Discrete Mathematics

Relations

Adil M. Khan

Professor of Computer Science

Innopolis University

"Mathematics is the door and key to the sciences!"
- Roger Bacon -

Relations on Sets

- A mathematical tool for describing associations between elements of sets.
- **For example:**
  - “Relationship between an employee and his salary”,
  - “a positive integer and the on that it divides”,
  - “a real number and the one that is larger than it”,
  - “ a real number x and the value *f(x)* where *f* is a function”,
  - “a computer program and the variable it uses”,
  - “a computer language and valid statement in this language”, and so on….
- **Applications:** Determining which pairs of cities are connected in a network, producing useful way to store information in computer database

How to Express Relations?

- Some relationships will have some order associated with them. For example, “x is less than y”, “*t* watched movie *u*, while eating snack *v*”

- But not all relationships are like that.

- **For Example:**
  - “*x* and *y* are the same height”

- Most relationships are of the first kind; therefore, the most direct way to express a relationship between two sets is to use ordered pairs.

Tuple

- **Tuple:**
  - A tuple is a collection of *n* (not necessarily distinct) objects in some order. We denote the ordered tuple containing objects $x_1, x_2, x_3, \ldots, x_n$ as $(x_1, x_2, x_3, \ldots, x_n)$. To specifically indicate that a tuple contains *n* elements, we sometimes call it an *n-tuple*.

- Two tuples are equal if they have the same elements in the same order.
- For example, (1, 2, 3), and (1, 1, 1, 1) are both tuples
- (1, 2, 3) and (1, 2, 3) are equal to one another
- (1, 2, 3) and (3, 2, 1) are not! So are (1, 1, 2) and (1, 2)

Cartesian Product

- **Cartesian Product:**
  - Let A and B be sets. The Cartesian Product of A and B, denoted A x B, is the set

$$A \text{ x } B = \{(a, b) | a \in A \text{ and } b \in B\}$$

  - Intuitively, it is the set of all ordered pairs of whose first element is in A and whose second element is in B.
  - The ordering matters! $A \text{ x } B \neq B \text{ x } A$

Cartesian Product

- Cartesian product of a set with empty set:

$$A \text{ x } \emptyset = \{(a, b) | a \in A \text{ and } b \in \emptyset\}$$

- What does the above set contain??

Cartesian Product---Cont.

- Cartesian product of more than two sets:

- $A = \{1,2,3\}$ $B = \{x, y\}$ $C = \{*, \blacksquare\}$

- $A \text{ x } B \text{ x } C = \left\{\begin{array}{l} \big(1,(x,\ *)\big), \big(1,(y,\ *)\big), \big(2,(x,\ *)\big), \big(2,(x,\ *)\big), \\ \big(3,(x,\ *)\big), \big(3,(y,\ *)\big), \big(1,(x,\blacksquare)\big), \big(1,(x,\blacksquare)\big), \\ \big(2,(x,\blacksquare)\big), \big(2,(y,\blacksquare)\big), \big(3,(x,\blacksquare)\big), \big(3,(x,\blacksquare)\big) \end{array}\right\}$

Cartesian Product---Cont.

- **Cartesian Power**:

- $A = \{1,2,3\}$

- In that case, the set *(A x A) – also called Cartesian square* is the set

- $A \text{ x } A = \{(1,1),(1,2),(1,3),(2,1),(2,2),(2,3),(3,1),(3,2),(3,3)\}$

Cartesian Product---Cont.

- For $n \geq 1$, the n^{th} Cartesian power of a set, denoted as A^n, is the set formed by taking the Cartesian product of A with itself n times.

Binary Relation

- “*x* divides *y*” is a binary relation over integers
- “*x* is reachable from *y*” is a binary relationship over nodes in a graph

- **Given that we just learned about Tuples and Cartesian Products, how can we formally define/express binary relations?**

Binary Relation

- Let's think of a relation as a property that holds true for certain group of objects.
- For example, x is less than y" is true for 1 and 2
- Then using the set builder form

$$R = \{(x, y) \in N^2 | x < y\}$$

- Similarly

$$S = \{(x, y, x) \in R^3 | xy = z\}$$

- That is, we start off with some property and convert it into a set of tuples

Binary Relation

- **More formally:**
  - "Let A and B be sets. A binary relation from A to B is a ***subset*** of $A \times B$"
- Let R be a binary relation over a set A. Then we write $a\boldsymbol{R}b$ if $(a, b) \in R$

Binary Relation---Cont.

- **Example:**
  - Define a relation E from Z to Z as follows:
    For all $(m,n) \in (Z \text{ x } Z)$,
    $m\ E\ n \iff (m - n)$ is even

A. Is *4 E 0* ?? Is *2 E 6* ??
B. list five integers that are related by E to 1.
C. Prove that if n is any odd integer, then n E 1.

