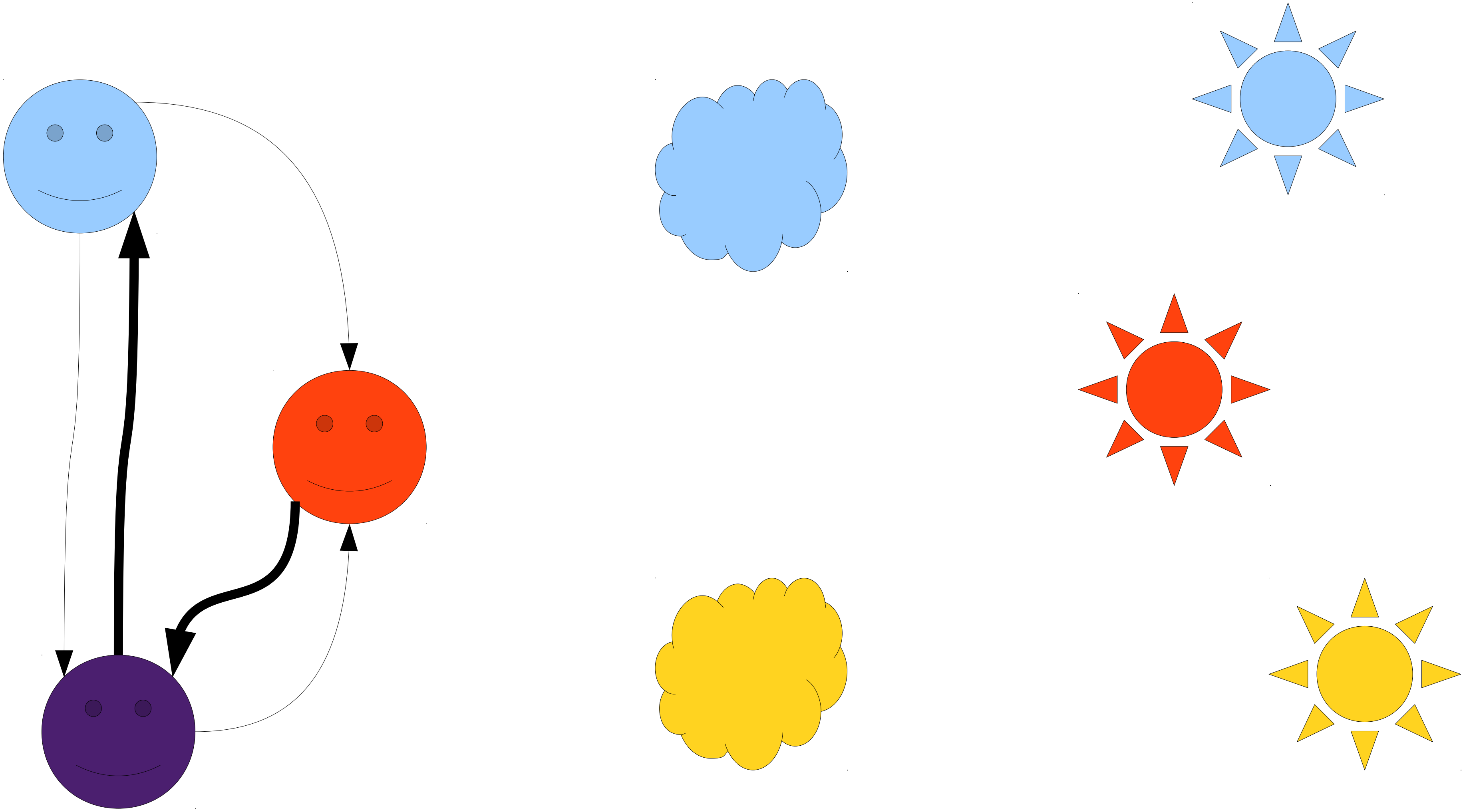
What property says this edge must be here?



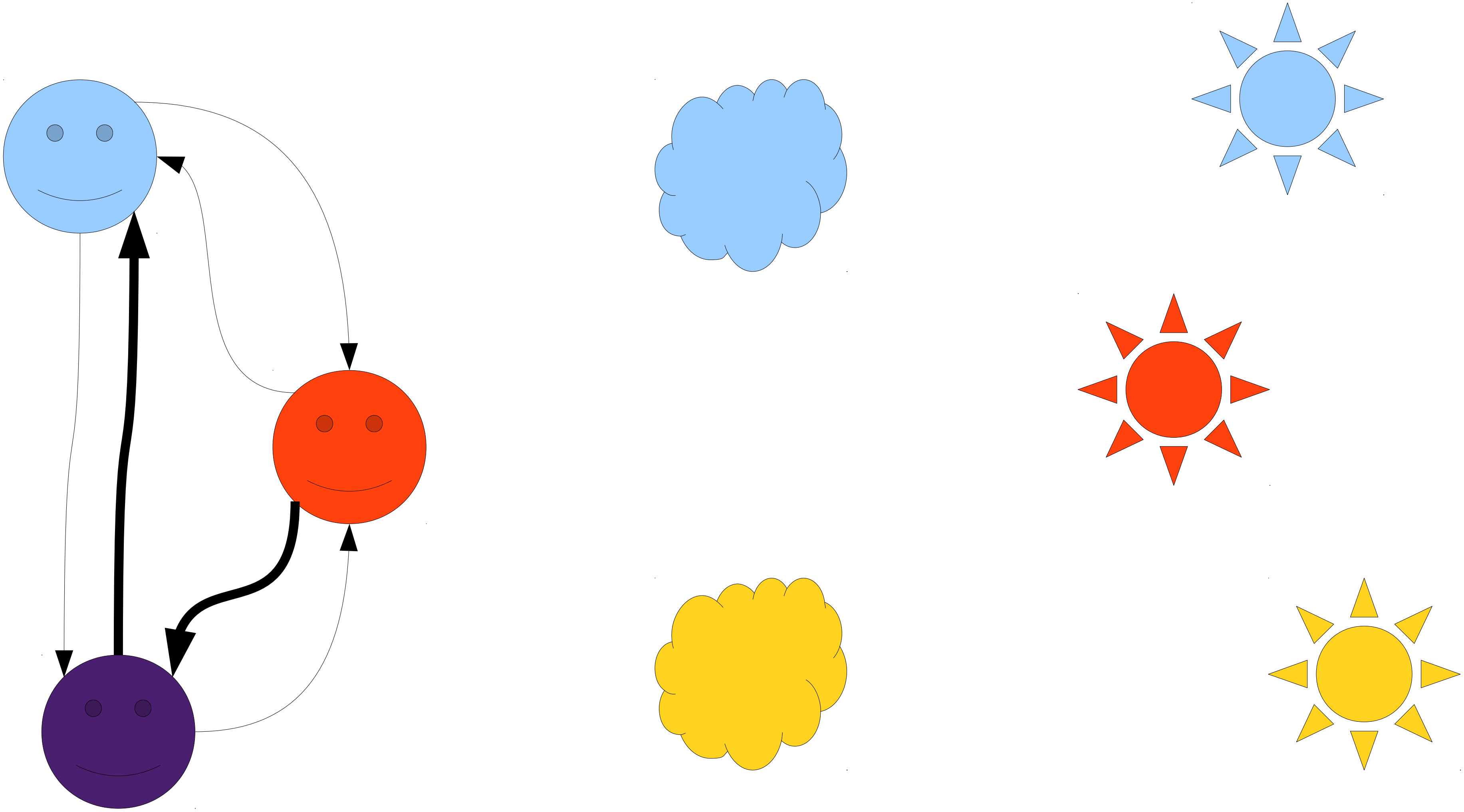
xRy ≡ x and y have the same shape.



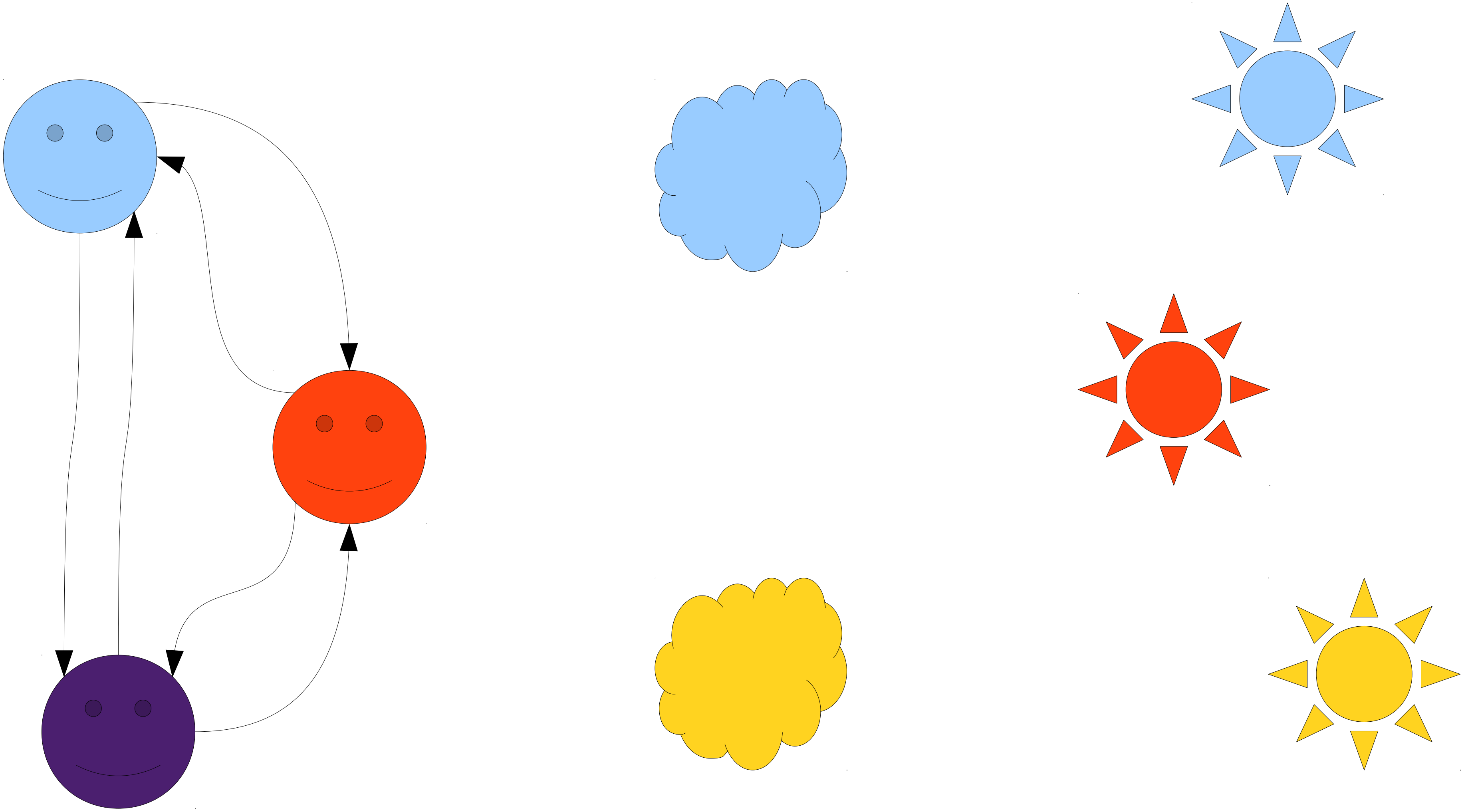
xRy ≡ x and y have the same shape.



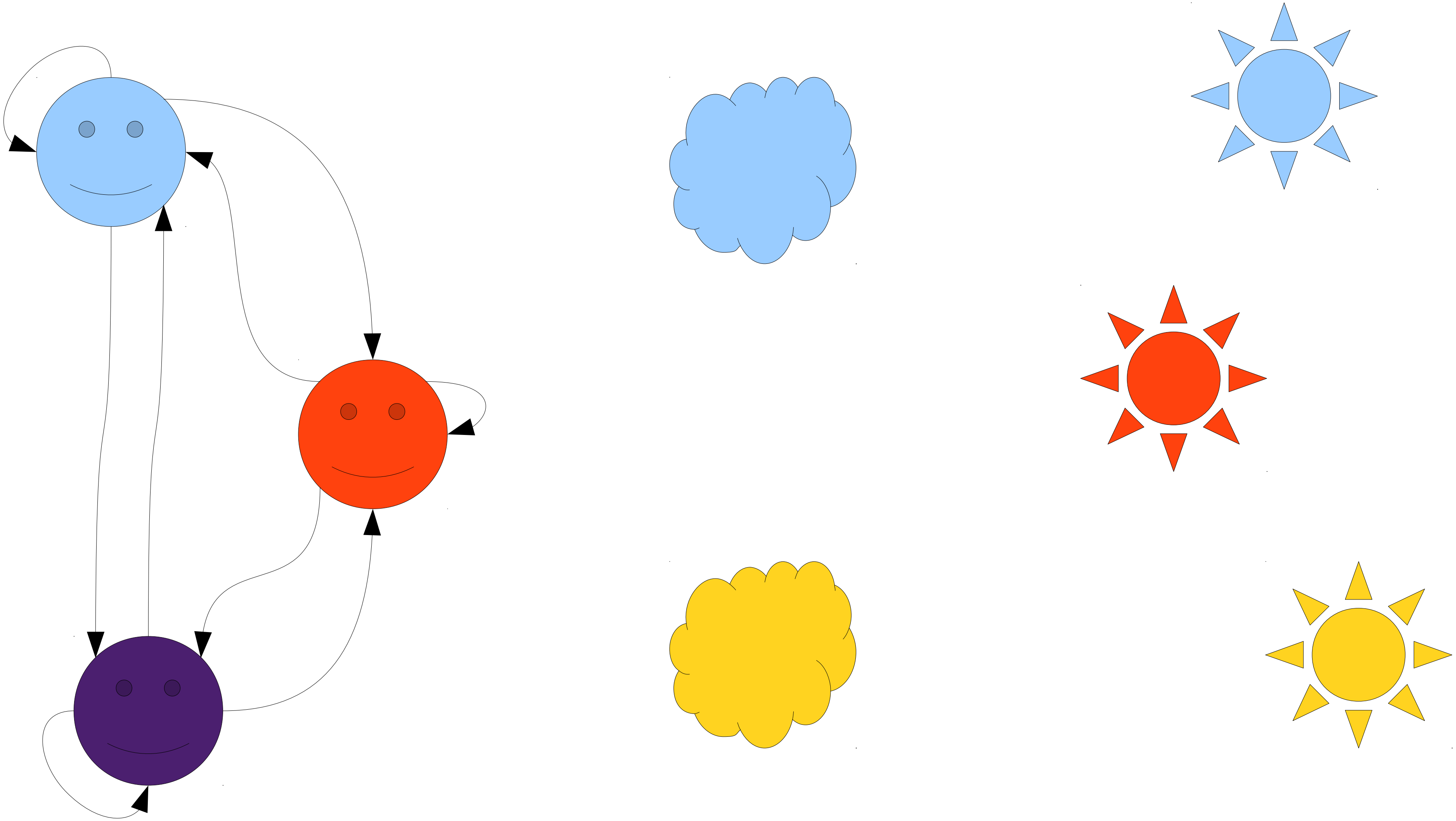
xRy ≡ x and y have the same shape.



xRy ≡ x and y have the same shape.

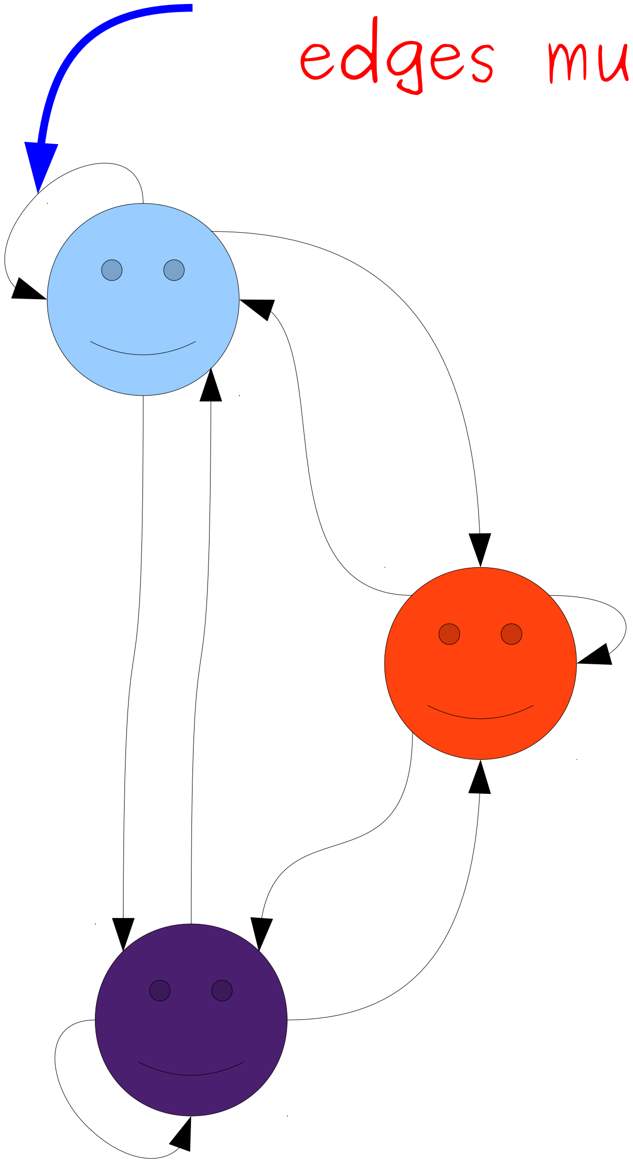


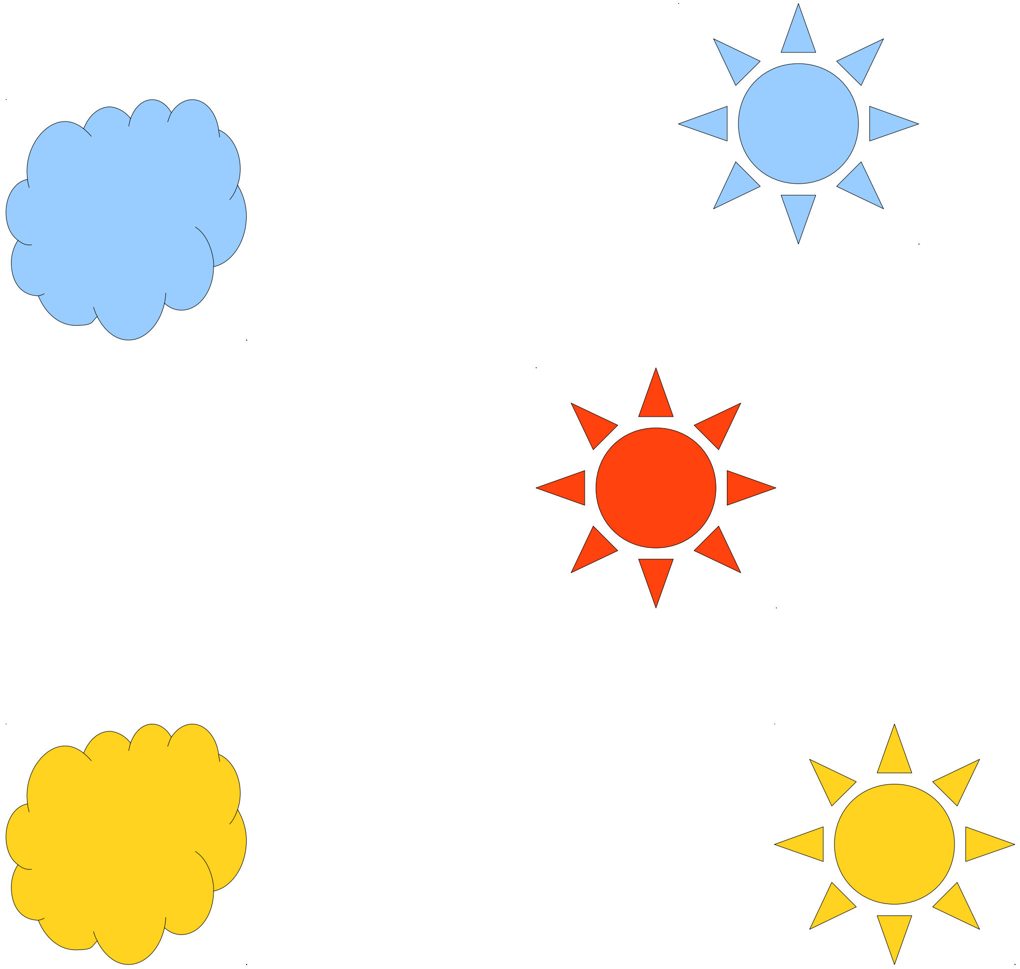
xRy ≡ x and y have the same shape.



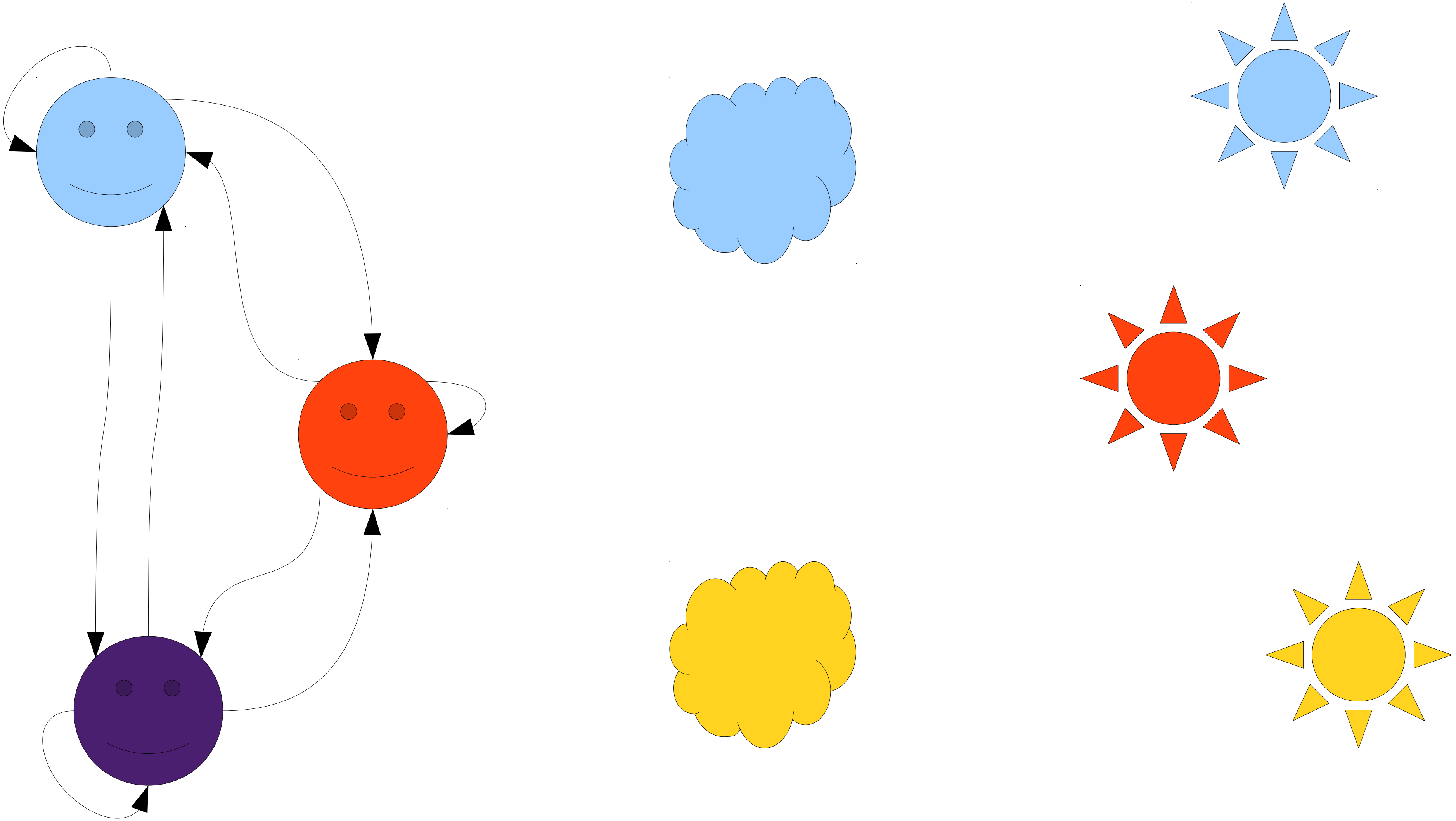
xRy ≡ x and y have the same shape.

What property says these edges must be here?

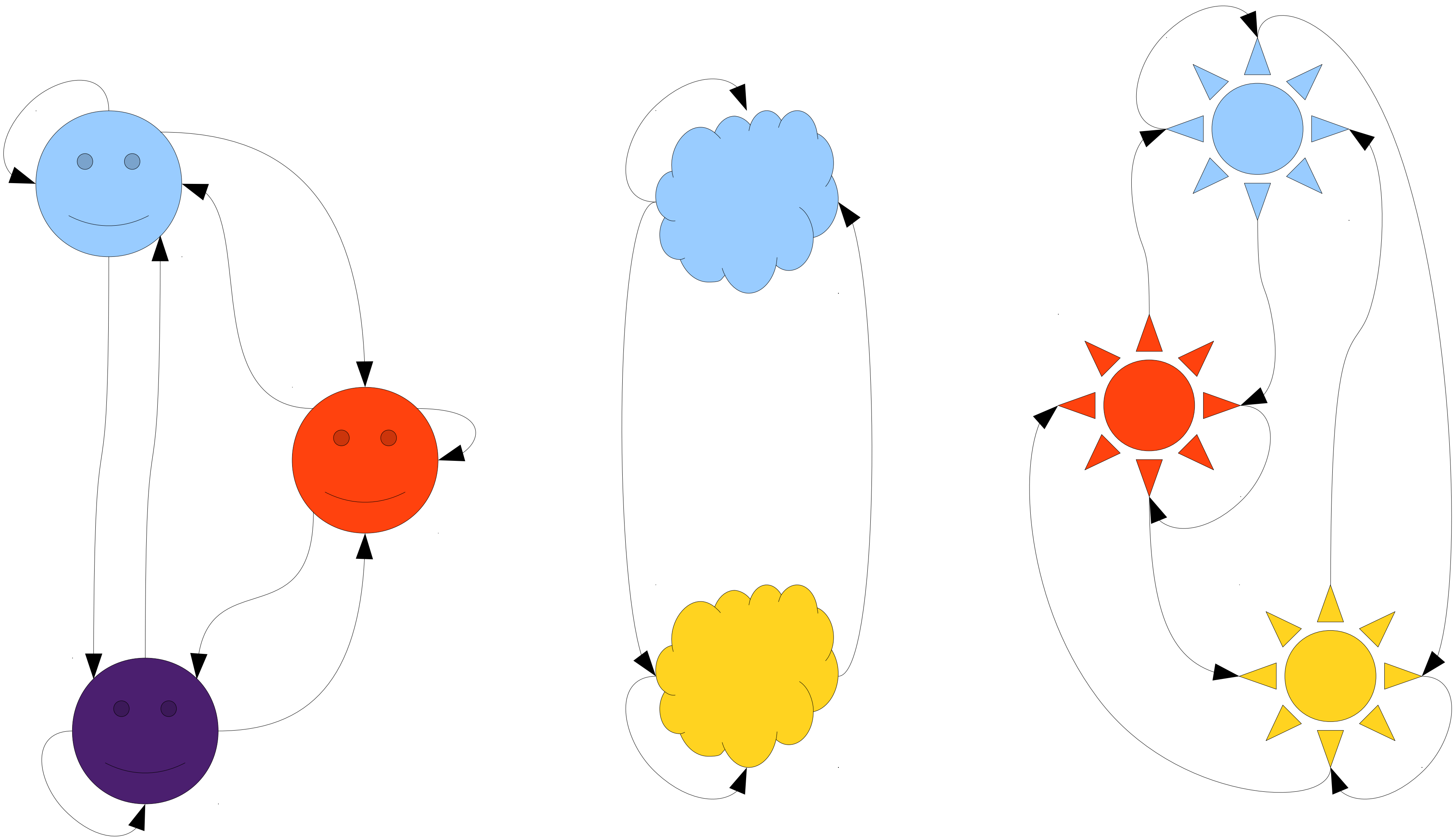




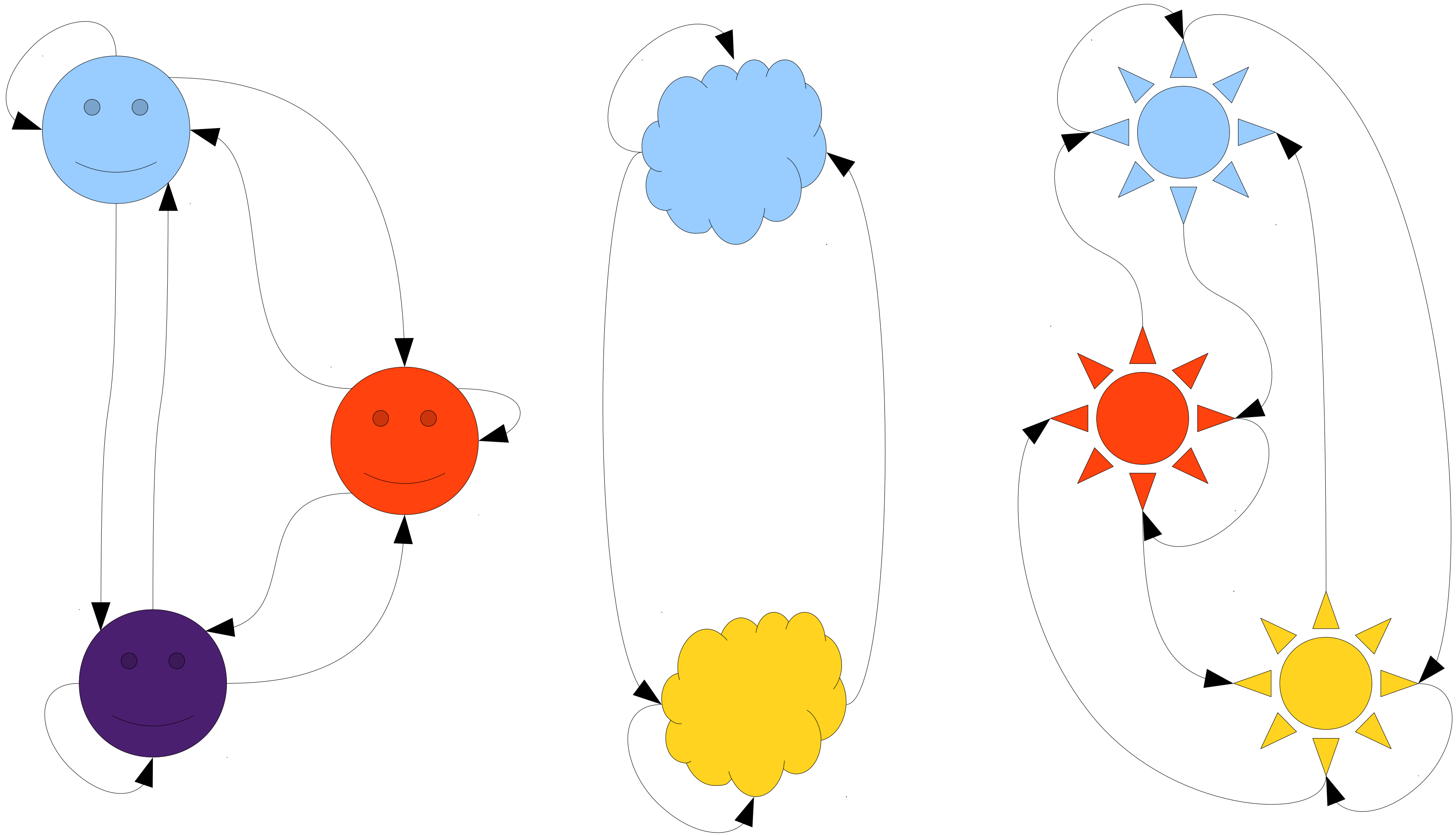
xRy ≡ x and y have the same shape.



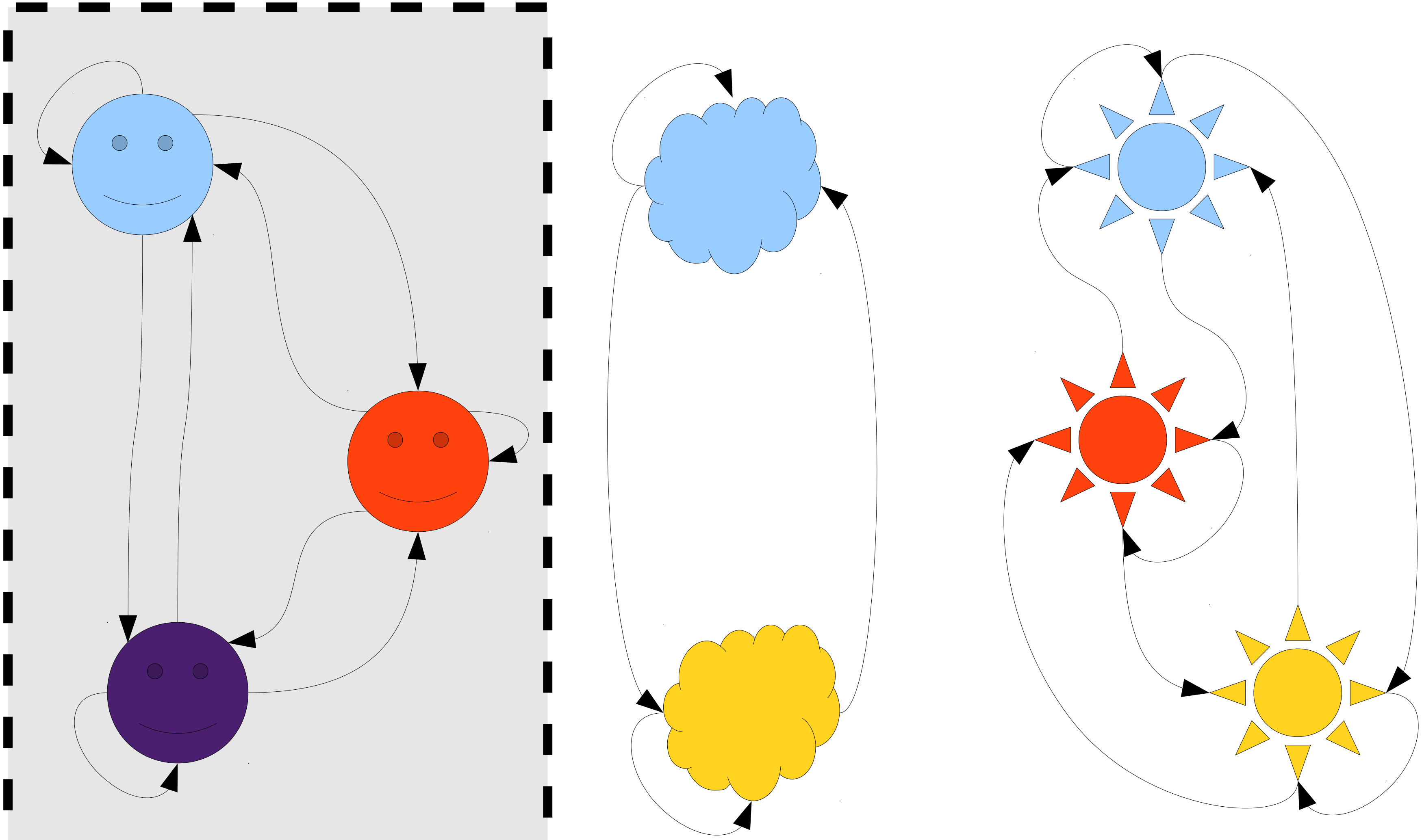
xRy ≡ x and y have the same shape.



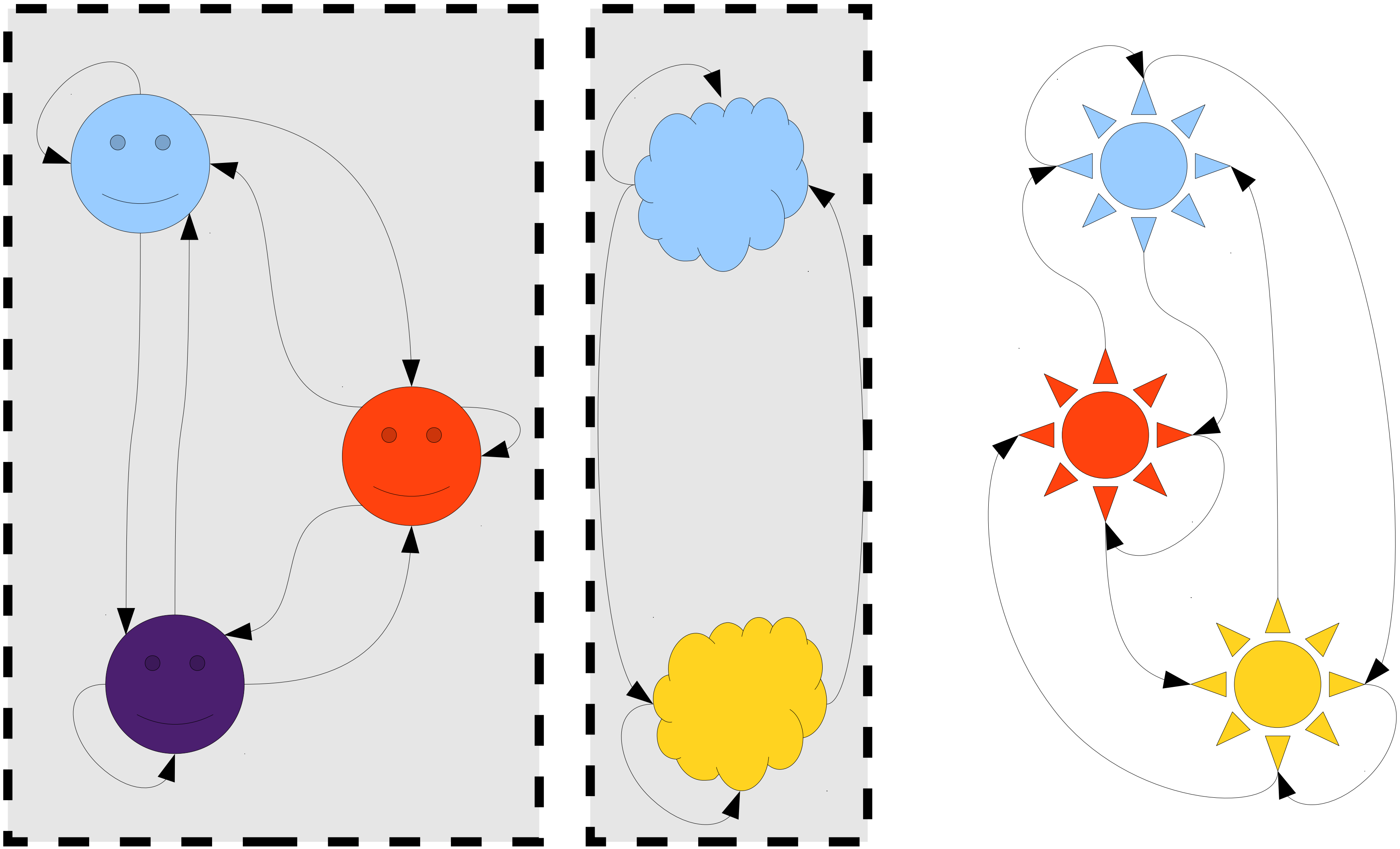
xRy ≡ x and y have the same shape.



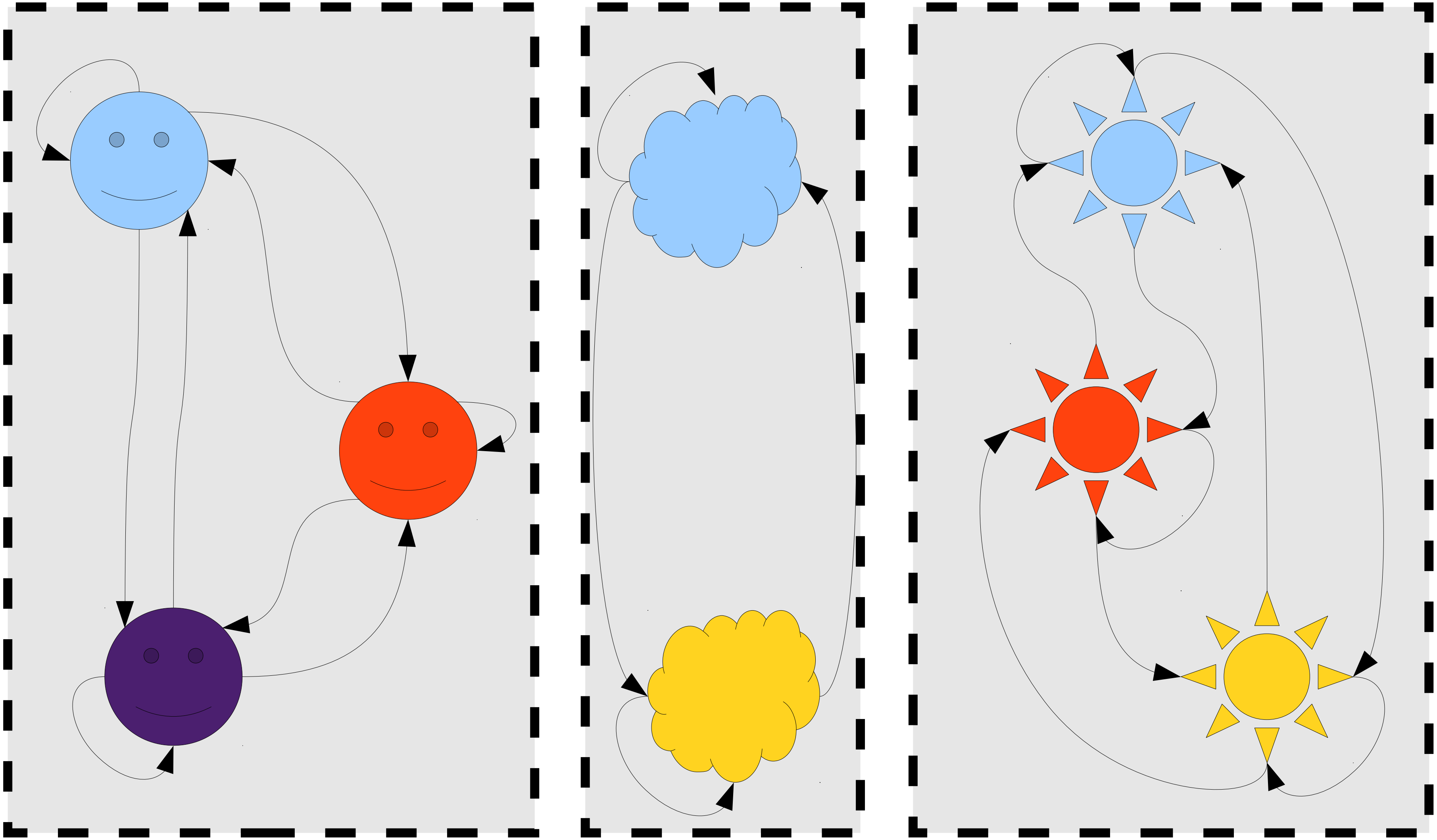
xRy ≡ x and y are the same shape.



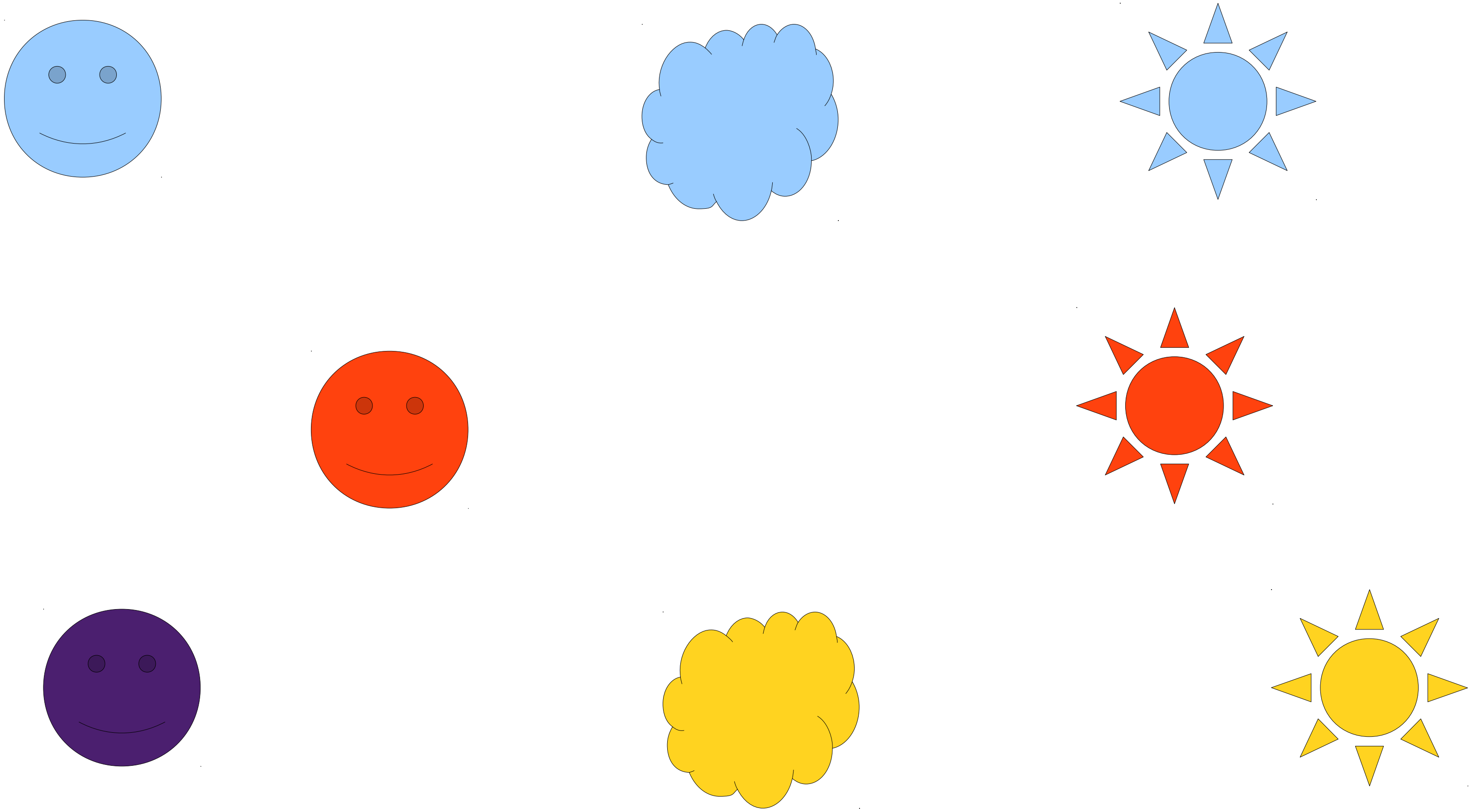
xRy ≡ x and y are the same shape.



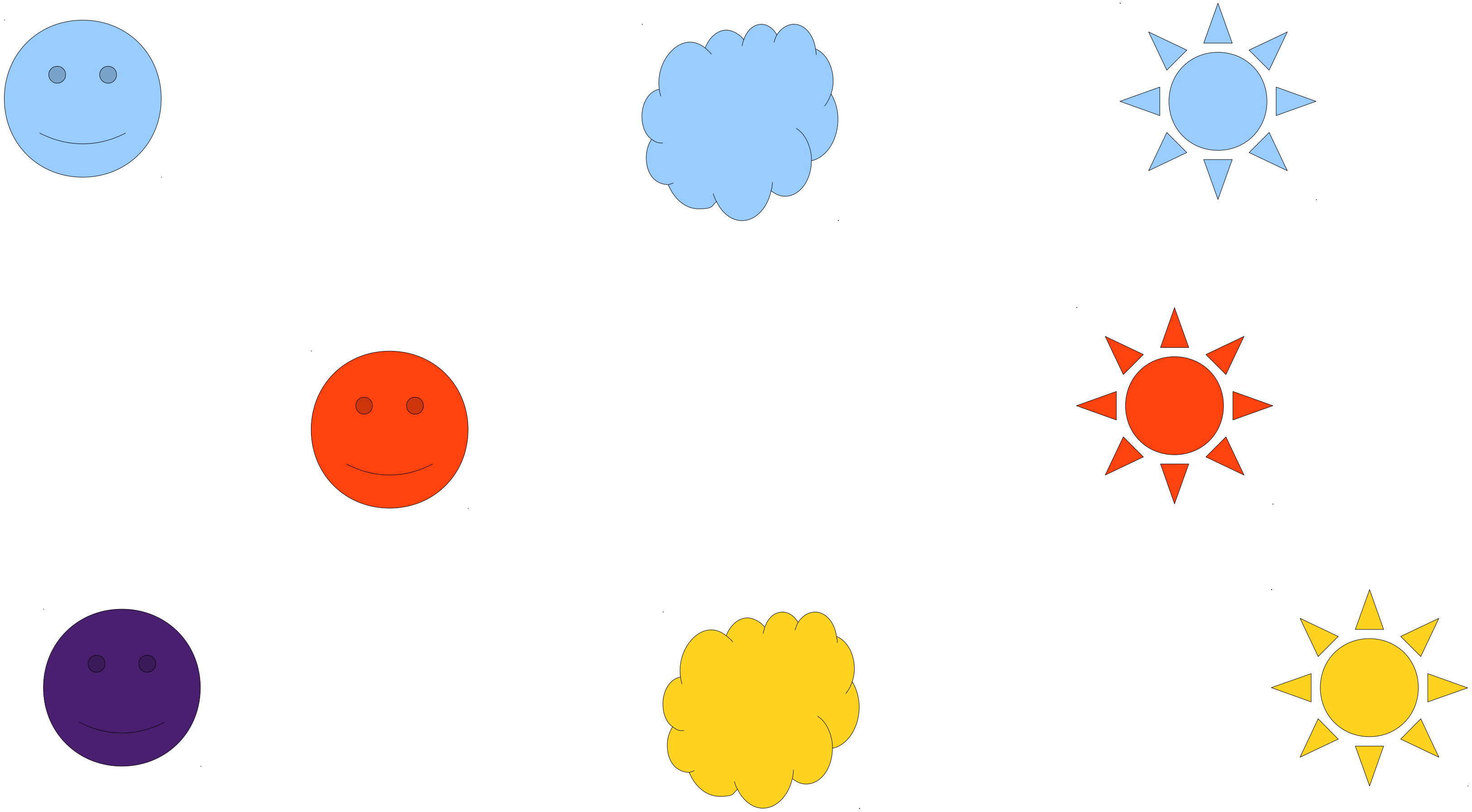
xRy ≡ x and y are the same shape.



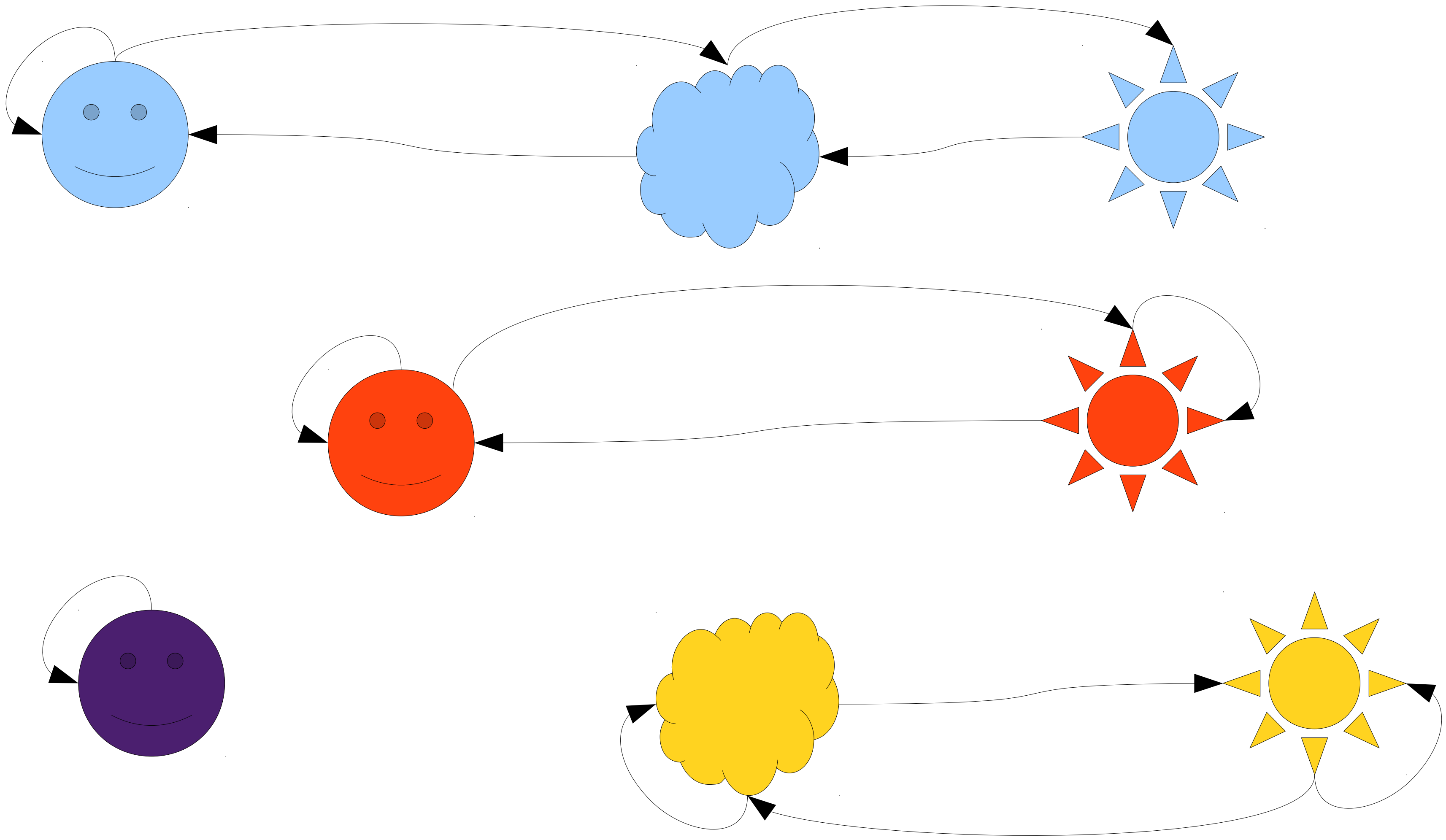
xRy ≡ x and y are the same shape.