- **Solution: A:**

Yes, *4 E 0* because 4-0 = 4 and 4 is even.
Yes *2 E 6* because 2-6 = -4 and -4 is even.

Binary Relation---Cont.

- **Solution: B:**

There are many such lists. One is

1 because $1-1 = 0$ is even

3 because $3-1 = 2$ is even

5 because $5-1 = 4$ is even

-1 because $-1-1 = -2$ is even

-3 because $-3-1 = -4$ is even

Binary Relation---Cont.

- **Solution: C: Proof:**

Suppose *n* is any odd integer.

Then $n = 2k + 1$ for some integer *k*.

Now by definition of *E*, (***n E 1***) if and only if, ***n-1*** is even.

By substitution,

$$n - 1 = (2k + 1) - 1 = 2k$$

and since *k* is an integer, 2*k* is even.

- Hence *n E 1*.

Functions as Relations

- A relation is similar to a function
- In fact, every function $f: A \to B$ is a relation.
- In general, the difference between a function and a relation is that

  - A relation might associate multiple elements of B with single element of A

  - Whereas, a function can only associate at most one element of B with each element of A

Inverse of a Relation

- If R is a relation from A to B, then a relation R^{-1} from B to A can be defined by interchanging the elements of all the ordered pairs of R.

- **Definition:**
  - Let R be a relation from A to B. define the inverse relation R^{-1} from B to A as follows:

$$R^{-1} = \{(y, x) \in (B\text{x}A) | (x, y) \in R\}$$

  - This definition can be written operationally as follows:

For all $x \in A$ and $y \in B$, $(y, x) \in R^{-1} \Leftrightarrow (x, y) \in R$

Inverse of a Relation---Example

- **Example:**

  Let $A = \{2,3,4\}$ and $B = \{2,6,8\}$ and let R be the "divides" relation from A to B: for all $(x, y) \in (A\text{x}B)$,

  $$x\ R\ y \iff x|y \qquad \boxed{x\ divides\ y}$$

  A. State explicitly which ordered pairs are in R and R^{-1}, and draw arrow diagrams for R and R^{-1}.

  B. Describe R^{-1} in words.

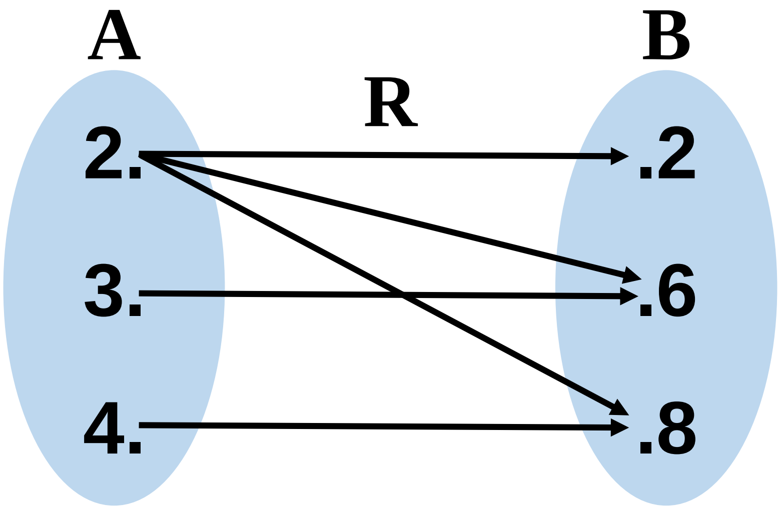
- **Solution: A:**

  $$R = \{(2,2), (2,6), (2,8), (3,6), (4,8)\}$$

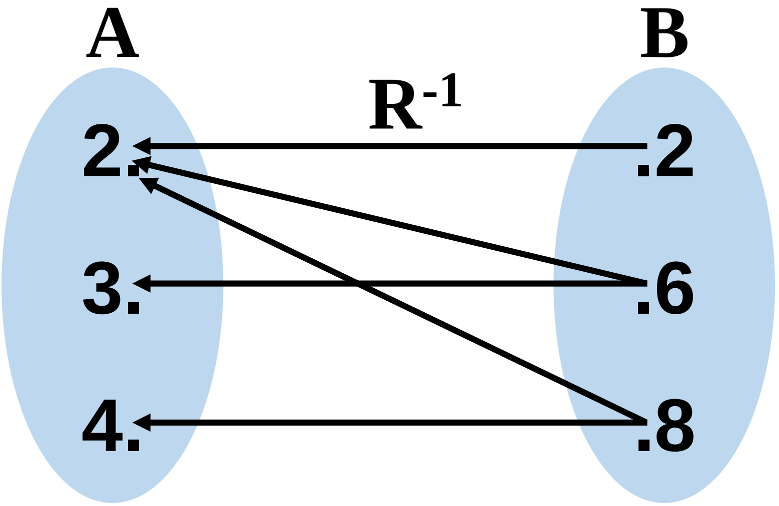
  $$R^{-1} = \{(2,2), (6,2), (8,2), (6,3), (8,4)\}$$

Inverse of a Relation---Example—Cont.