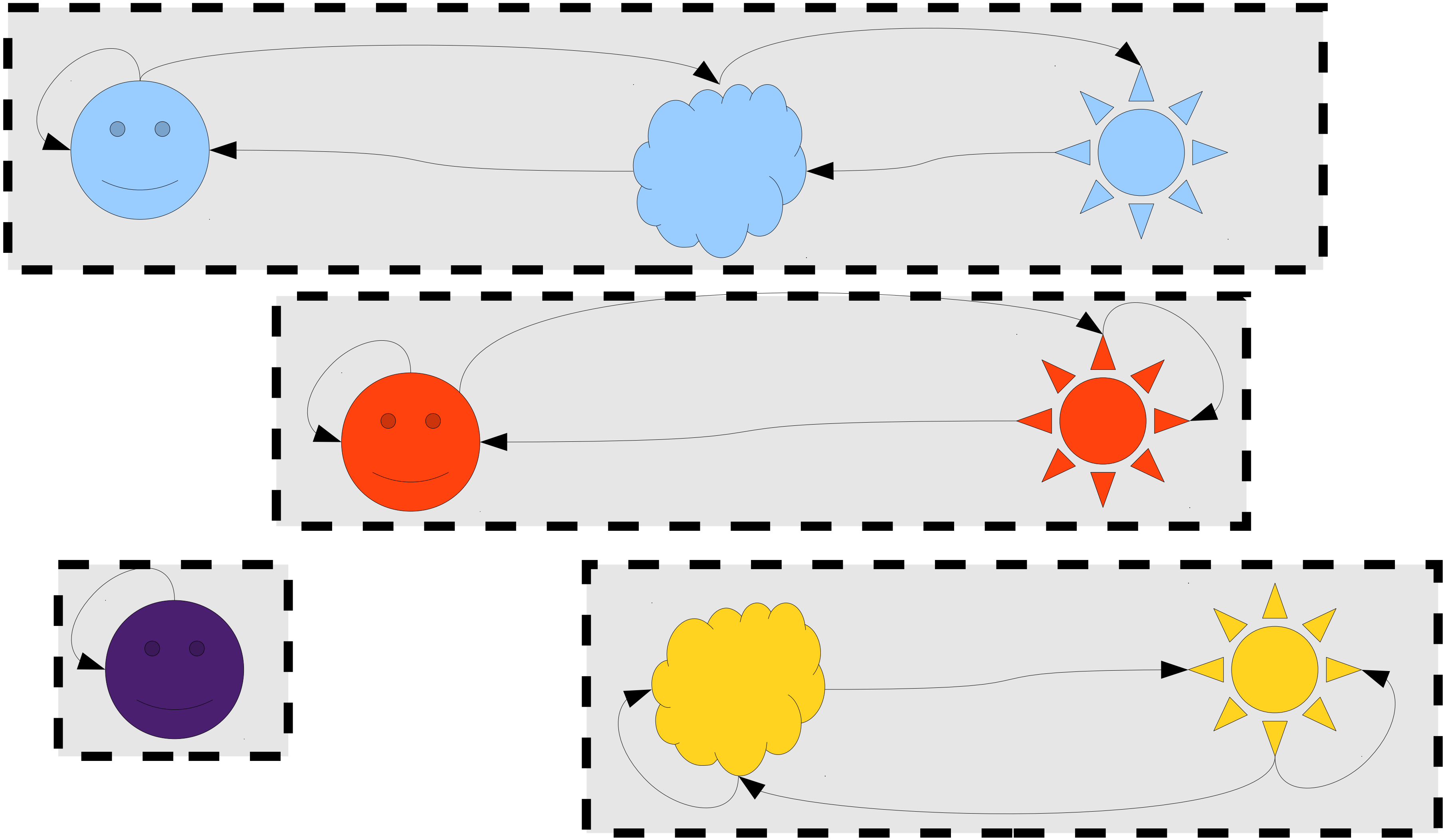
xRy ≡ x and y are the same shape.



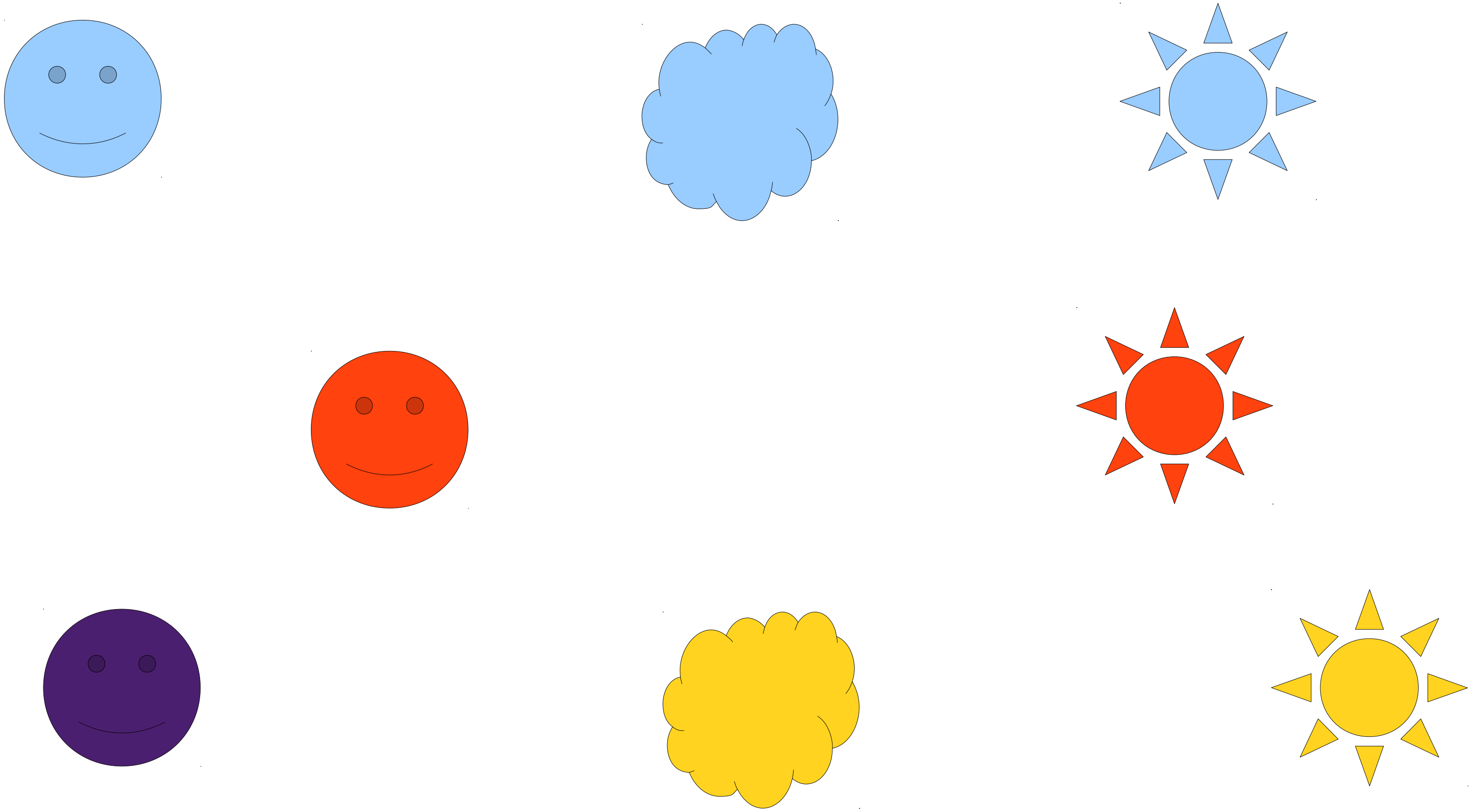
xRy ≡ x and y are the same **color**.



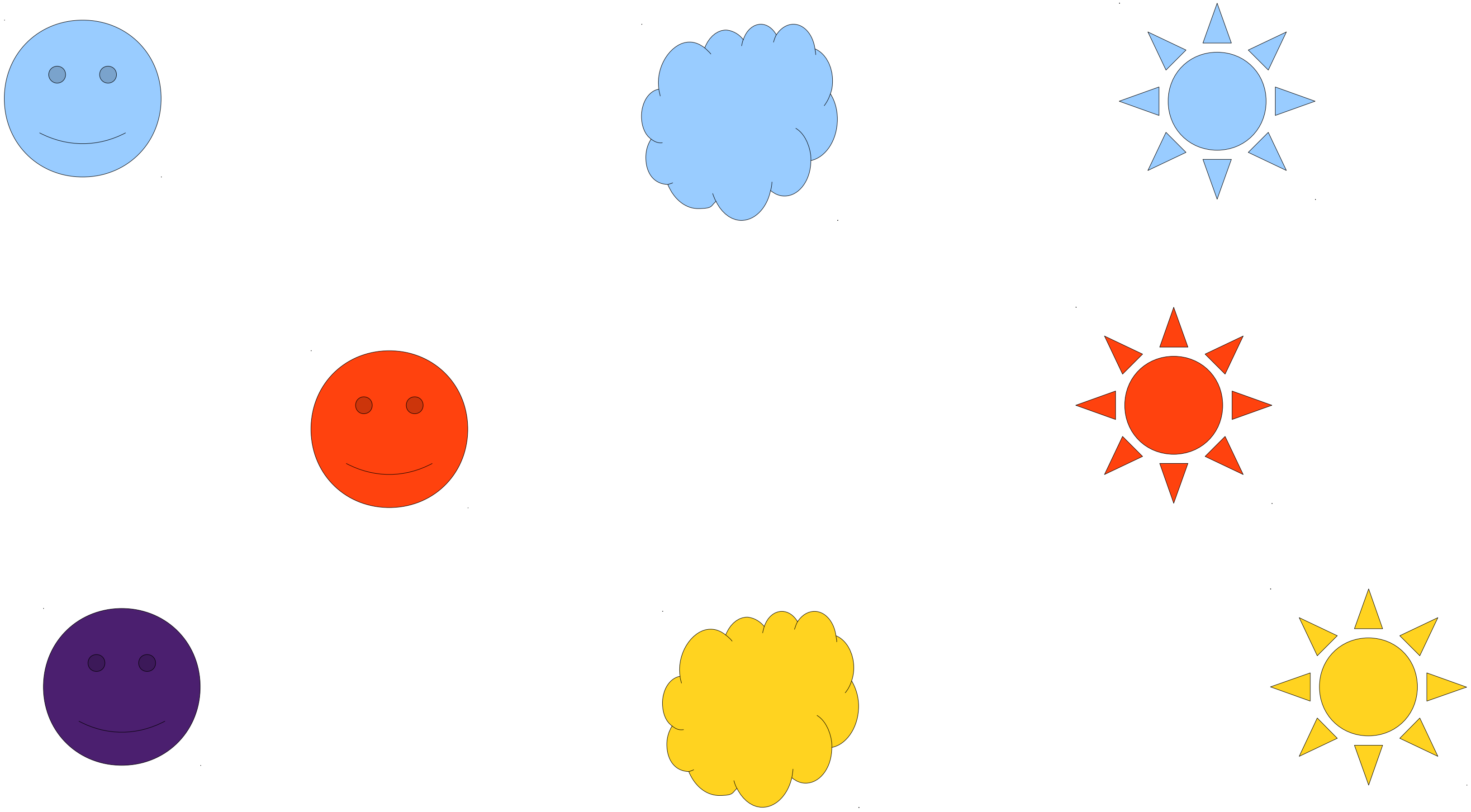
xRy ≡ x and y are the same **color**.



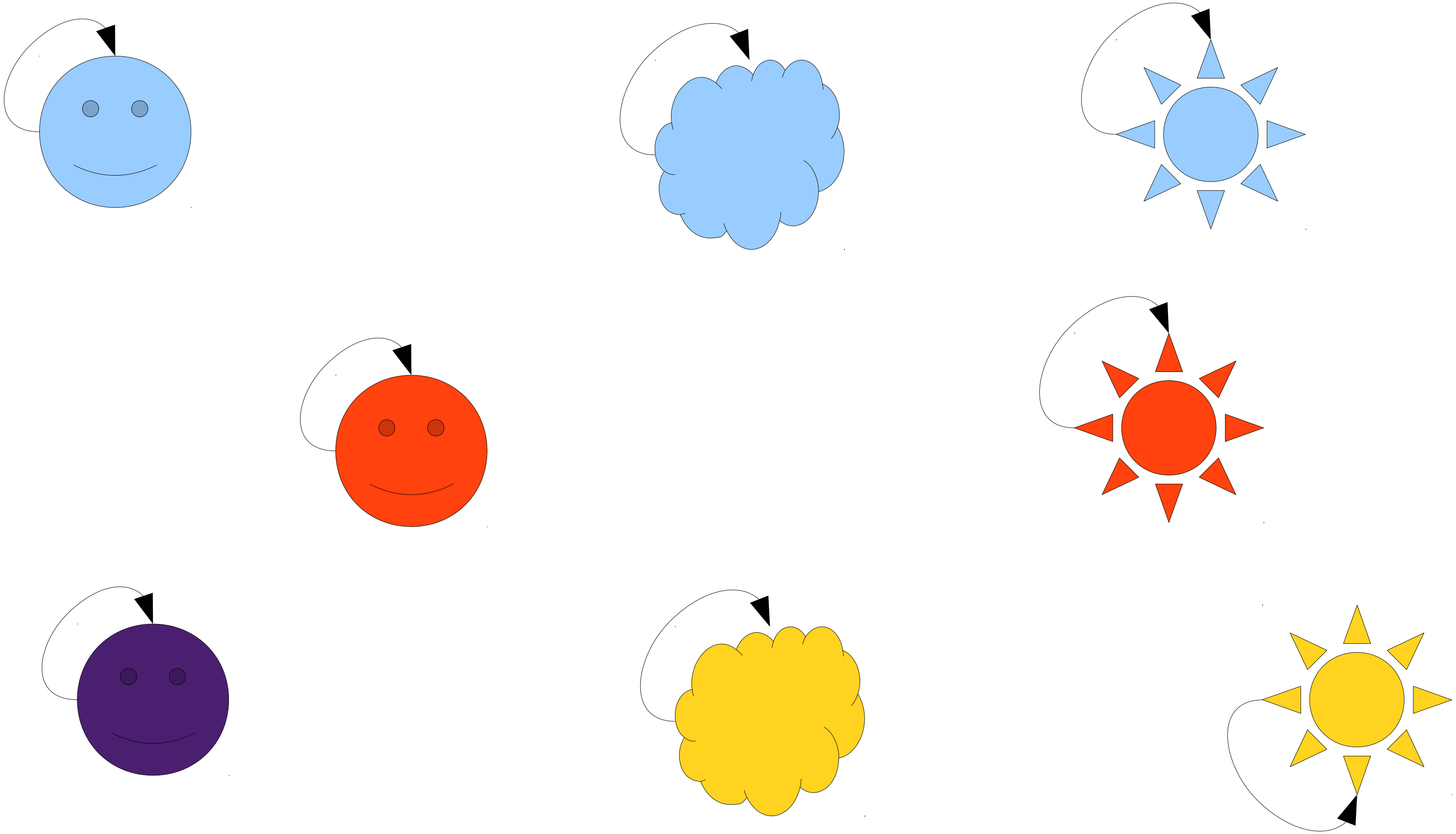
xRy ≡ x and y are the same **color**.



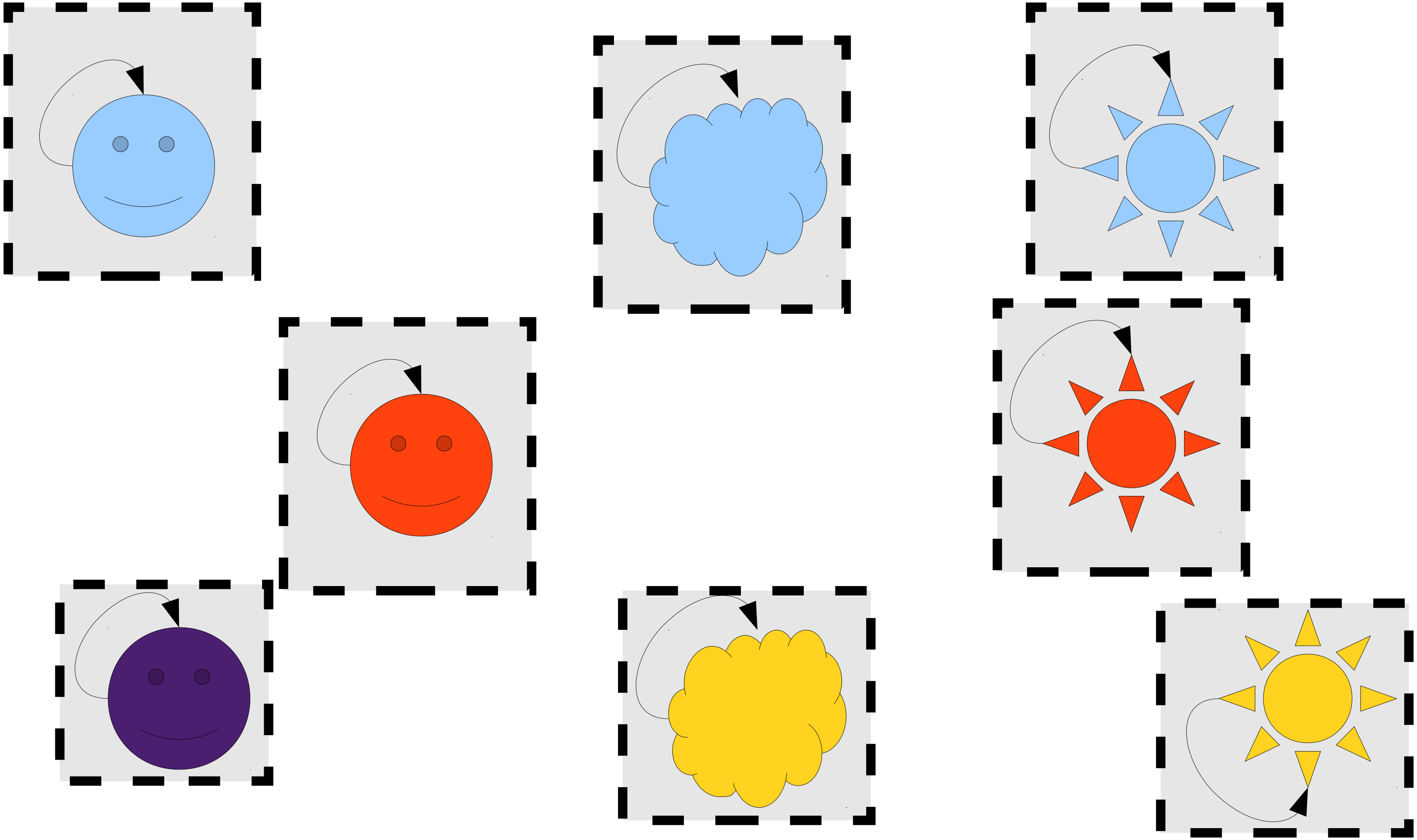
xRy ≡ x and y are the same **color**.



$xRy \equiv x = y$



$xRy \equiv x = y$



$$xRy \equiv x = y$$

Equivalence Classes

- Given an equivalence relation *R* over a set *A*, for any *a* ∈ *A*, the **equivalence class of *a*** is the set

$$[x]_R \equiv \{ a \mid a \in A \text{ and } xRa \}$$

- Informally, the set of all elements equal to *a*.
- *R* **partitions** the set *A* into a set of equivalence classes.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Theorem: Let *R* be an equivalence relation over a set *A*. Then every element of *A* belongs to exactly one equivalence class.

Theorem: Let *R* be an equivalence relation over a set *A*. Then every element of *A* belongs to exactly one equivalence class.

How do we prove this?

Existence and Uniqueness

- The proof we are attempting is a type of proof called an **existence and uniqueness** proof.
- We need to show that for any $a \in A$, there **exists** an equivalence class containing *a* and that this equivalence class is **unique**.
- These are two completely separate steps.

Proving Existence

- To prove **existence**, we need to show that for any $a \in A$, that a belongs to at least one equivalence class.
- This is just a proof of an existential statement.
- Can we find an equivalence class containing a?

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Theorem: Let *R* be an equivalence relation over a set *A*. Then every element of *A* belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$. Since our choice of a was arbitrary, every $a \in A$ belongs to at least one equivalence class – namely, $[a]_R$.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$. Since our choice of a was arbitrary, every $a \in A$ belongs to at least one equivalence class – namely, $[a]_R$.

To see that every $a \in A$ belongs to at most one equivalence class, **?????**

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$. Since our choice of a was arbitrary, every $a \in A$ belongs to at least one equivalence class – namely, $[a]_R$.

To see that every $a \in A$ belongs to at most one equivalence class, **?????**

How do we prove this?

Proving Uniqueness

- To prove that there is a **unique** object with some property, we can do the following:
  - Consider any two arbitrary objects *x* and *y* with that property.
  - Show that $x = y$.
  - Conclude, therefore, that there is only one object with that property, and we just gave it two different names.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$. Since our choice of a was arbitrary, every $a \in A$ belongs to at least one equivalence class – namely, $[a]_R$.

To see that every $a \in A$ belongs to at most one equivalence class, **?????**

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$. Since our choice of a was arbitrary, every $a \in A$ belongs to at least one equivalence class – namely, $[a]_R$.

To see that every $a \in A$ belongs to at most one equivalence class, we show that for any $a \in A$, if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R = [y]_R$.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$. Since our choice of a was arbitrary, every $a \in A$ belongs to at least one equivalence class – namely, $[a]_R$.

To see that every $a \in A$ belongs to at most one equivalence class, we show that for any $a \in A$, if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R = [y]_R$. To do this, we prove that if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R \subseteq [y]_R$.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$. Since our choice of a was arbitrary, every $a \in A$ belongs to at least one equivalence class – namely, $[a]_R$.

To see that every $a \in A$ belongs to at most one equivalence class, we show that for any $a \in A$, if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R = [y]_R$. To do this, we prove that if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R \subseteq [y]_R$. By interchanging $[x]_R$ and $[y]_R$, we can conclude that $[y]_R \subseteq [x]_R$, from which we have $[x]_R = [y]_R$.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$. Since our choice of a was arbitrary, every $a \in A$ belongs to at least one equivalence class – namely, $[a]_R$.

To see that every $a \in A$ belongs to at most one equivalence class, we show that for any $a \in A$, if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R = [y]_R$. To do this, we prove that if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R \subseteq [y]_R$. By interchanging $[x]_R$ and $[y]_R$, we can conclude that $[y]_R \subseteq [x]_R$, from which we have $[x]_R = [y]_R$.

Assume that $a \in [x]_R$ and $a \in [y]_R$.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$. Since our choice of a was arbitrary, every $a \in A$ belongs to at least one equivalence class – namely, $[a]_R$.

To see that every $a \in A$ belongs to at most one equivalence class, we show that for any $a \in A$, if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R = [y]_R$. To do this, we prove that if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R \subseteq [y]_R$. By interchanging $[x]_R$ and $[y]_R$, we can conclude that $[y]_R \subseteq [x]_R$, from which we have $[x]_R = [y]_R$.

Assume that $a \in [x]_R$ and $a \in [y]_R$. Consider any $t \in [x]_R$.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$. Since our choice of a was arbitrary, every $a \in A$ belongs to at least one equivalence class – namely, $[a]_R$.

To see that every $a \in A$ belongs to at most one equivalence class, we show that for any $a \in A$, if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R = [y]_R$. To do this, we prove that if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R \subseteq [y]_R$. By interchanging $[x]_R$ and $[y]_R$, we can conclude that $[y]_R \subseteq [x]_R$, from which we have $[x]_R = [y]_R$.

Assume that $a \in [x]_R$ and $a \in [y]_R$. Consider any $t \in [x]_R$. Since $t \in [x]_R$, we know xRt.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$. Since our choice of a was arbitrary, every $a \in A$ belongs to at least one equivalence class – namely, $[a]_R$.

To see that every $a \in A$ belongs to at most one equivalence class, we show that for any $a \in A$, if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R = [y]_R$. To do this, we prove that if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R \subseteq [y]_R$. By interchanging $[x]_R$ and $[y]_R$, we can conclude that $[y]_R \subseteq [x]_R$, from which we have $[x]_R = [y]_R$.

Assume that $a \in [x]_R$ and $a \in [y]_R$. Consider any $t \in [x]_R$. Since $t \in [x]_R$, we know xRt. Since $a \in [x]_R$, we have that xRa.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$. Since our choice of a was arbitrary, every $a \in A$ belongs to at least one equivalence class – namely, $[a]_R$.

To see that every $a \in A$ belongs to at most one equivalence class, we show that for any $a \in A$, if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R = [y]_R$. To do this, we prove that if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R \subseteq [y]_R$. By interchanging $[x]_R$ and $[y]_R$, we can conclude that $[y]_R \subseteq [x]_R$, from which we have $[x]_R = [y]_R$.

Assume that $a \in [x]_R$ and $a \in [y]_R$. Consider any $t \in [x]_R$. Since $t \in [x]_R$, we know xRt. Since $a \in [x]_R$, we have that xRa. Since R is an equivalence relation, R is symmetric and transitive.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$. Since our choice of a was arbitrary, every $a \in A$ belongs to at least one equivalence class – namely, $[a]_R$.

To see that every $a \in A$ belongs to at most one equivalence class, we show that for any $a \in A$, if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R = [y]_R$. To do this, we prove that if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R \subseteq [y]_R$. By interchanging $[x]_R$ and $[y]_R$, we can conclude that $[y]_R \subseteq [x]_R$, from which we have $[x]_R = [y]_R$.

Assume that $a \in [x]_R$ and $a \in [y]_R$. Consider any $t \in [x]_R$. Since $t \in [x]_R$, we know xRt. Since $a \in [x]_R$, we have that xRa. Since R is an equivalence relation, R is symmetric and transitive. By symmetry, from xRa we know aRx.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$. Since our choice of a was arbitrary, every $a \in A$ belongs to at least one equivalence class – namely, $[a]_R$.

To see that every $a \in A$ belongs to at most one equivalence class, we show that for any $a \in A$, if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R = [y]_R$. To do this, we prove that if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R \subseteq [y]_R$. By interchanging $[x]_R$ and $[y]_R$, we can conclude that $[y]_R \subseteq [x]_R$, from which we have $[x]_R = [y]_R$.

Assume that $a \in [x]_R$ and $a \in [y]_R$. Consider any $t \in [x]_R$. Since $t \in [x]_R$, we know xRt. Since $a \in [x]_R$, we have that xRa. Since R is an equivalence relation, R is symmetric and transitive. By symmetry, from xRa we know aRx. By transitivity, from aRx and xRt we know aRt.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$. Since our choice of a was arbitrary, every $a \in A$ belongs to at least one equivalence class – namely, $[a]_R$.

To see that every $a \in A$ belongs to at most one equivalence class, we show that for any $a \in A$, if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R = [y]_R$. To do this, we prove that if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R \subseteq [y]_R$. By interchanging $[x]_R$ and $[y]_R$, we can conclude that $[y]_R \subseteq [x]_R$, from which we have $[x]_R = [y]_R$.

Assume that $a \in [x]_R$ and $a \in [y]_R$. Consider any $t \in [x]_R$. Since $t \in [x]_R$, we know xRt. Since $a \in [x]_R$, we have that xRa. Since R is an equivalence relation, R is symmetric and transitive. By symmetry, from xRa we know aRx. By transitivity, from aRx and xRt we know aRt. Since $a \in [y]_R$, we also know yRa.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$. Since our choice of a was arbitrary, every $a \in A$ belongs to at least one equivalence class – namely, $[a]_R$.

To see that every $a \in A$ belongs to at most one equivalence class, we show that for any $a \in A$, if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R = [y]_R$. To do this, we prove that if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R \subseteq [y]_R$. By interchanging $[x]_R$ and $[y]_R$, we can conclude that $[y]_R \subseteq [x]_R$, from which we have $[x]_R = [y]_R$.

Assume that $a \in [x]_R$ and $a \in [y]_R$. Consider any $t \in [x]_R$. Since $t \in [x]_R$, we know xRt. Since $a \in [x]_R$, we have that xRa. Since R is an equivalence relation, R is symmetric and transitive. By symmetry, from xRa we know aRx. By transitivity, from aRx and xRt we know aRt. Since $a \in [y]_R$, we also know yRa. By transitivity, from yRa and aRt we know yRt.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$. Since our choice of a was arbitrary, every $a \in A$ belongs to at least one equivalence class – namely, $[a]_R$.

To see that every $a \in A$ belongs to at most one equivalence class, we show that for any $a \in A$, if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R = [y]_R$. To do this, we prove that if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R \subseteq [y]_R$. By interchanging $[x]_R$ and $[y]_R$, we can conclude that $[y]_R \subseteq [x]_R$, from which we have $[x]_R = [y]_R$.

Assume that $a \in [x]_R$ and $a \in [y]_R$. Consider any $t \in [x]_R$. Since $t \in [x]_R$, we know xRt. Since $a \in [x]_R$, we have that xRa. Since R is an equivalence relation, R is symmetric and transitive. By symmetry, from xRa we know aRx. By transitivity, from aRx and xRt we know aRt. Since $a \in [y]_R$, we also know yRa. By transitivity, from yRa and aRt we know yRt. Thus, $t \in [y]_R$.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$. Since our choice of a was arbitrary, every $a \in A$ belongs to at least one equivalence class – namely, $[a]_R$.

To see that every $a \in A$ belongs to at most one equivalence class, we show that for any $a \in A$, if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R = [y]_R$. To do this, we prove that if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R \subseteq [y]_R$. By interchanging $[x]_R$ and $[y]_R$, we can conclude that $[y]_R \subseteq [x]_R$, from which we have $[x]_R = [y]_R$.

Assume that $a \in [x]_R$ and $a \in [y]_R$. Consider any $t \in [x]_R$. Since $t \in [x]_R$, we know xRt. Since $a \in [x]_R$, we have that xRa. Since R is an equivalence relation, R is symmetric and transitive. By symmetry, from xRa we know aRx. By transitivity, from aRx and xRt we know aRt. Since $a \in [y]_R$, we also know yRa. By transitivity, from yRa and aRt we know yRt. Thus, $t \in [y]_R$. Since our choice of t was arbitrary, $[x]_R \subseteq [y]_R$.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$. Since our choice of a was arbitrary, every $a \in A$ belongs to at least one equivalence class – namely, $[a]_R$.

To see that every $a \in A$ belongs to at most one equivalence class, we show that for any $a \in A$, if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R = [y]_R$. To do this, we prove that if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R \subseteq [y]_R$. By interchanging $[x]_R$ and $[y]_R$, we can conclude that $[y]_R \subseteq [x]_R$, from which we have $[x]_R = [y]_R$.

Assume that $a \in [x]_R$ and $a \in [y]_R$. Consider any $t \in [x]_R$. Since $t \in [x]_R$, we know xRt. Since $a \in [x]_R$, we have that xRa. Since R is an equivalence relation, R is symmetric and transitive. By symmetry, from xRa we know aRx. By transitivity, from aRx and xRt we know aRt. Since $a \in [y]_R$, we also know yRa. By transitivity, from yRa and aRt we know yRt. Thus, $t \in [y]_R$. Since our choice of t was arbitrary, $[x]_R \subseteq [y]_R$. By our above reasoning, $[x]_R = [y]_R$.

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$. Since our choice of a was arbitrary, every $a \in A$ belongs to at least one equivalence class – namely, $[a]_R$.

To see that every $a \in A$ belongs to at most one equivalence class, we show that for any $a \in A$, if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R = [y]_R$. To do this, we prove that if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R \subseteq [y]_R$. By interchanging $[x]_R$ and $[y]_R$, we can conclude that $[y]_R \subseteq [x]_R$, from which we have $[x]_R = [y]_R$.

Assume that $a \in [x]_R$ and $a \in [y]_R$. Consider any $t \in [x]_R$. Since $t \in [x]_R$, we know xRt. Since $a \in [x]_R$, we have that xRa. Since R is an equivalence relation, R is symmetric and transitive. By symmetry, from xRa we know aRx. By transitivity, from aRx and xRt we know aRt. Since $a \in [y]_R$, we also know yRa. By transitivity, from yRa and aRt we know yRt. Thus, $t \in [y]_R$. Since our choice of t was arbitrary, $[x]_R \subseteq [y]_R$. By our above reasoning, $[x]_R = [y]_R$. ■

Theorem: Let R be an equivalence relation over a set A. Then every element of A belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of A belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since R is an equivalence relation, R is reflexive, so aRa. Thus $a \in [a]_R$. Since our choice of a was arbitrary, every $a \in A$ belongs to at least one equivalence class – namely, $[a]_R$.

To see that every $a \in A$ belongs to at most one equivalence class, we show that for any $a \in A$, if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R = [y]_R$. To do this, we prove that if $a \in [x]_R$ and $a \in [y]_R$, then $[x]_R \subseteq [y]_R$. By interchanging $[x]_R$ and $[y]_R$, we can conclude that $[y]_R \subseteq [x]_R$, from which we have $[x]_R = [y]_R$.

Assume that $a \in [x]_R$ and $a \in [y]_R$. Consider any $t \in [x]_R$. Since $t \in [x]_R$, we know xRt. Since $a \in [x]_R$, we have that xRa. Since R is an equivalence relation, R is symmetric and transitive. By symmetry, from xRa we know aRx. By transitivity, from aRx and xRt we know aRt. Since $a \in [y]_R$, we also know yRa. By transitivity, from yRa and aRt we know yRt. Thus, $t \in [y]_R$. Since our choice of t was arbitrary, $[x]_R \subseteq [y]_R$. By our above reasoning, $[x]_R = [y]_R$. ■

Theorem: Let *R* be an equivalence relation over a set *A*. Then every element of *A* belongs to exactly one equivalence class.

Proof: We need to show that every $a \in A$ belongs to at least one equivalence class and to at most one equivalence class.

To see that every element of *A* belongs to at least one equivalence class, consider any $a \in A$ and the equivalence class $[a]_R = \{ x \mid x \in A \text{ and } aRx \}$. Since *R* is an equivalence relation, *R* is reflexive, so *aRa*. Thus $a \in [a]_R$. Since our choice of *a* was arbitrary, every $a \in A$ belongs to at least one equiv

To se … ow that f … prove … and $[y]_R$, w

Assume that $a \in [x]_R$ and $a \in [y]_R$. Consider any $t \in [x]_R$. Since $t \in [x]_R$, we know *xRt*. Since $a \in [x]_R$, we have that *xRa*. Since *R* is an equivalence relation, *R* is symmetric and transitive. By symmetry, from *xRa* we know *aRx*. By transitivity, from *aRx* and *xRt* we know *aRt*. Since $a \in [y]_R$, we also know *yRa*. By transitivity, from *yRa* and *aRt* we know *yRt*. Thus, $t \in [y]_R$. Since our choice of *t* was arbitrary, $[x]_R \subseteq [y]_R$. By our above reasoning, $[x]_R = [y]_R$. ■

This proof helps to justify our definition of equivalence relations. We need all three of the properties we've listed in order for this proof to work, and we don't need any others.

Order Relations

“x is larger than y”

“x is tastier than y”

“x runs faster than y”

“x is a subset of y”

“x divides y”

“x is a part of y”

Informally

An **order relation** is a relation that ranks elements against one another.

Again, do <u>not</u> use this definition in proofs! It's just an intuition!

Properties of Order Relations

$$x \le y$$

Properties of Order Relations

$x \leq y$

$1 \leq 5$ **and** $5 \leq 8$

Properties of Order Relations

$x \leq y$

$1 \leq 5$ **and** $5 \leq 8$

$1 \leq 8$

Properties of Order Relations

$x \leq y$

$42 \leq 99$ **and** $99 \leq 137$

Properties of Order Relations

$x \leq y$

$42 \leq 99$ **and** $99 \leq 137$

$42 \leq 137$

Properties of Order Relations

$x \leq y$

$x \leq y$ **and** $y \leq z$

Properties of Order Relations

$x \leq y$

$x \leq y$ **and** $y \leq z$

$x \leq z$

Properties of Order Relations

$x \leq y$

$x \leq y$ **and** $y \leq z$

$x \leq z$

Transitivity

Properties of Order Relations

$x \leq y$

Properties of Order Relations

$x \leq y$

$$1 \leq 1$$

Properties of Order Relations

$x \leq y$

$$42 \leq 42$$

Properties of Order Relations

$x \leq y$

$$137 \leq 137$$

Properties of Order Relations

$x \leq y$

$$x \leq x$$

Properties of Order Relations

$x \leq y$

$$x \leq x$$

Reflexivity

Properties of Order Relations

$$x \leq y$$

Properties of Order Relations

$x \le y$

$19 \le 21$

Properties of Order Relations

$x \leq y$

$19 \leq 21$

$21 \leq 19$?

Properties of Order Relations

$x \leq y$

$19 \leq 21$

~~$21 \leq 19$~~?

Properties of Order Relations

$x \leq y$

$$42 \leq 137$$

Properties of Order Relations

$x \leq y$

$42 \leq 137$

$137 \leq 42$?

Properties of Order Relations

$x \leq y$

$42 \leq 137$

~~$137 \leq 42$~~?

Properties of Order Relations

$x \leq y$

$137 \leq 137$

Properties of Order Relations

$x \leq y$

$137 \leq 137$

$137 \leq 137$?

Properties of Order Relations

$x \leq y$

$137 \leq 137$

$\mathbf{137 \leq 137}$

Antisymmetry

A binary relation *R* over a set *A* is called **antisymmetric** iff

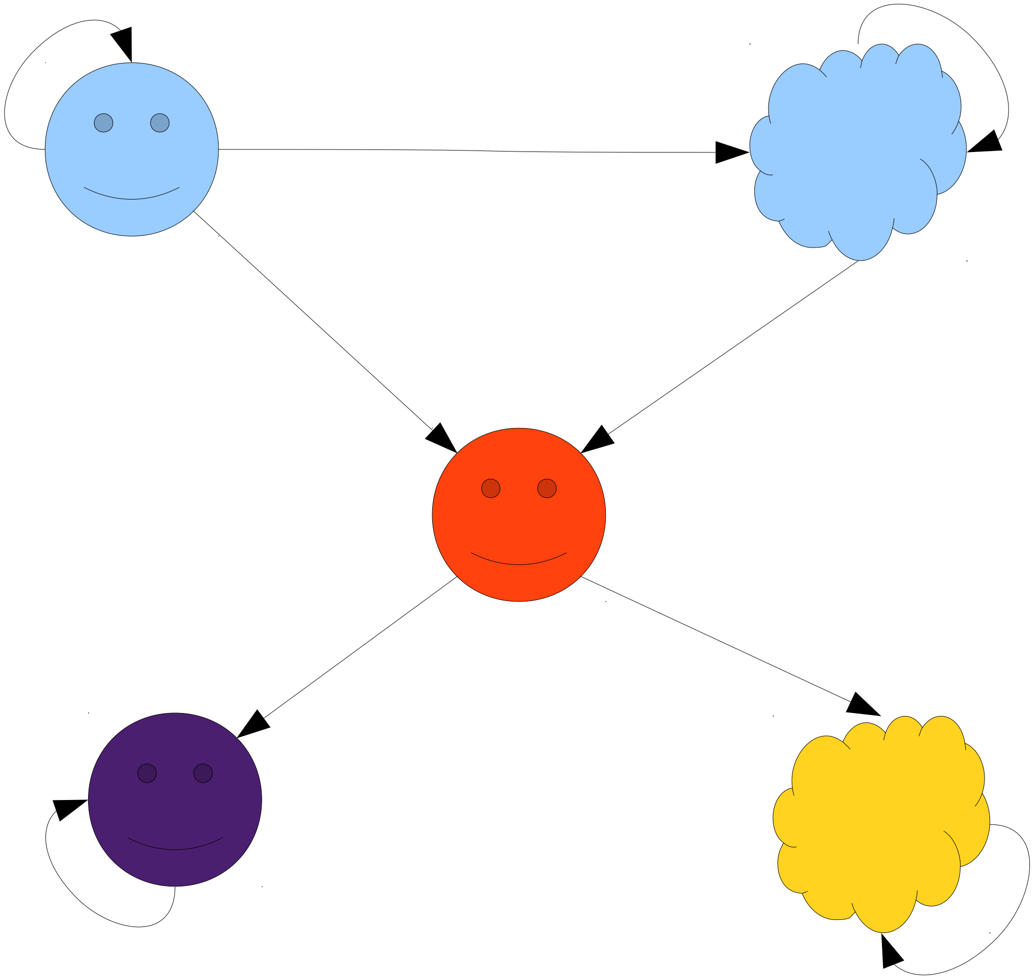
For any $x \in A$ and $y \in A$,
if xRy and yRx, then x = y.

Equivalently:

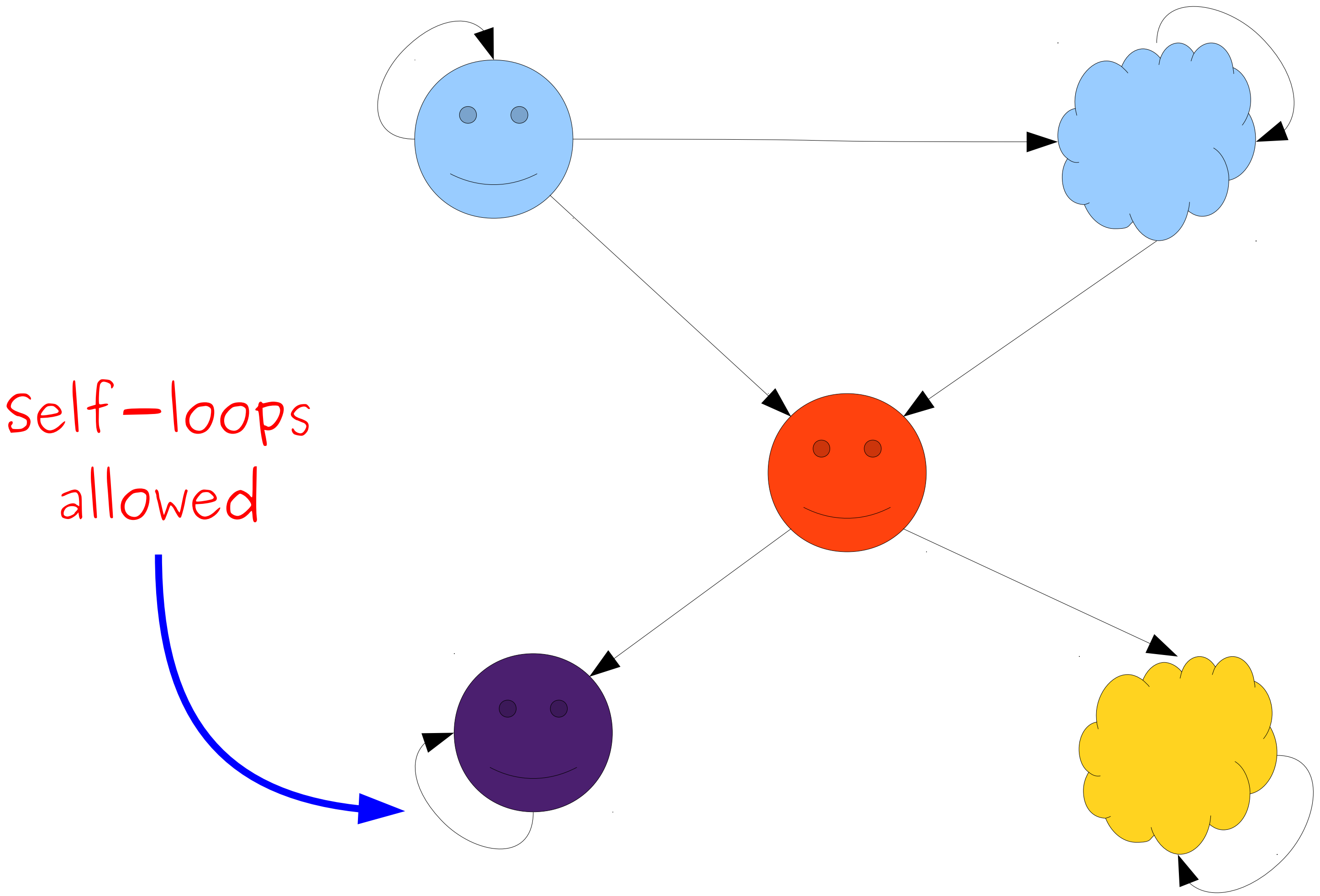
For any $x \in A$ and $y \in A$,
If xRy and $y \neq x$, then $y \not{R} x$.

An Intuition for Antisymmetry

An Intuition for Antisymmetry

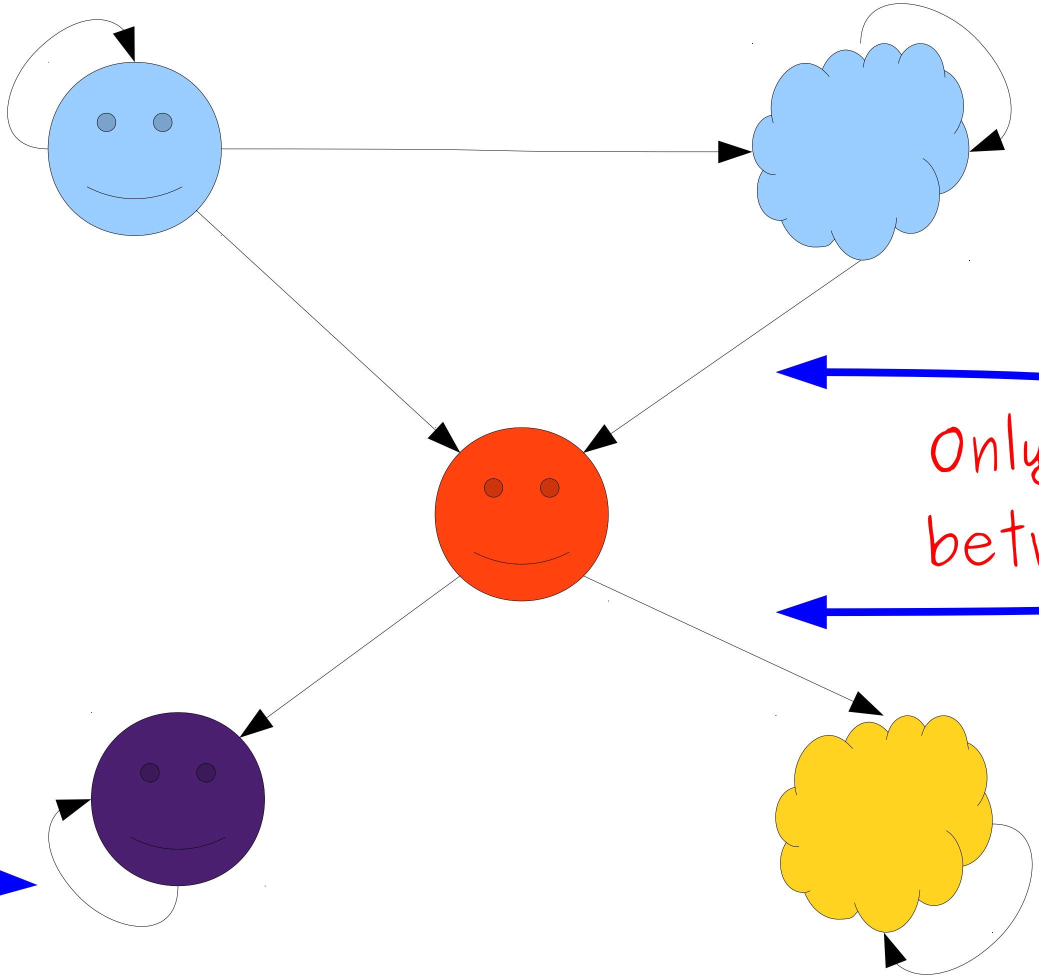


An Intuition for Antisymmetry



Self-loops allowed

An Intuition for Antisymmetry



Self-loops allowed

Only one edge between nodes

An Important Detail

- A binary relation R over a set A is antisymmetric iff for any $x \in A$ and $y \in A$, if $x\mathrm{R}y$ and $y\mathrm{R}x$, then $x = y$.
- Is the relation < over real numbers antisymmetric?

An Important Detail

- A binary relation R over a set A is antisymmetric iff for any $x \in A$ and $y \in A$, if $x\mathrm{R}y$ and $y\mathrm{R}x$, then $x = y$.
- Is the relation < over real numbers antisymmetric?
- **Yes**: This is vacuously true.
  - It's never possible for $x < y$ and $y < x$ to be true simultaneously.
  - The claim "if $x\mathrm{R}y$ and $y\mathrm{R}x$, then $x = y$" is thus vacuously true.

Partial Orders

- A binary relation *R* is a **partial order** if it is
  - **reflexive**,
  - **antisymmetric**, and
  - **transitive**.
- A pair (*S*, *R*), where *R* is a partial order over *S*, is called a **partially ordered set** or **poset**.

Partial Orders

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

Why "partial"?

- A pair (*S*, *R*), where *R* is a partial order over *S*, is called a **partially ordered set** or **poset**.