- **Solution: A:**

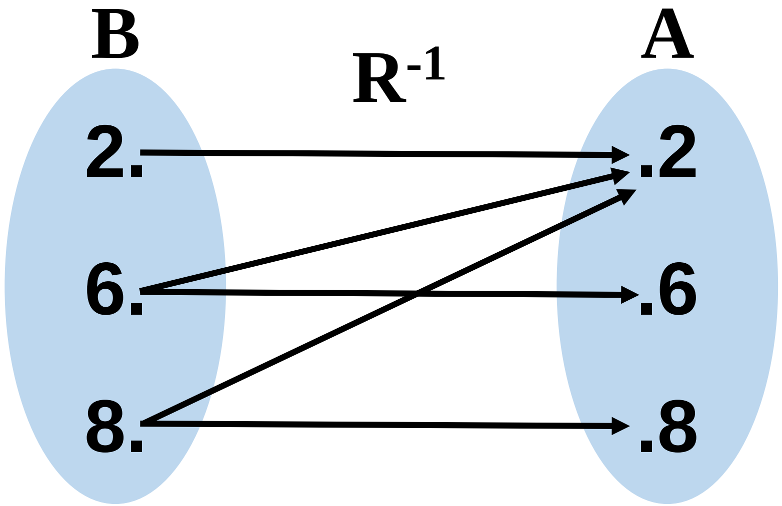


To draw the arrow diagram for R^{-1}, you can copy the arrow diagram for R but reverse the directions of the arrows.



Inverse of a Relation---Example—Cont.

- **Solution: A:** Or you can redraw the diagram so that B is on the left.



Solution: B:

R^{-1}, can be described in words as follows:

For all $(y, x) \in (B\text{x}A)$,

$y\ R^{-1}x \ \Leftrightarrow y$ is multiple of x.

Directed Graph of a Relation

- When a relation R is defined on a set A, the arrow diagram can be modified so that it becomes a directed graph.

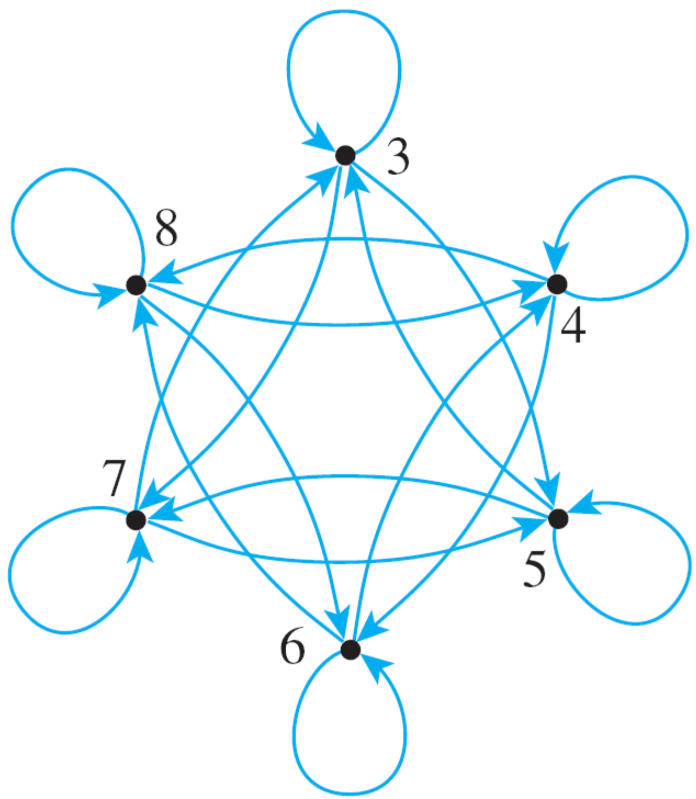
- Instead of representing A as two separate set of points, represent A only once, and draw an arrow from each point of A to each related point.

Directed Graph of a Relation

Let $A = \{3, 4, 5, 6, 7, 8\}$ and define a relation R on A as follows: For all $x, y \in A$,

$$x \; R \; y \Leftrightarrow 2 \mid (x - y).$$

Draw the directed graph of R.



N-ary Relations and Relational Databases

Form the mathematical foundation for relational database theory,

an *n-ary* relation is a subset of a Cartesian product of *n* sets.