2008 Summer Olympics

Gold	Silver	Bronze	Total
51	21	28	100
36	38	36	110
23	21	28	72
19	13	15	47
14	15	17	46

Inspired by http://tartarus.org/simon/2008-olympics-hasse/.
Data from http://news.bbc.co.uk/sport2/hi/olympics/medals_table/default.stm

2008 Summer Olympics

Gold	Silver	Bronze	Total
51	21	28	100
36	38	36	110
23	21	28	72
19	13	15	47
14	15	17	46

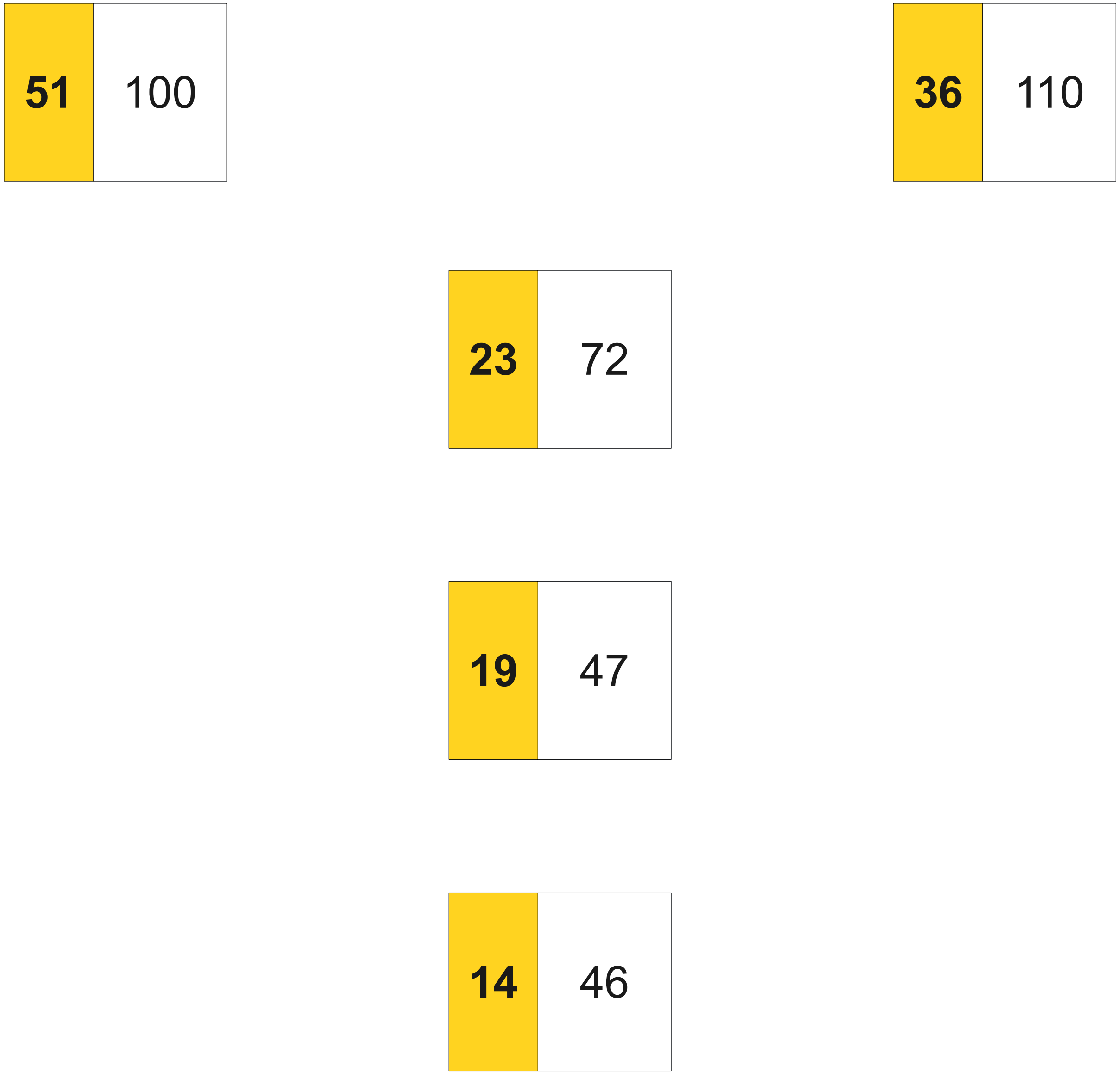
Inspired by http://tartarus.org/simon/2008-olympics-hasse/.
Data from http://news.bbc.co.uk/sport2/hi/olympics/medals_table/default.stm

Define the relationship

$$(\mathbf{gold_0}, \mathbf{total_0})\mathit{R}(\mathbf{gold_1}, \mathbf{total_1})$$

to be true when

$$\mathbf{gold_0} \le \mathbf{gold_1} \text{ and } \mathbf{total_0} \le \mathbf{total_1}$$



51 100

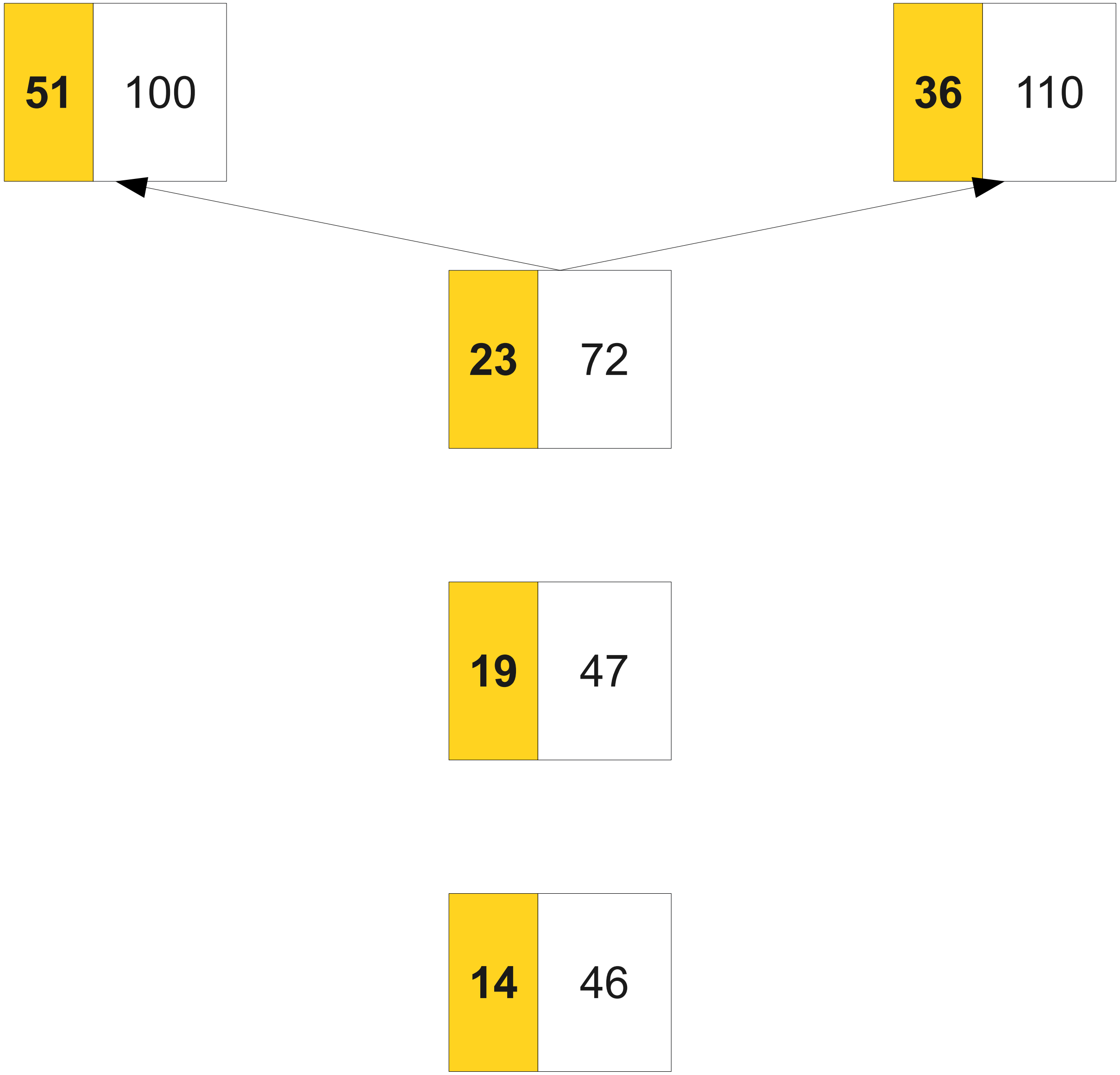
36 110

23 72

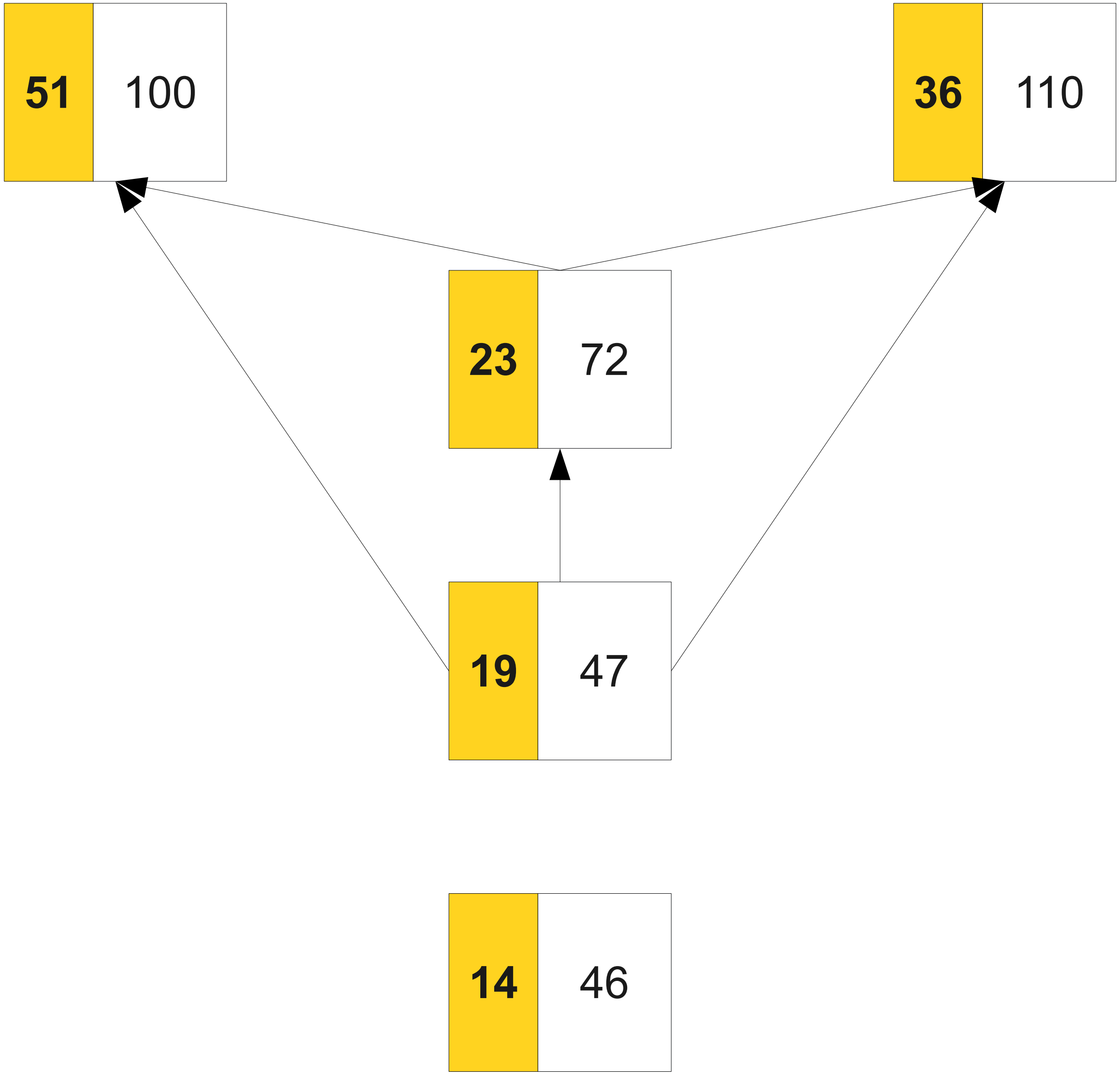
19 47

14 46

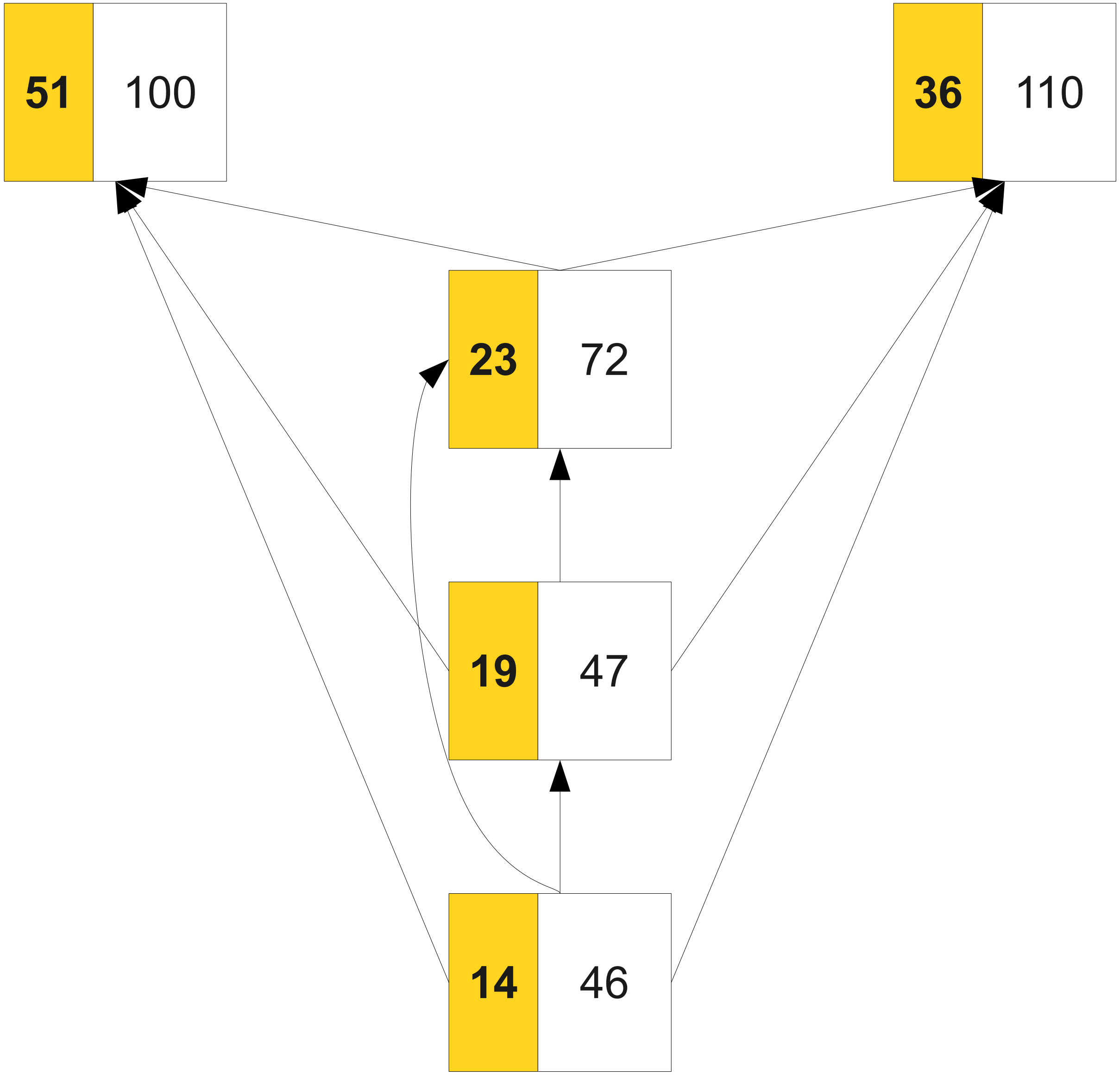
$(g, t)R(g', t') \equiv g \leq g'$ and $t \leq t'$



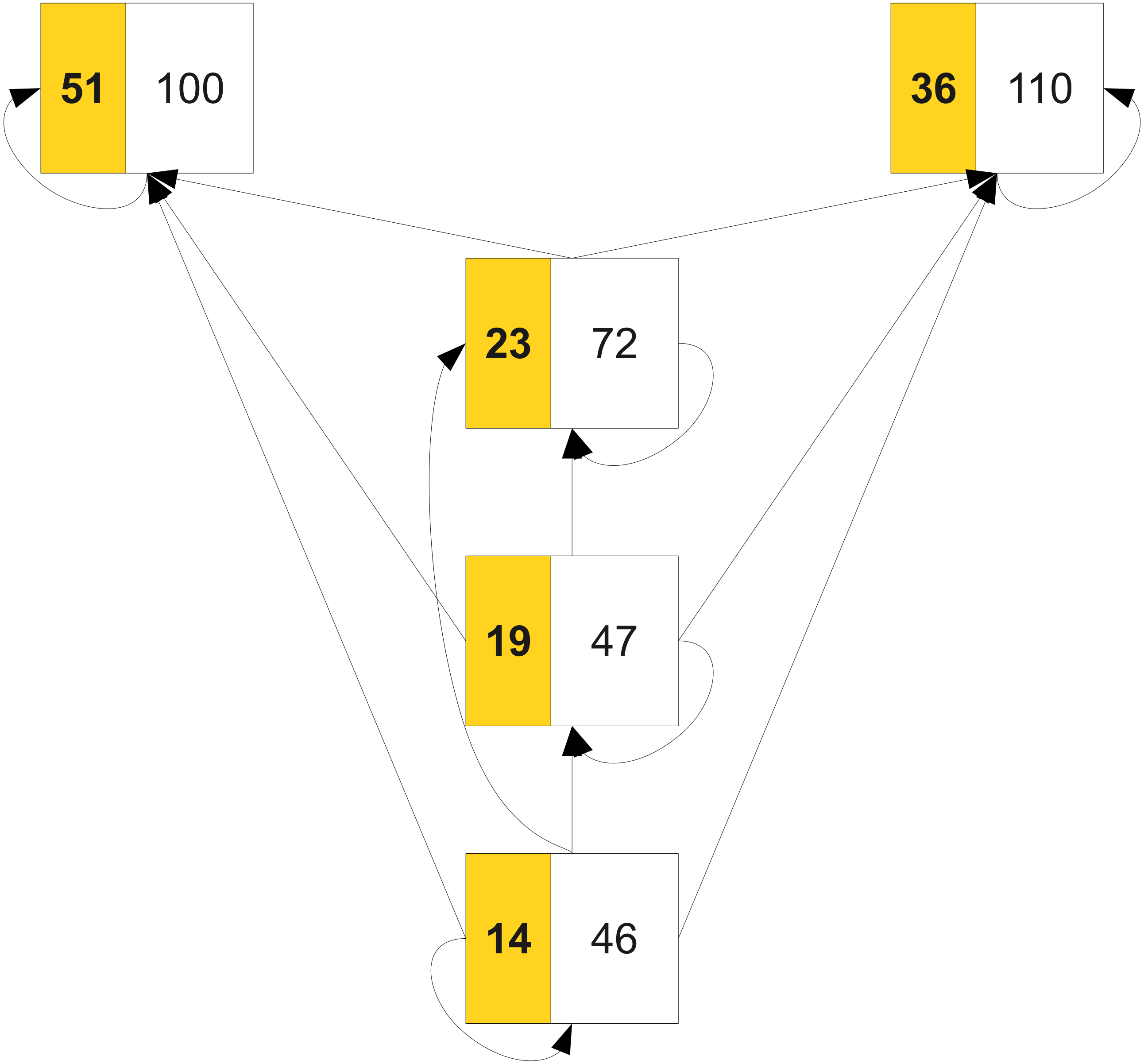
$$(g, t)R(g', t') \equiv g \leq g' \text{ and } t \leq t'$$



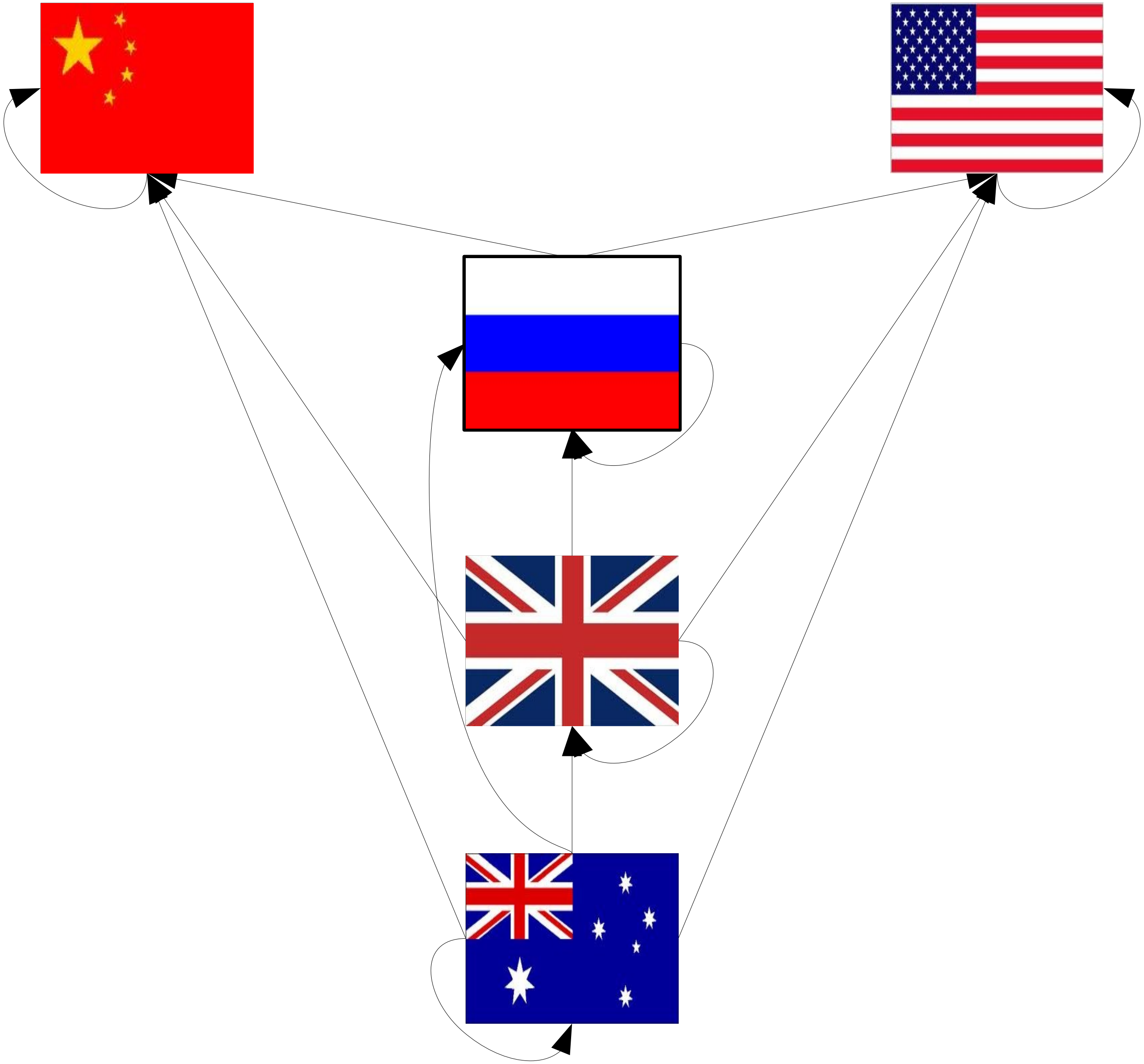
(g, t)R(g', t') ≡ g ≤ g' and t ≤ t'



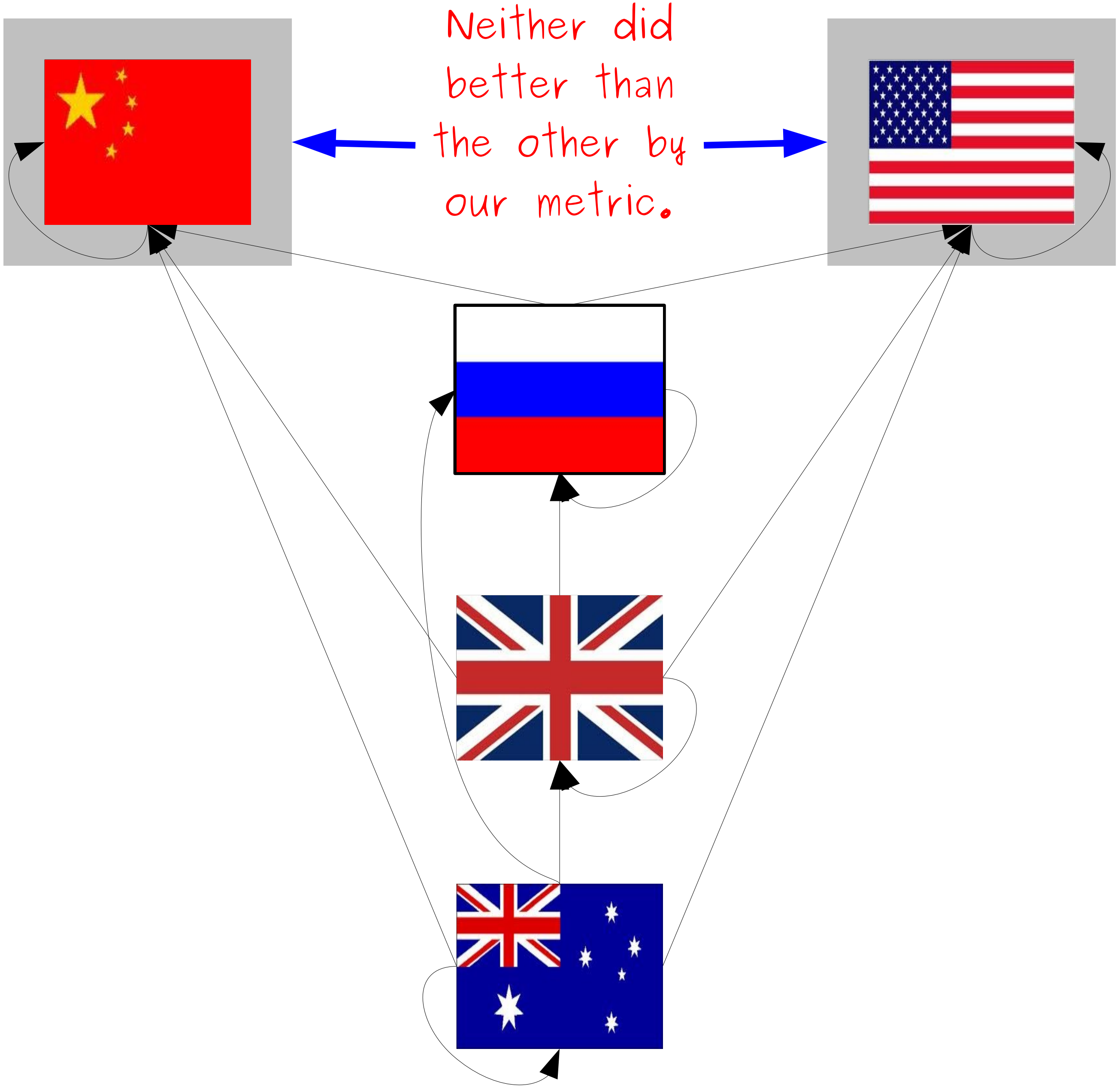
$$(g, t)R(g', t') \equiv g \leq g' \text{ and } t \leq t'$$



(g, t)R(g', t') ≡ g ≤ g' and t ≤ t'



$(g, t)R(g', t') \equiv g \leq g' \text{ and } t \leq t'$

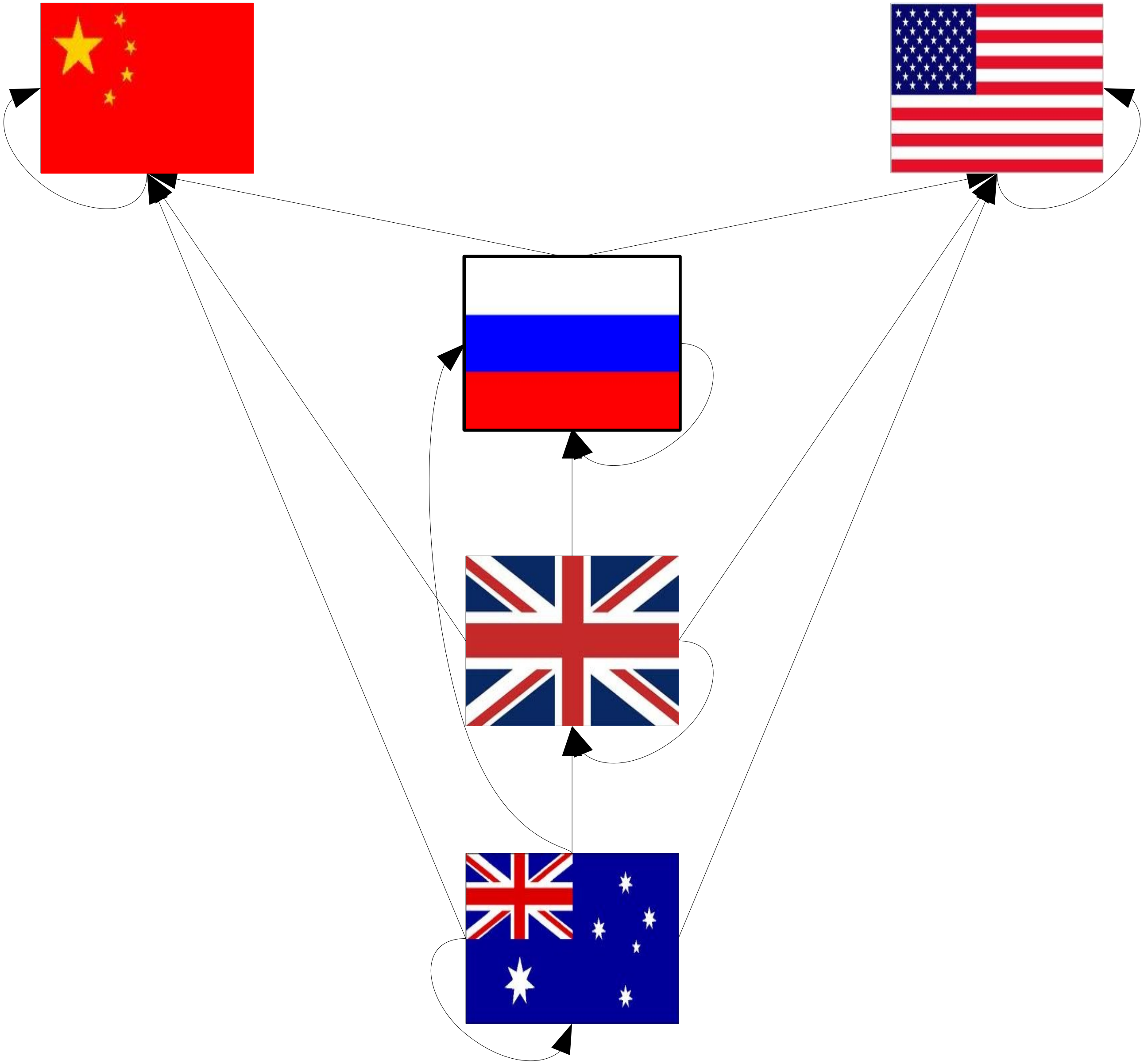


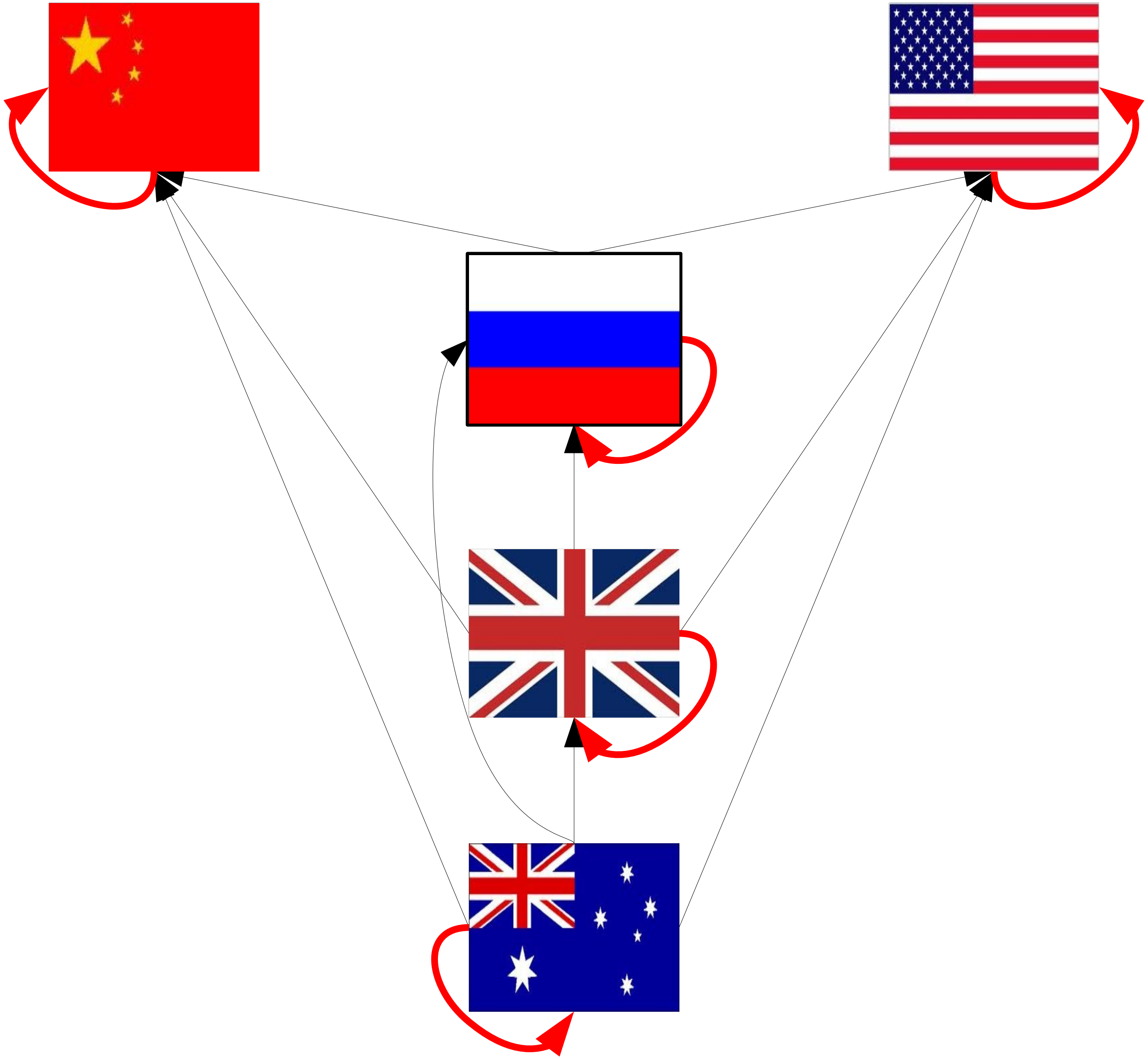
Neither did better than the other by our metric.

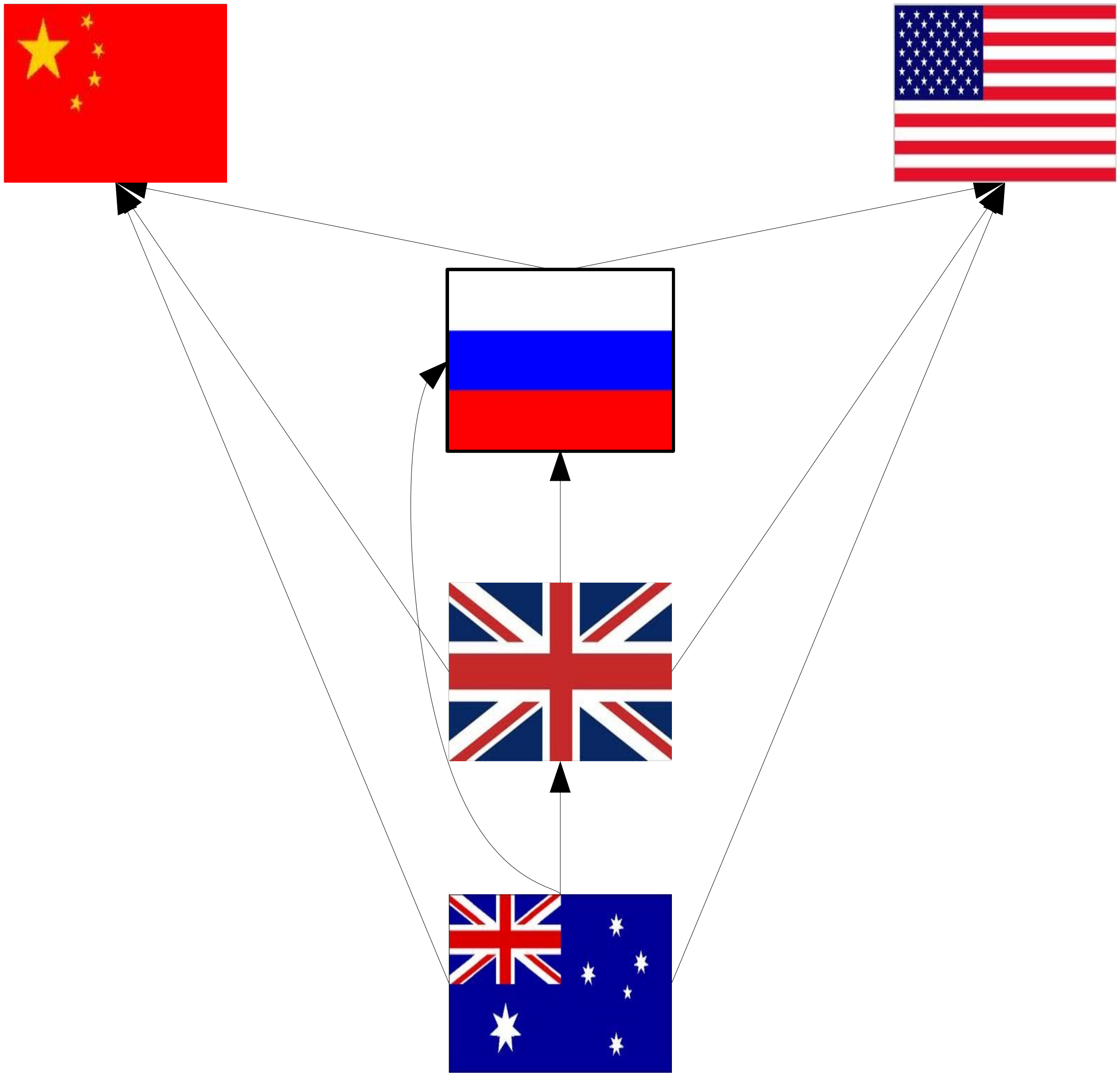
$$(g, t)R(g', t') \equiv g \leq g' \text{ and } t \leq t'$$

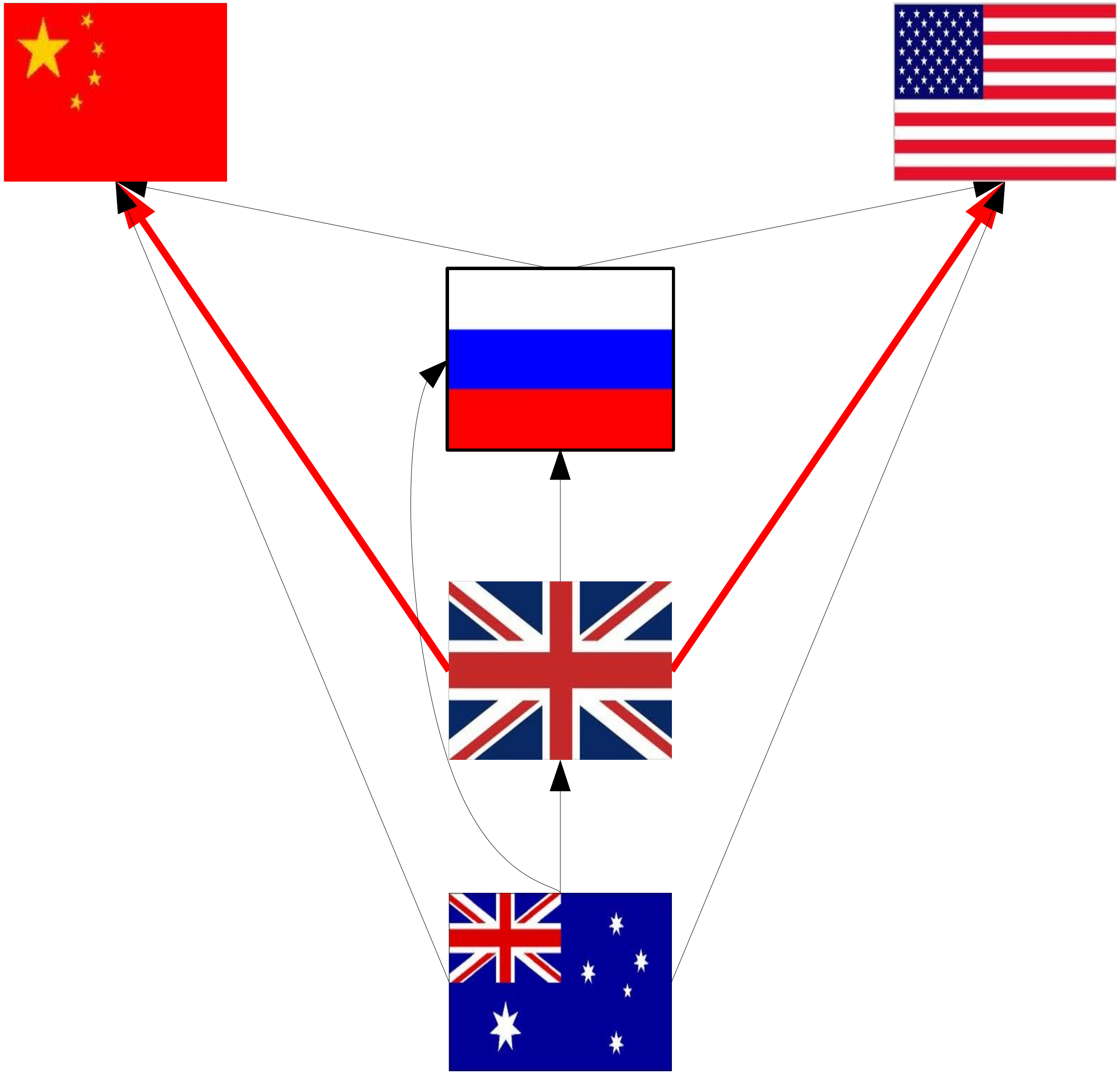
Partial and Total Orders

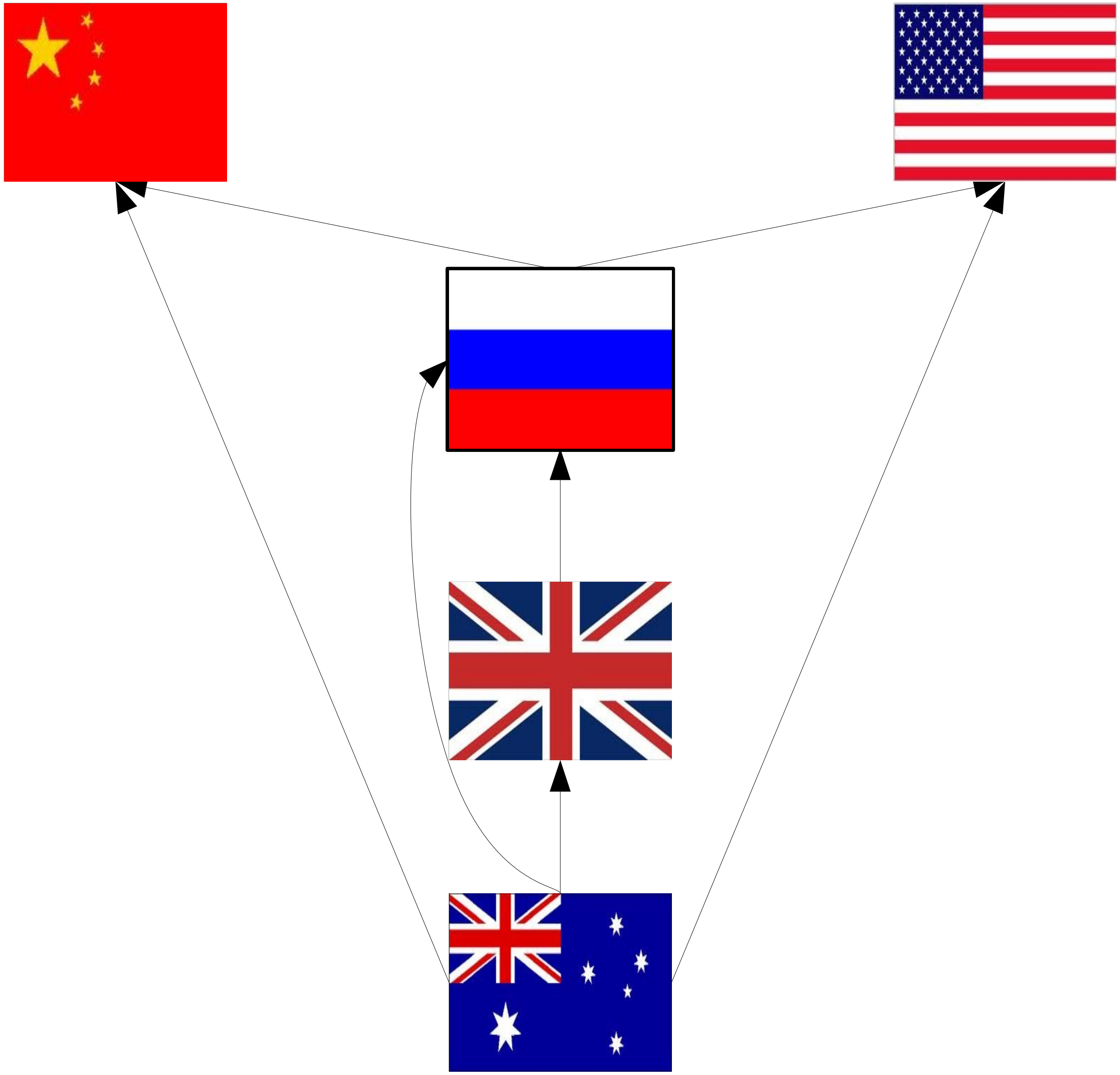
- A relation R over a set A is called **total** iff for any $x \in A$ and $y \in A$, either xRy or yRx.
    - Could both be true?
- A **partial order** is called a **total order** if it is total.
- Examples:
    - Integers ordered by ≤.
    - Strings ordered alphabetically.

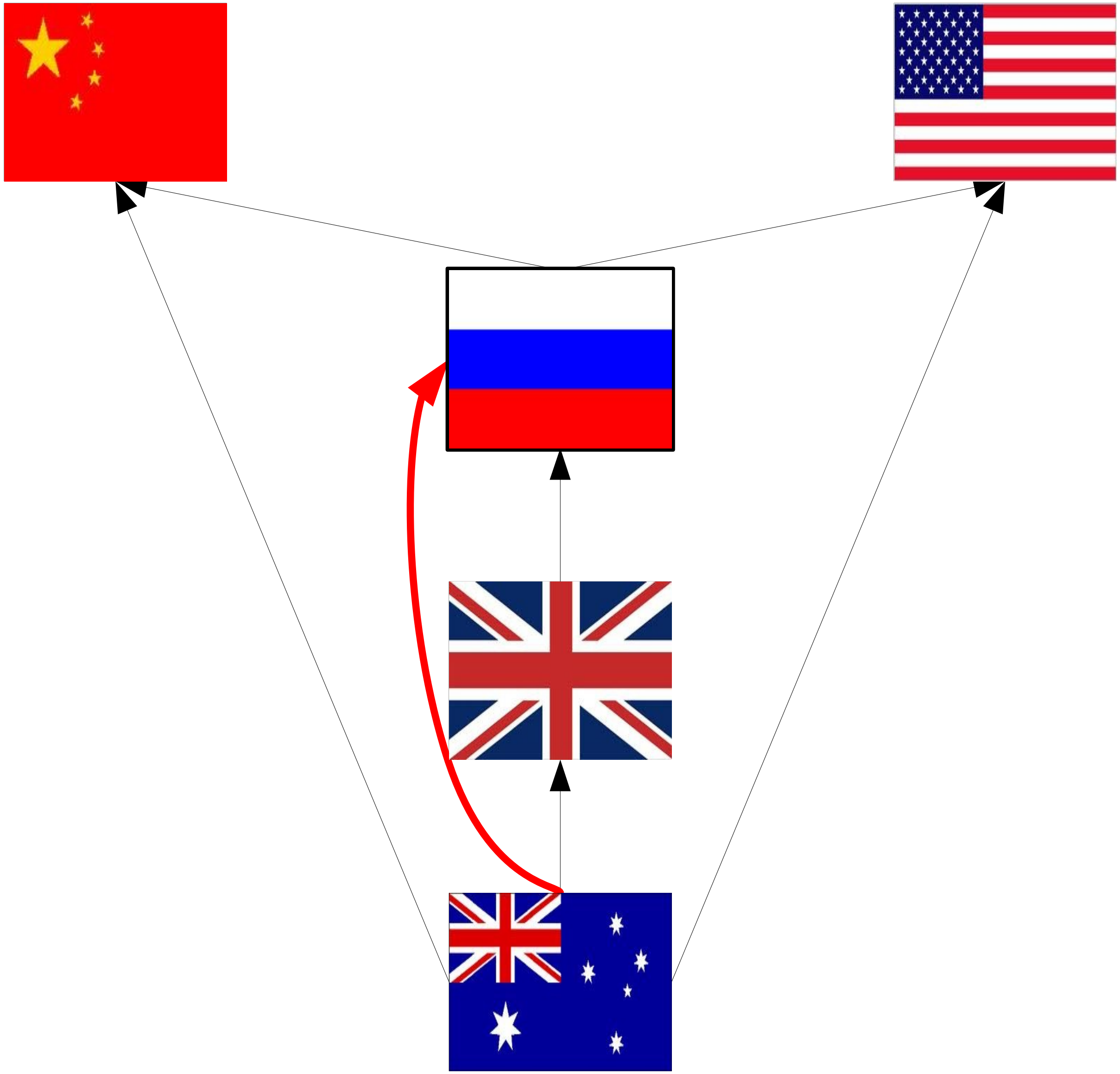


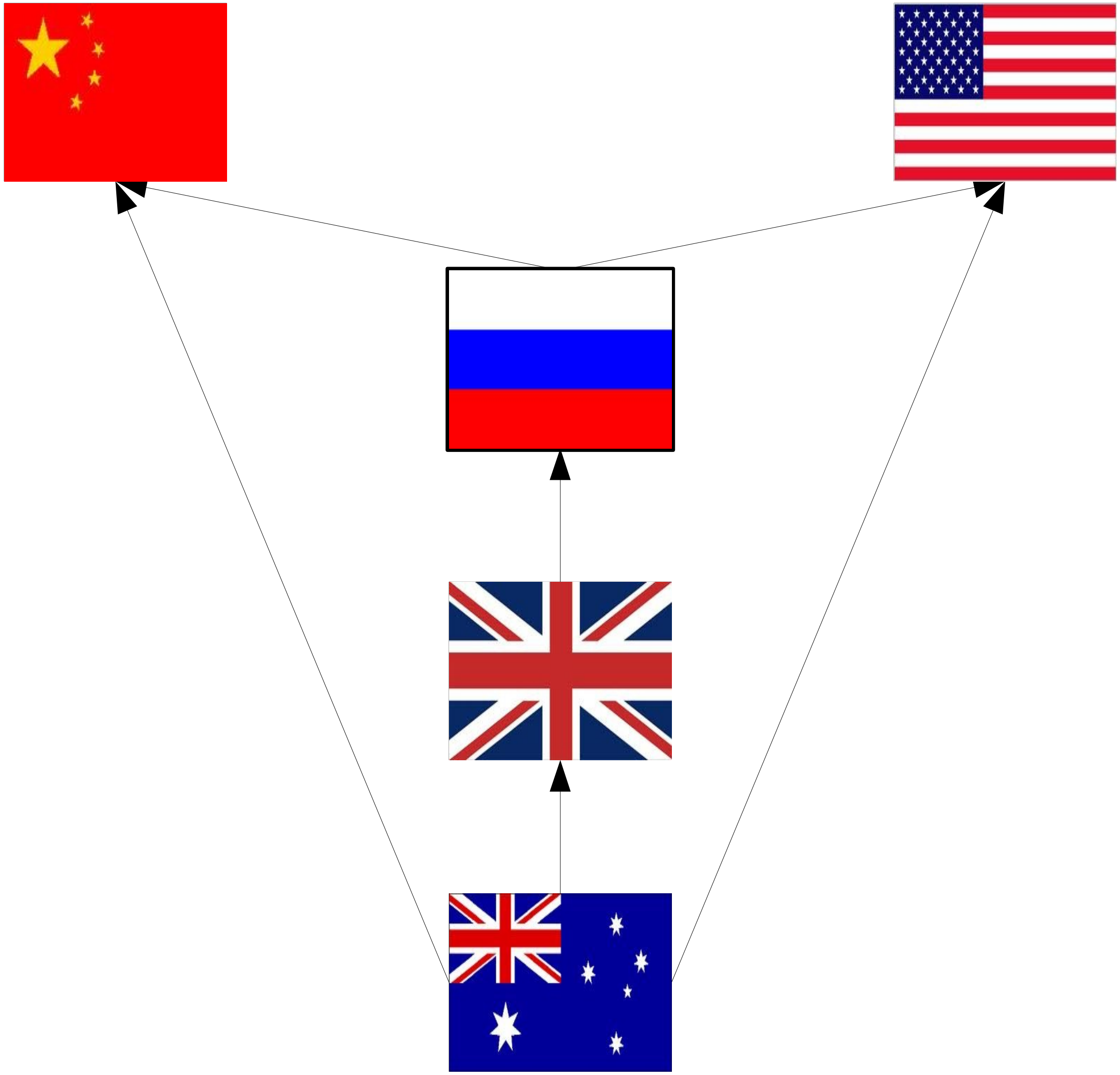


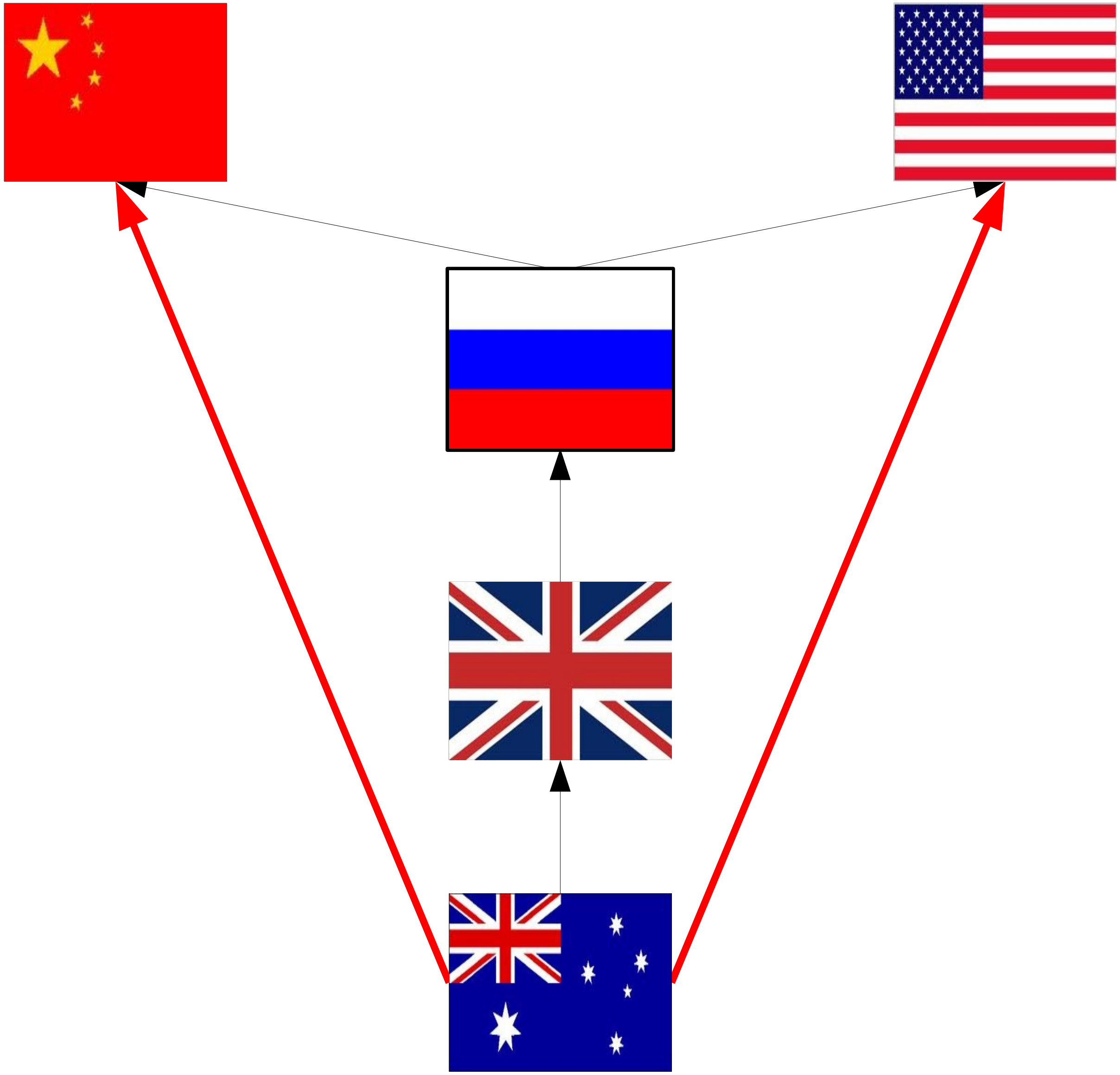


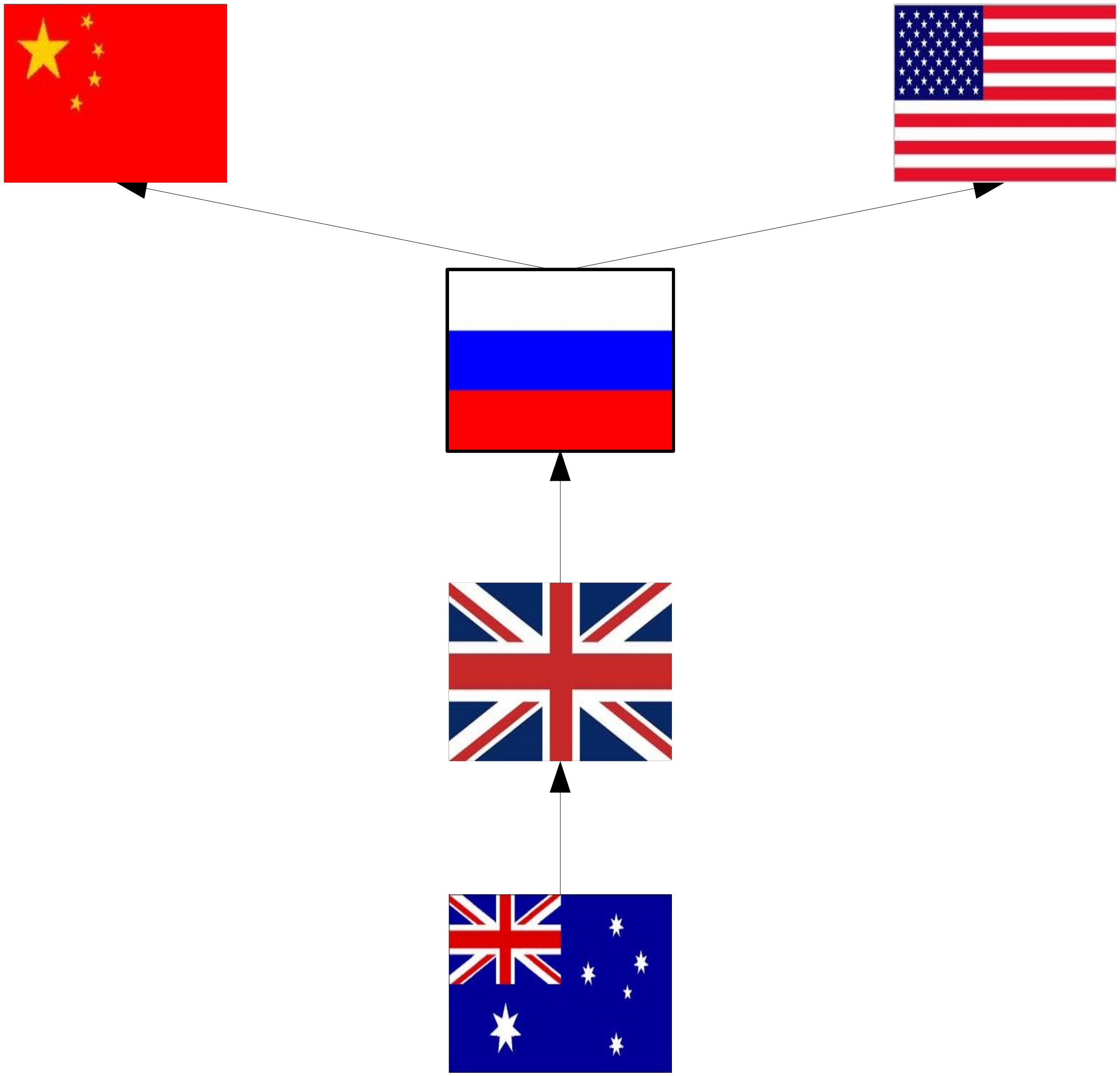


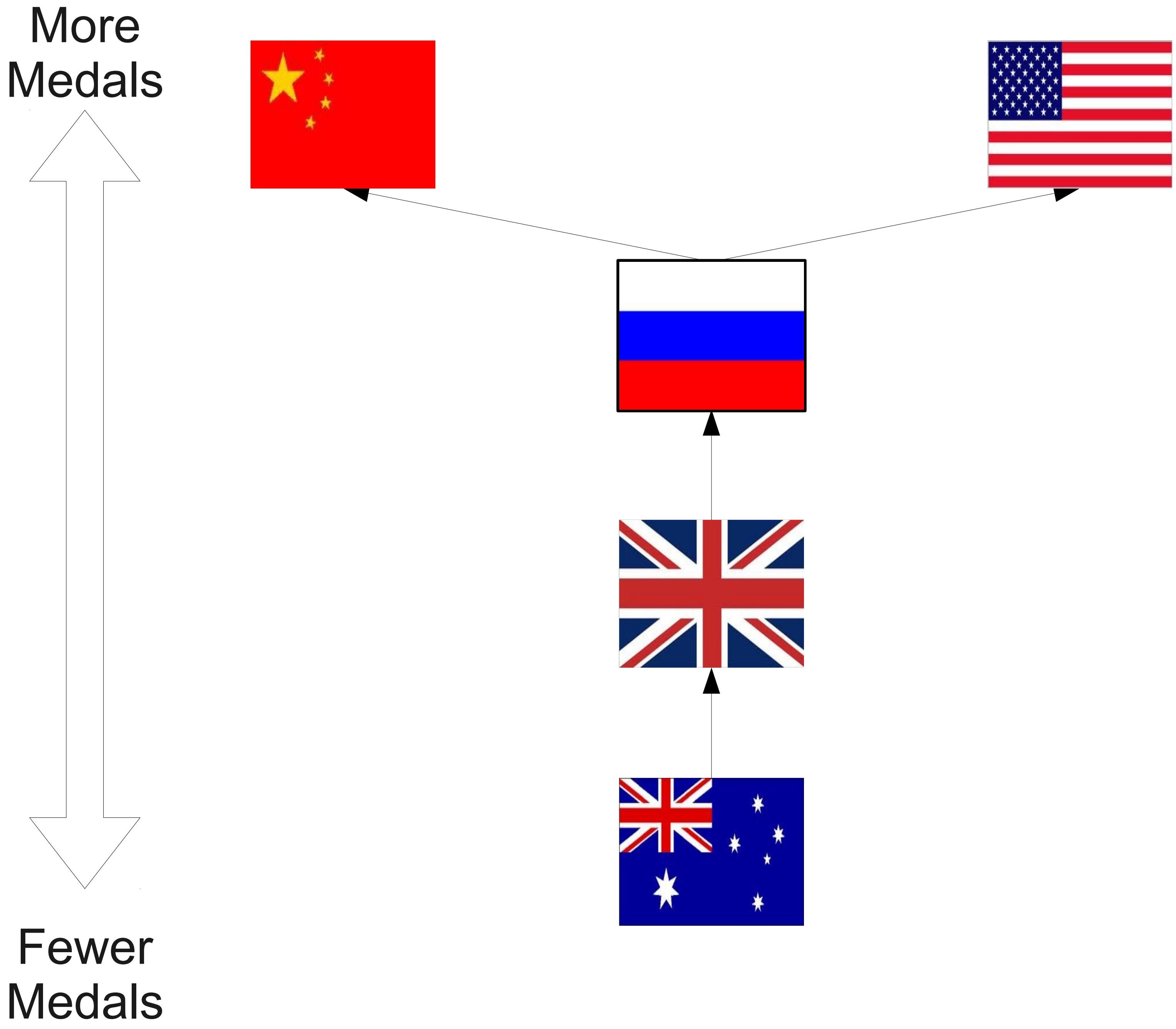






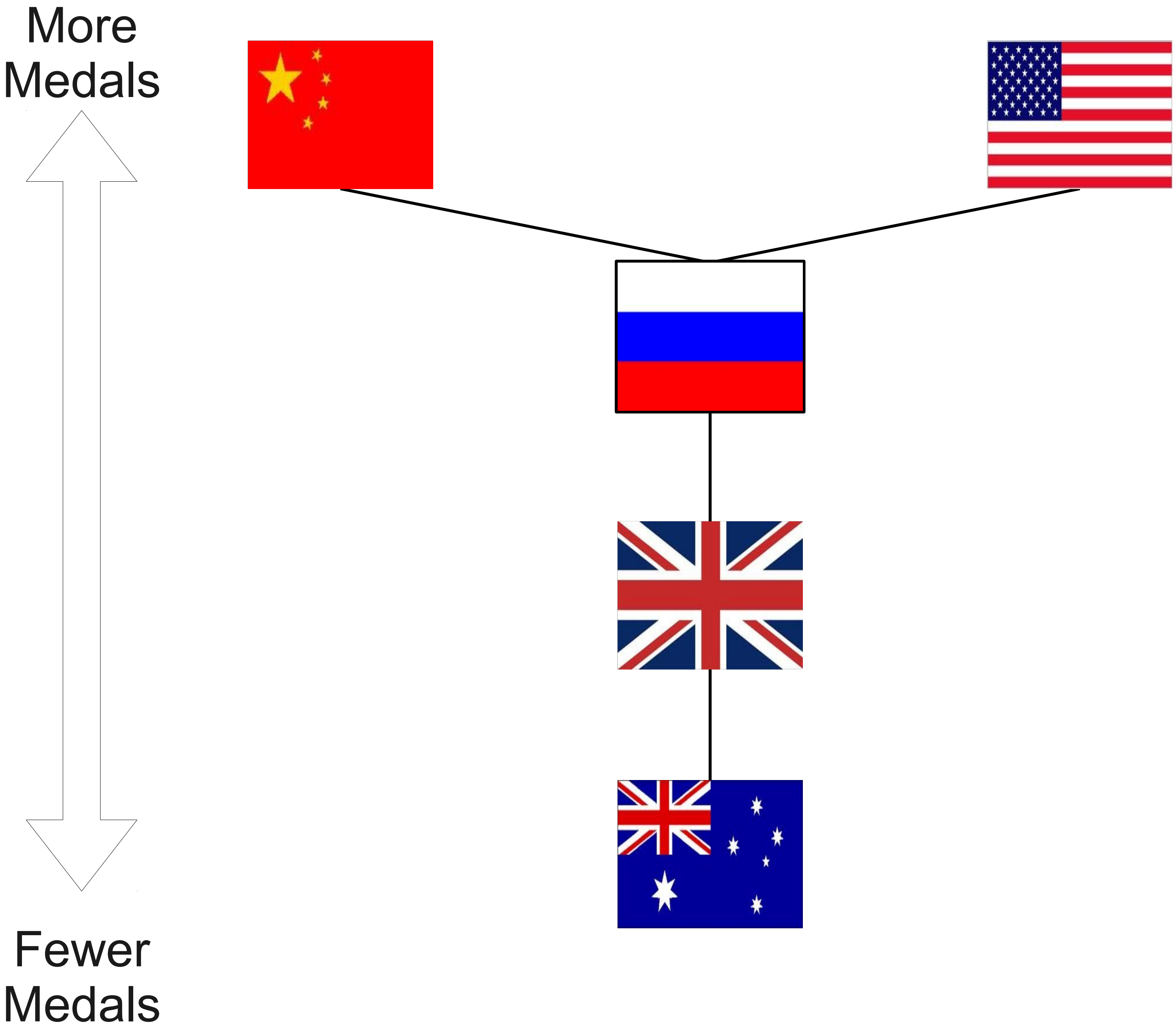






More Medals

Fewer Medals



More Medals

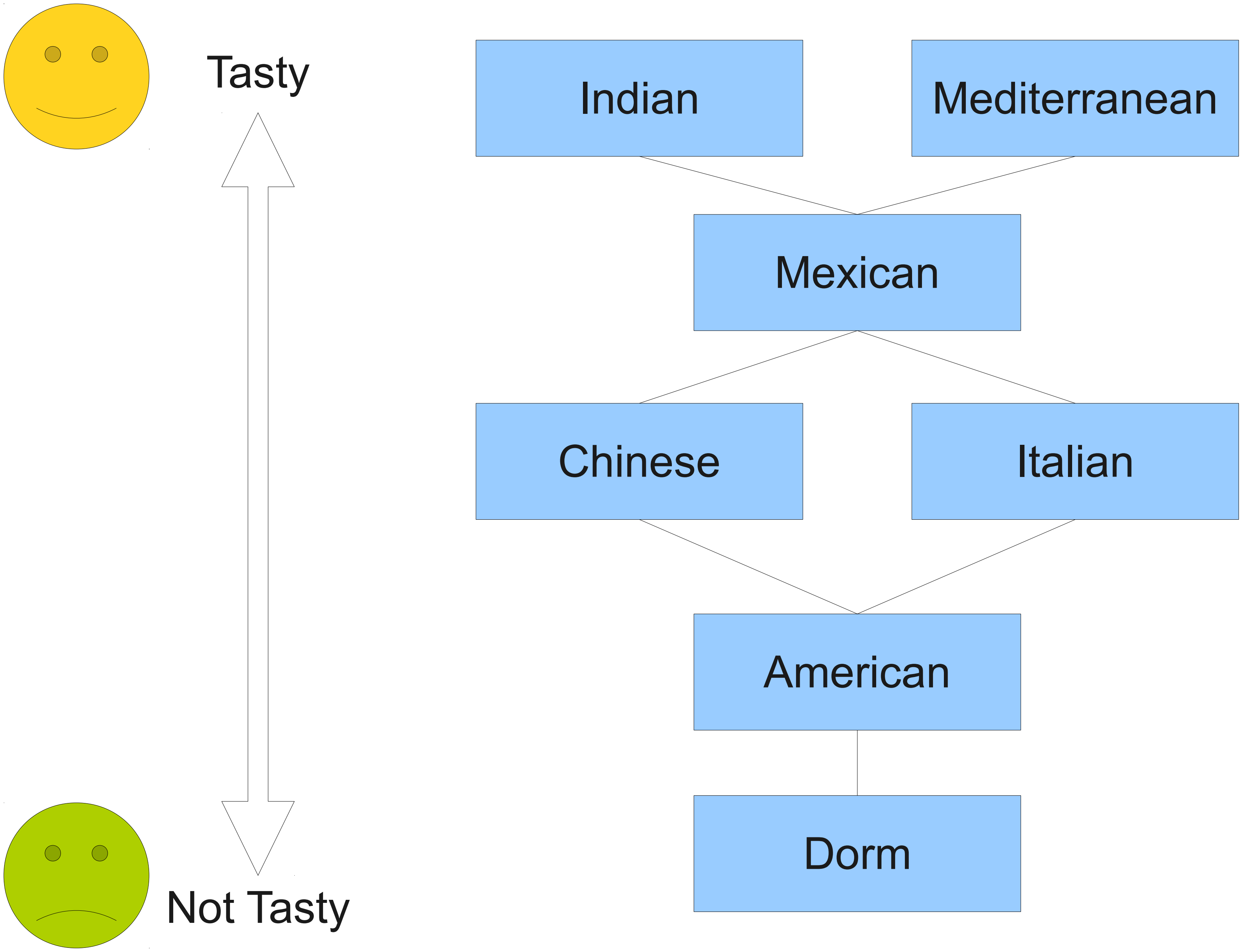
Fewer Medals

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.

This is a good justification for our definition! These drawings encode the structure we'd like, and the three properties we've picked guarantee us that they mean what we want them to mean.

- 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

Indian

Mediterranean

Mexican

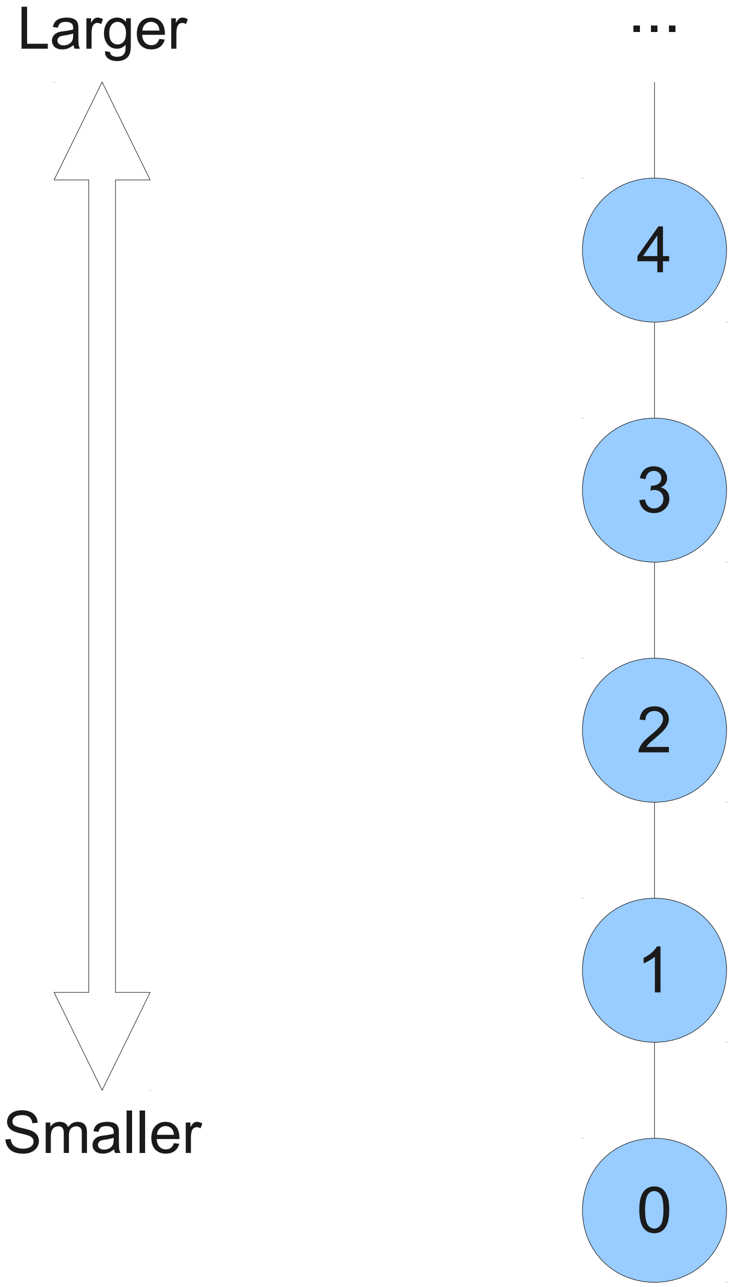
Chinese

Italian

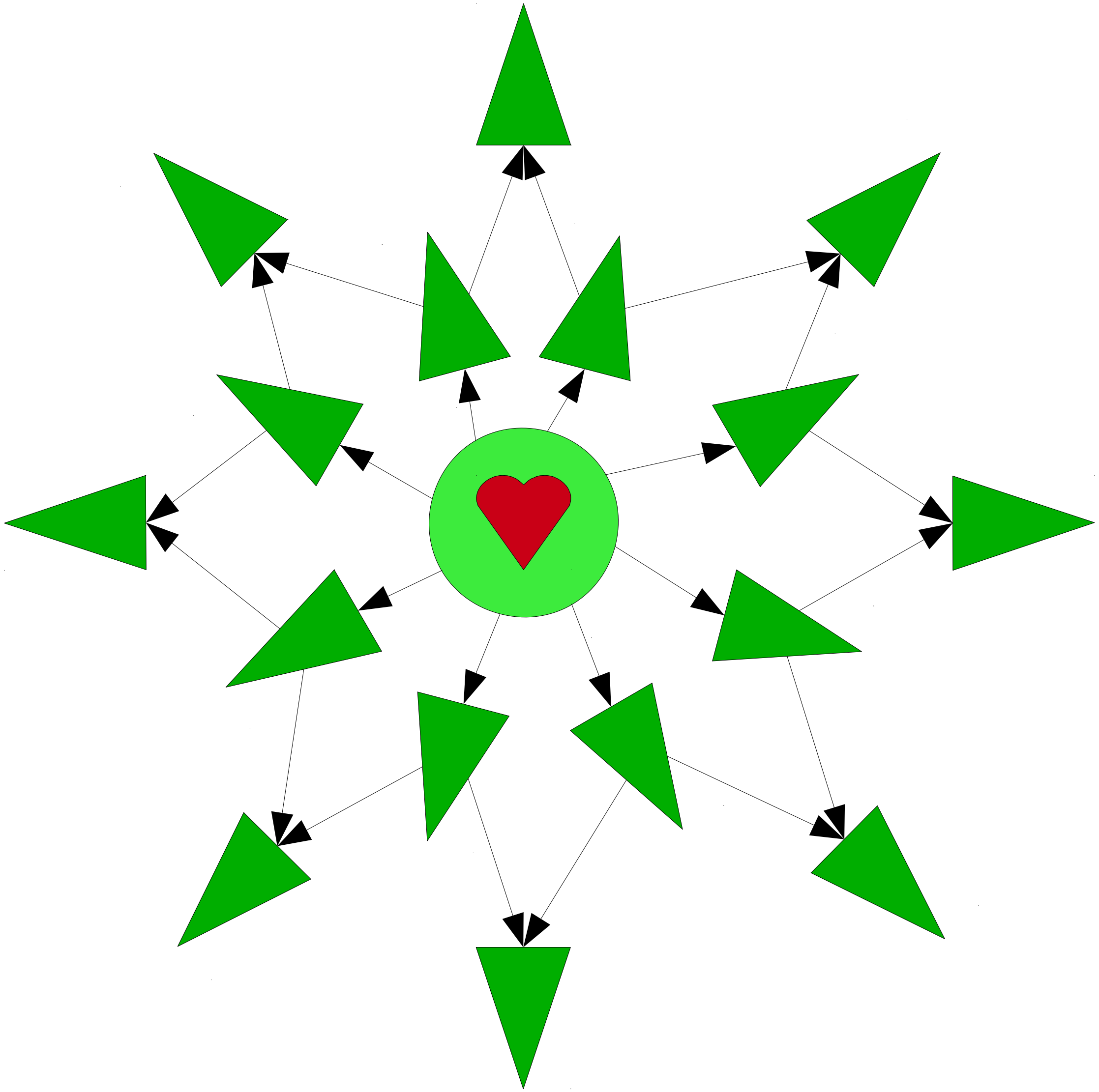
American

Dorm

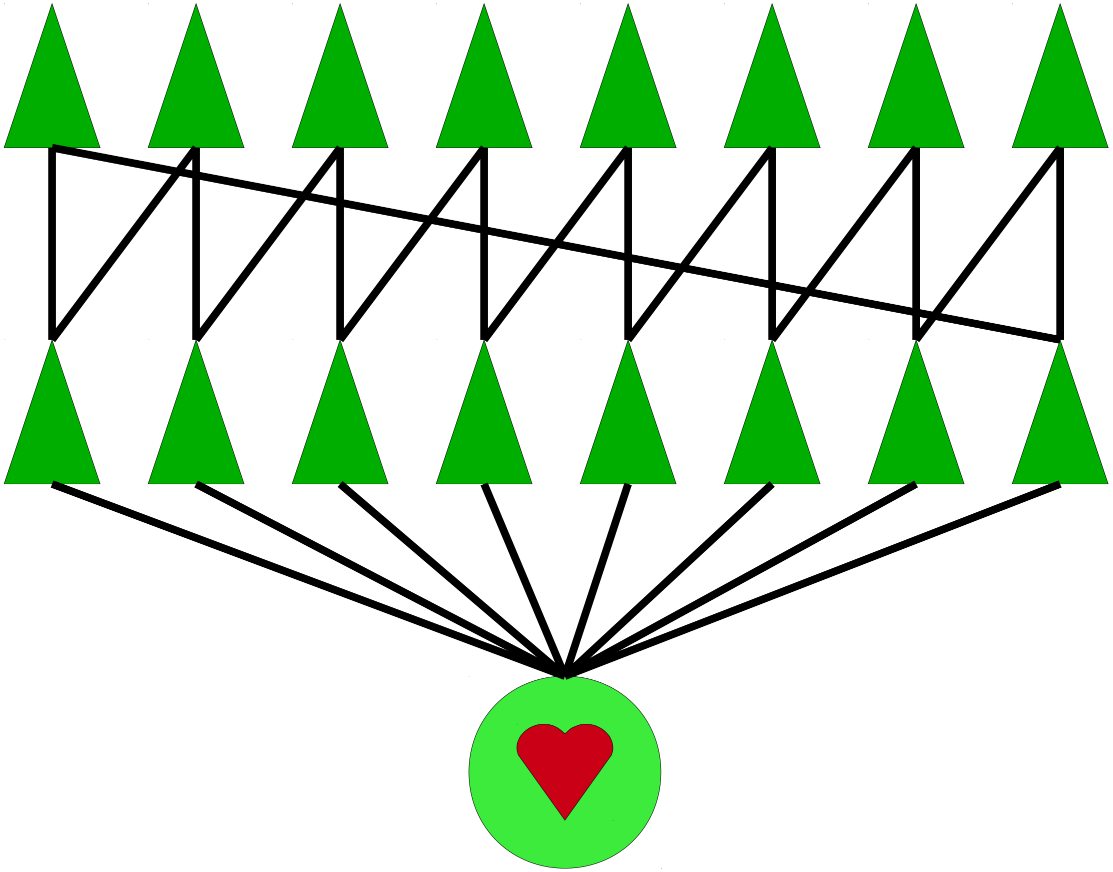
Not Tasty


Larger
...
4
3
2
1
0
Smaller

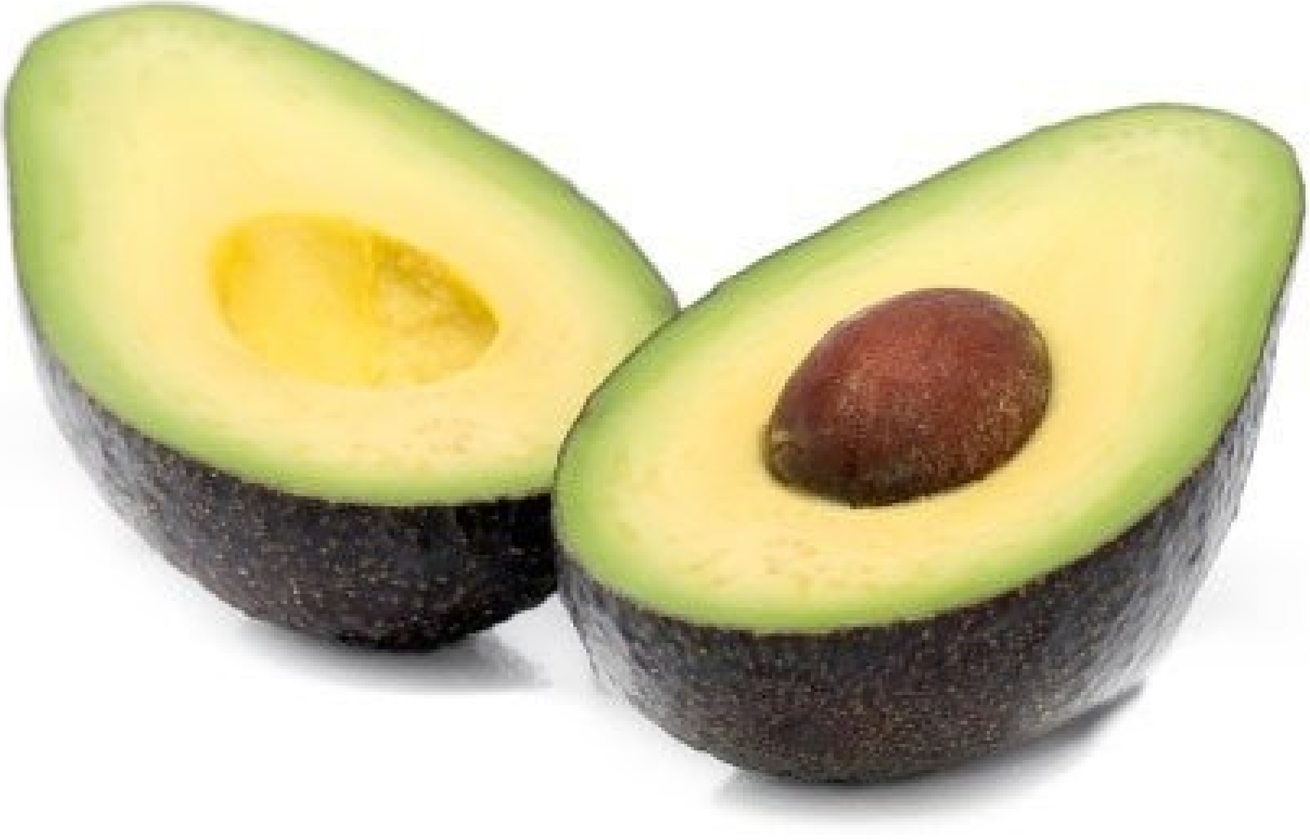
Hasse Artichokes



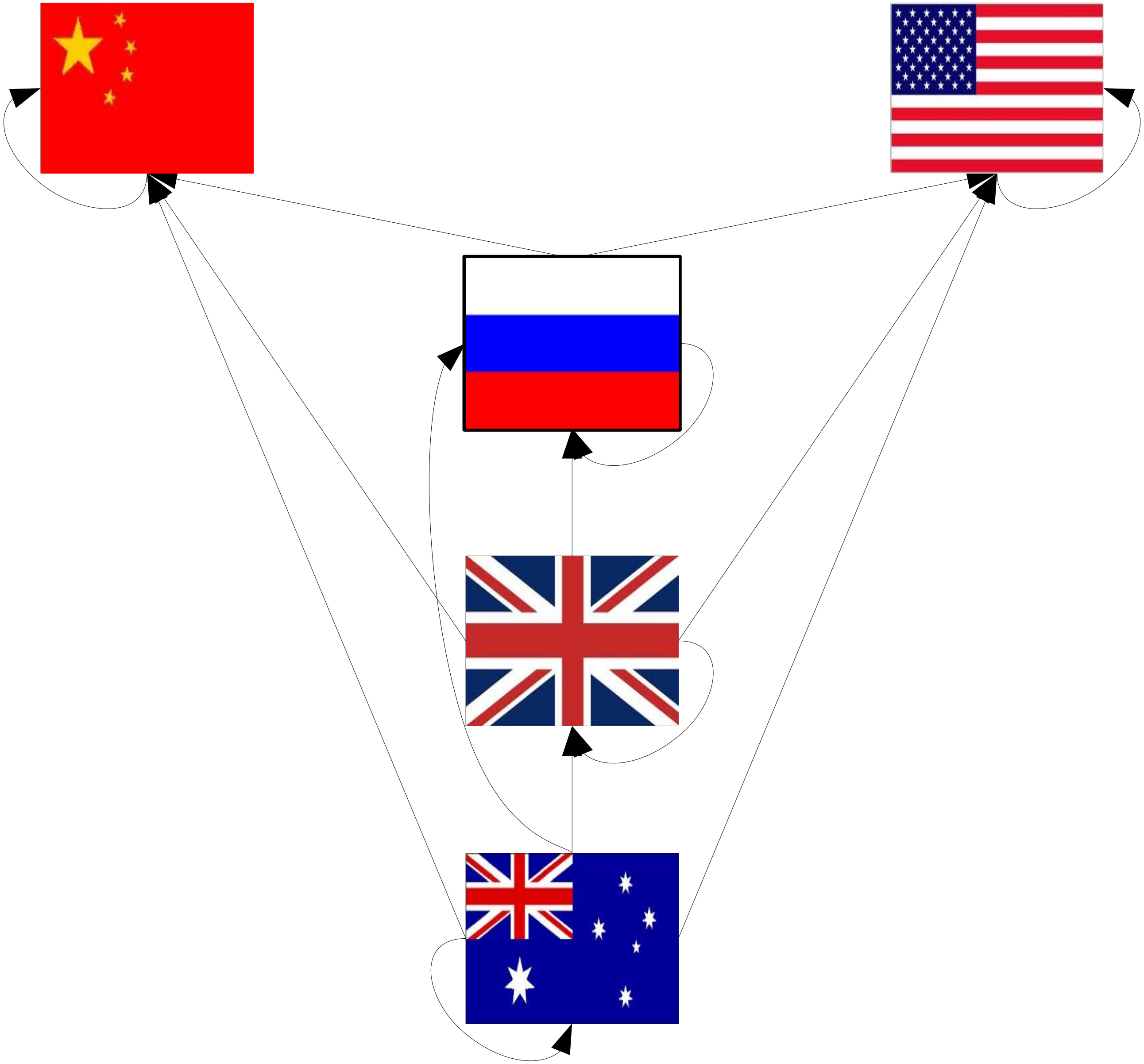
Hasse Artichokes

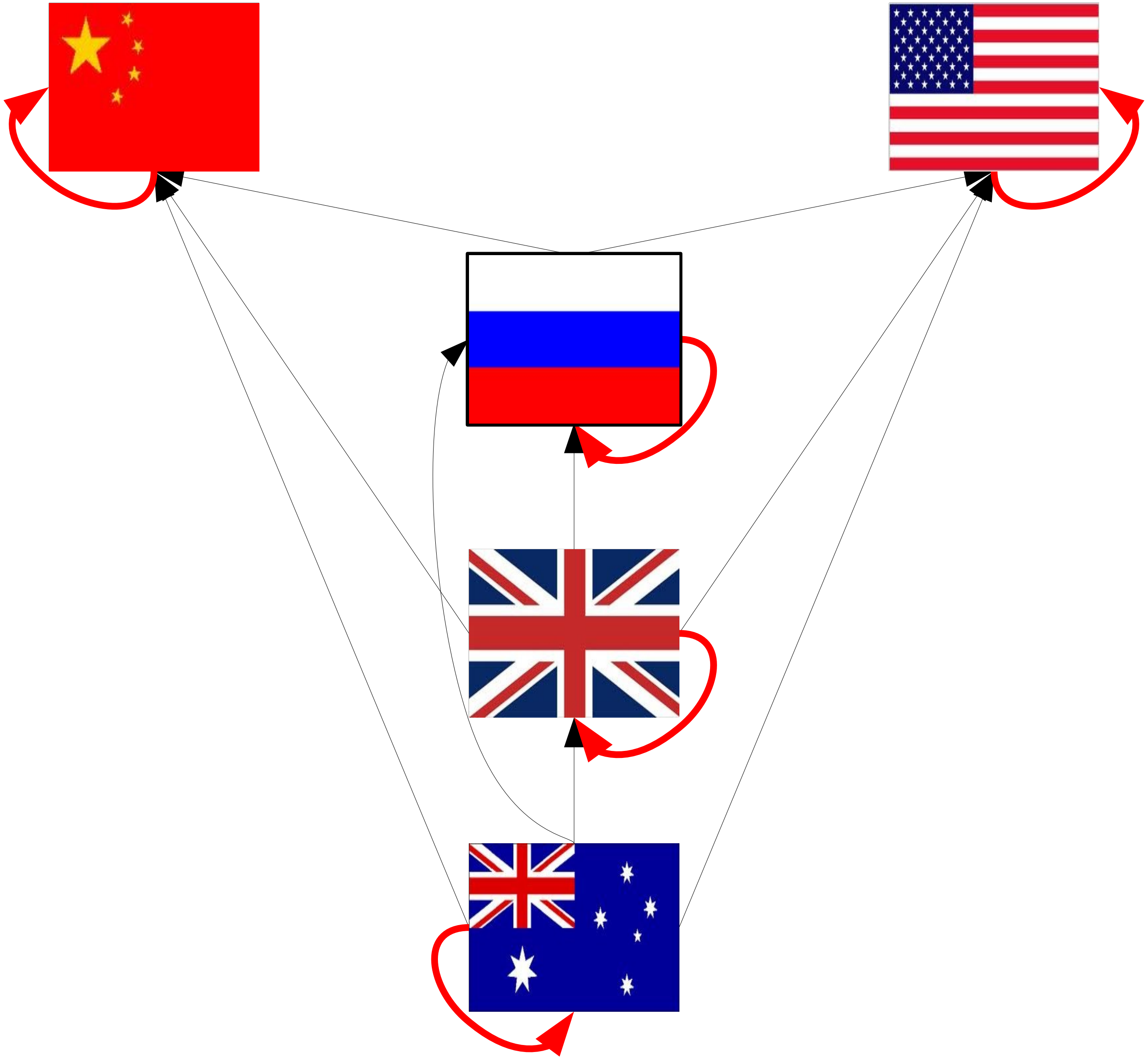


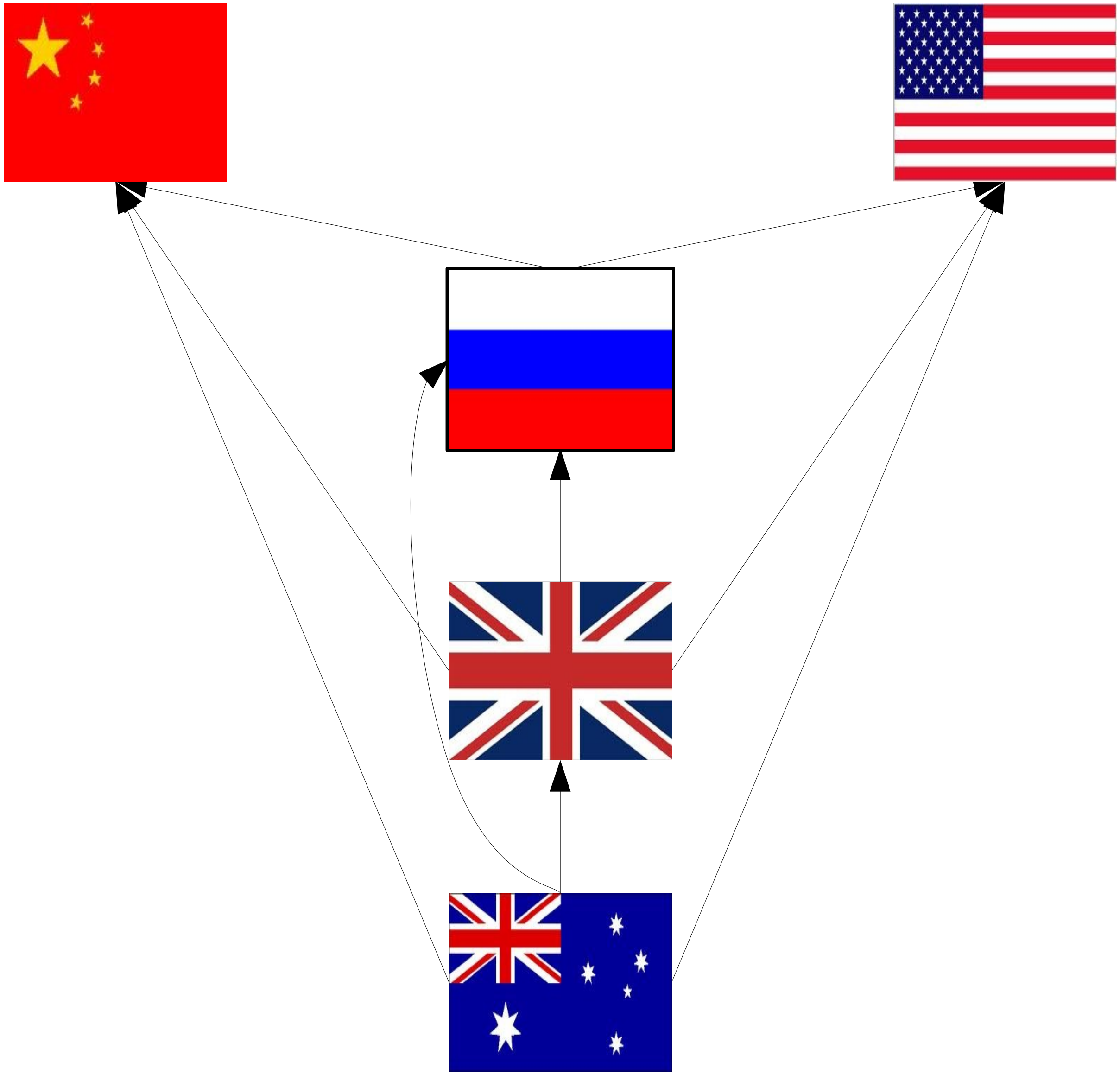
Hass Avocado

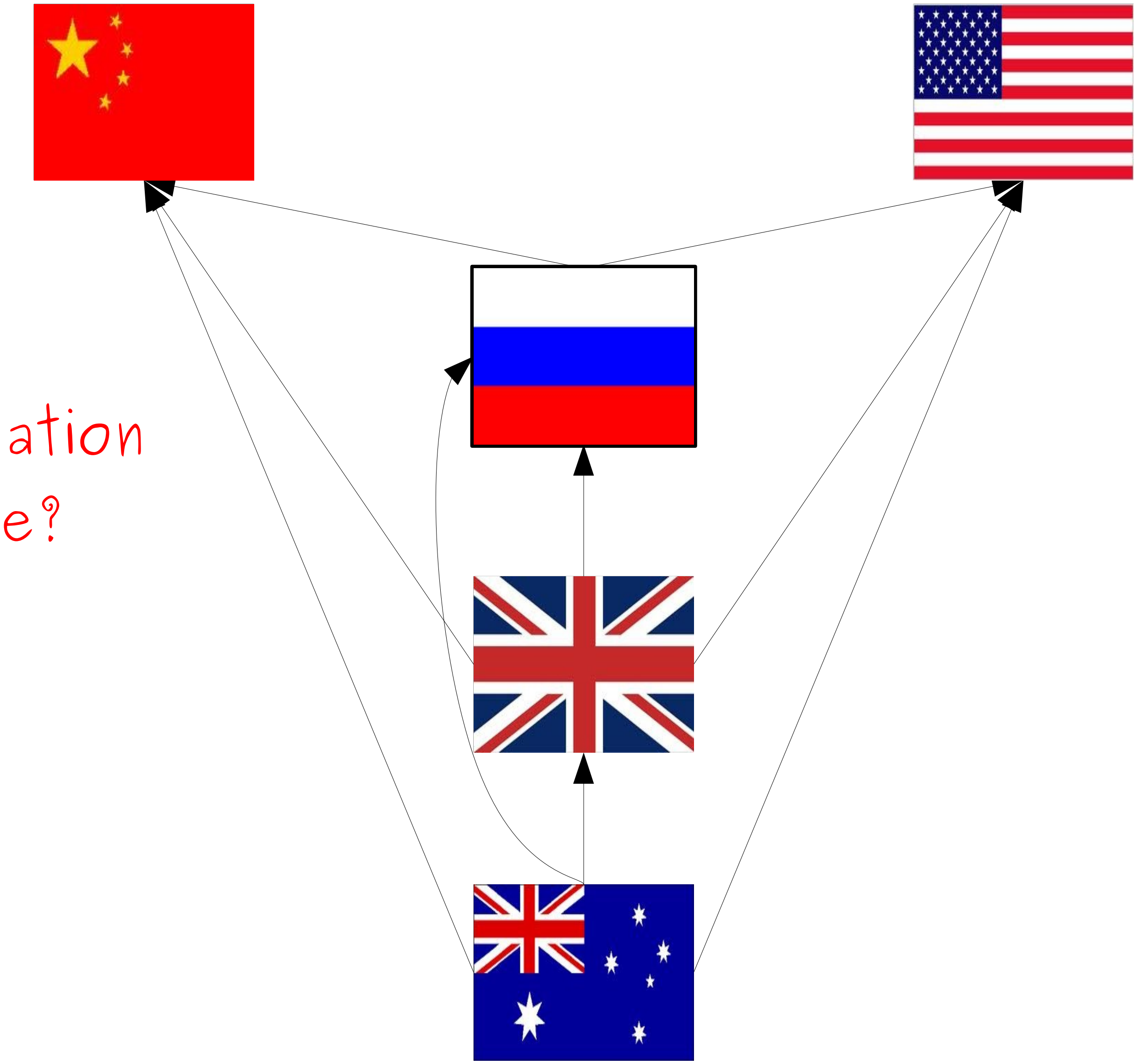


One Final Type of Order









Is this relation reflexive?

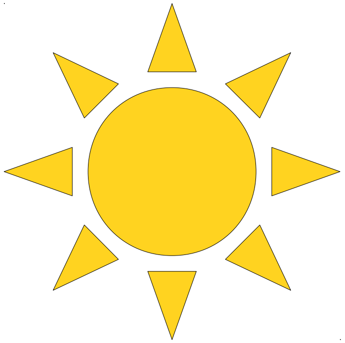
Irreflexivity

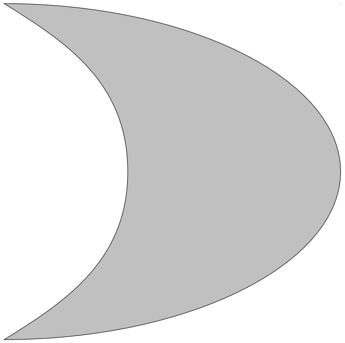
- Let R be a binary relation over A.
- R is **irreflexive** iff for any $a \in A$, aRa is false.
  - x is heavier than y
  - $x < y$
  - $x \neq y$
- Note that irreflexive does **not** mean “not reflexive.”
- Reflexive: Every element is **always** related to itself.
- Irreflexive: Every element is **never** related to itself.

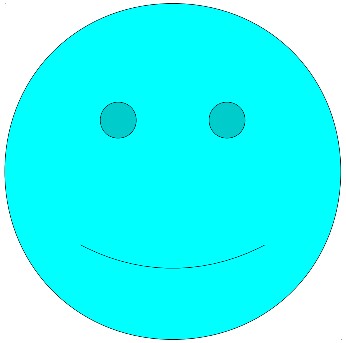
Strict Orders

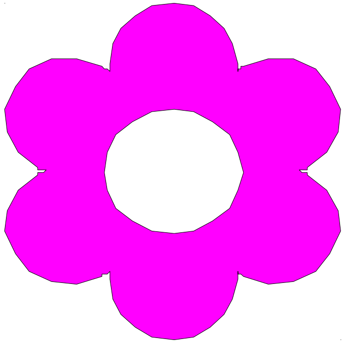
- A binary relation *R* over a set *A* is called a **strict order** iff it is
  - **irreflexive**,
  - **antisymmetric**, and
  - **transitive**.

Turning Things Around

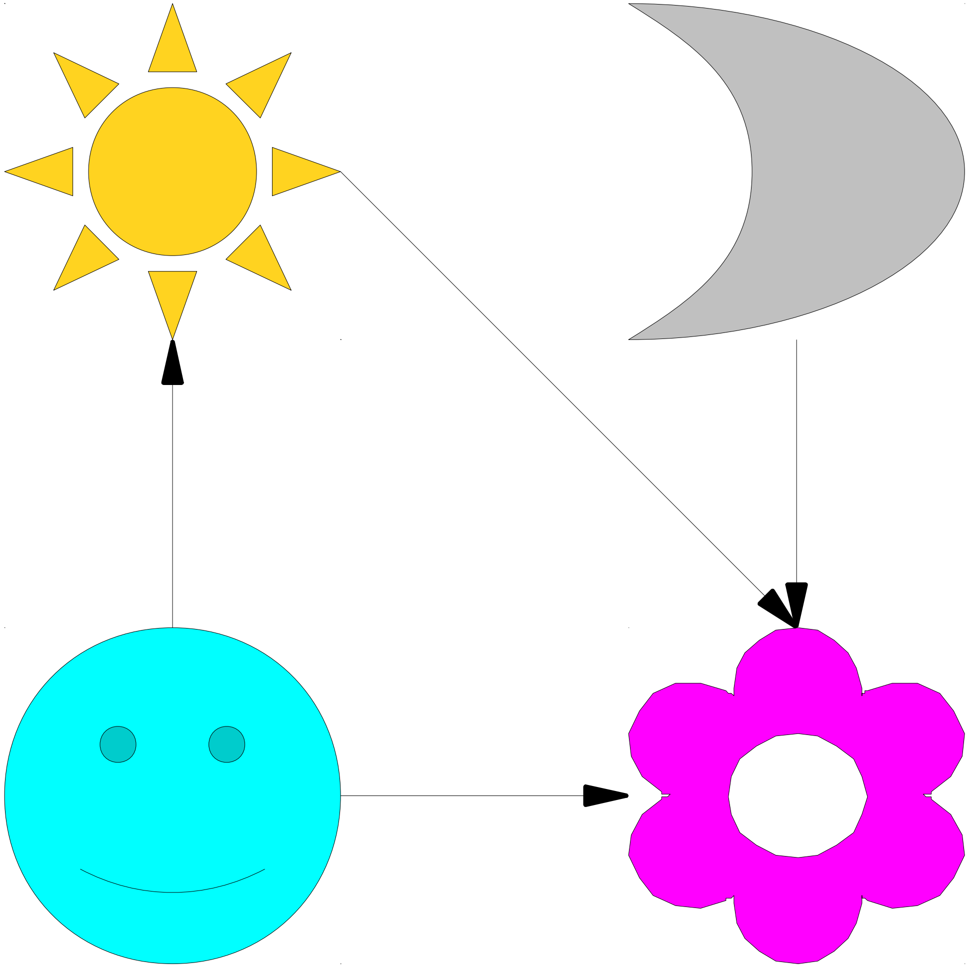




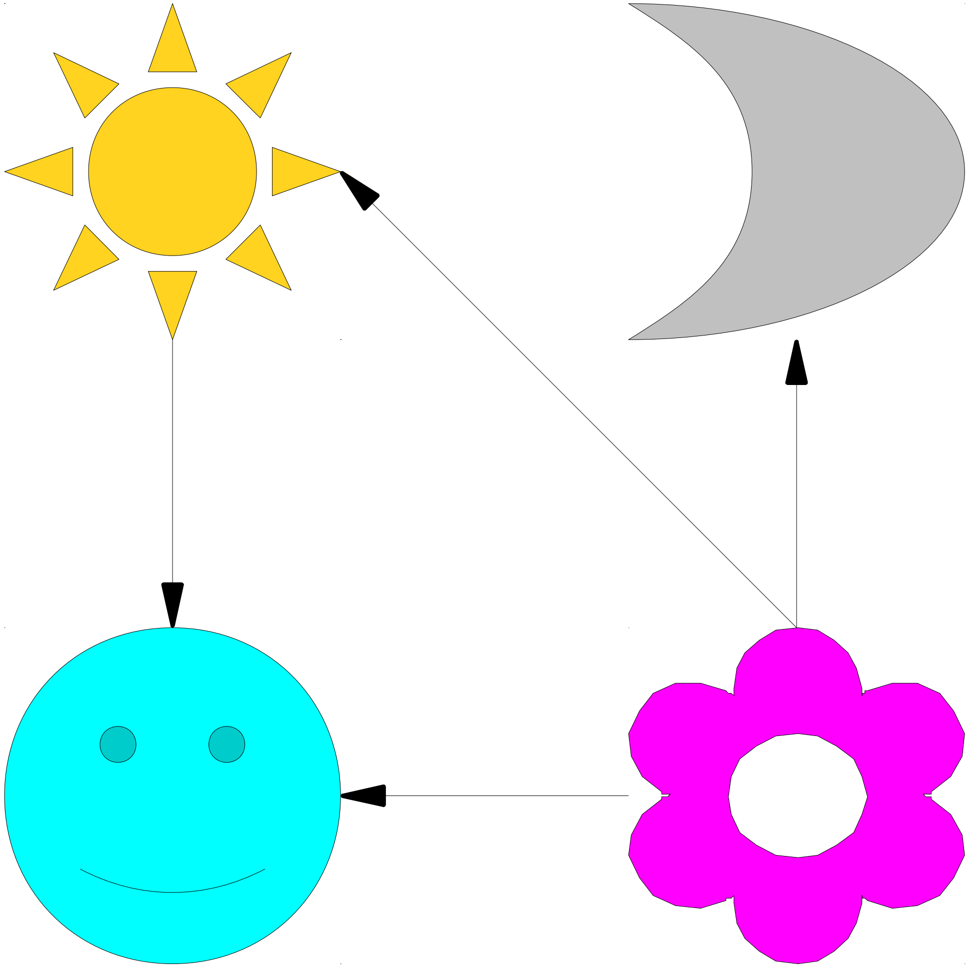




Turning Things Around



Turning Things Around



Inverses

- Given a relation *R*, the **inverse relation of R** (denoted ***R*** $^{-1}$) is the relation

$$R^{-1} = \{ (b, a) \mid aRb \}$$

- Example: The inverse of ≤ is ≥, since $a \geq b$ iff $b \leq a$.
- Note: inverse relations are **not** the same the opposite of the original relation.
  - The inverse of ≤ is **not** >.
- We will see this used more next lecture when we talk about functions.

Important Terms for Today

- Cartesian Product
- Ordered Pair
- Graph
- Path
- Connectivity
- Cycle
- Degree
- DAG
- Topological Sort
- Relation
- Reflexivity
- Symmetry
- Transitivity
- Antisymmetry
- Irreflexivity
- Totality
- Equivalence relation
- Equivalence class
- Partial order
- Hasse Diagram
- Total order
- Strict order
- Inverse relation