- **Definition:**

  Given sets $A_1, A_2, A_3, \ldots, A_n$, and *n-ary* relation *R* on $A_1 \text{x} A_2 \text{x} A_3 \text{ x} \ldots \text{x} A_n$ is a subset of $A_1 \text{x} A_2 \text{x} A_3 \text{ x} \ldots \text{x} A_n$. The special cases of 2-ary, 3-ary and 4-ary relations are called **binary**, **ternary**, and **quaternary relations**, respectively.

N-ary Relations and Relational Databases

- **Example---Database:**
  - A simple hospital database
  - Let A_1 be a set of positive integers, A_2 a set of alphabetic character strings, A_3 a set of numeric character strings, and A_4 a set of alphabetic character strings.
- Let R be a quaternary relation on $A_1 \times A_2 \times A_3 \times A_4$ as follows:

$(a_1, a_2, a_3, a_4) \in R \Leftrightarrow$ a patient with patient ID number a_1, named a_2, was admitted on date a_3, with primary diagnosis a_4.

N-ary Relations and Relational Databases

- **Example---Database---Cont.:**

  - At a particular hospital, this relation might contain the following 4-tuples:

    - (011985, John Schmidt, 020710, asthma)
    - (574329, Tak Kurosawa, 0114910, pneumonia)
    - (466581, Mary Lazars, 0103910, appendicitis)
    - …

N-ary Relations and Relational Databases

- **Example---Database---Cont.:**

- Such tuples are normally thought of as being written in tables.

- Each row of the table corresponds to one tuple, whereas column headers describe the elements in them

- Data can be manipulated using operations

N-ary Relations and Relational Databases

- **Example---Database---Cont.:**

- From example, in SQL, if the above database is denoted S, the result of the query.

```
SELECT Patient_ID#, Name FROM S WHERE
Admission_Date = 010310
```

Would be:

```
466581 Mary Lazars,
244388 Sarah Wu.
```

Which is a list of the ID numbers and names of all patients admitted on 01-03-10.

N-ary Relations and Relational Databases

- **Example---Database---Cont.:**

  How is this results obtained?

  - By taking the intersection of the set $A_1 \times A_2 \times \{010310\} \times A_4$ with the database and,

  - then projecting onto the first two coordinates.

Special Binary Relations

- Relations that arise frequently in discrete math and computer science

- First, a basic set of terms is needed to describe different types of relations

- Given these terms, we can then introduce broad categories of relations that have certain similarities

Special Binary Relations---Cont.

Example:

- Let’s consider the following relations:
  - $x \leq y$
  - x is in the same connected component as y
  - x is the same color as y
- They are widely different from each other
- Yet they have two properties in common

Special Binary Relations---Cont.

Example---Cont.:

- $x \le y$
- x is in the same connected component as y
- x is the same color as y

- All these relations always relate an object to itself (not all binary relations have this property, $x < y$)
- Such relations are called *Reflexive*
- A binary relation R over a set A is called reflexive if for all $x \in A$, we have $\underline{x\ R\ x}$.
- Mathematically, $\boldsymbol{\forall a \in A.\ aRa}$
- A binary relation that lacks this property is called *Irreflexive – are they the opposite of each other?*

Special Binary Relations---Cont.

Example---Cont.:

- $x \leq y$
- x is in the same connected component as y
- x is the same color as y

- Secondly, all these relations have another property in common and that is:
- if xRy and yRz (where R is the appropriate binary relation), then it is also the case that xRz
- Relations that have this property are called *Transitive*.
- Not all binary relations have this property (Can you think of a relation that does not have this property?)

Special Binary Relations---Cont.

More Properties:

- Consider the following relations
  - $x = y$ (reflexive, and transitive)
  - $x \neq y$ (irreflexive and not transitive)
  - $x \leftrightarrow y$ (reflexive, and transitive)
- Though different, these relations do have one property in common
- That is, **if xRy then yRx**
- Relations that have this property are called ***Symmetric***
- ***Mathematically,* (∀a ∈ A. ∀b ∈ A. (aRb → bRa)**
- Not all relations are symmetric. Those that are not are called ***Asymmetric***
- Give an example of a binary relation that is asymmetric

Special Binary Relations---Cont.

All Three Together:

- **Reflexivity:** Every object has the same traits as itself. Thus our relation should be reflexive.

- **Symmetry:** If x and y have some trait in common, then surely y and x have the same trait in common.

- **Transitivity:** If x and y have some trait in common and y and z have the same trait in common, then x and z should have that same trait in common.

Equivalence Relations

- Informally, it is a binary relation over some set that tells whether two objects have some essential traits in common

- Formally, A binary relation *R* over a set *A* is called an **equivalence relation** if it is **reflexive, symmetric**, and **transitive.**

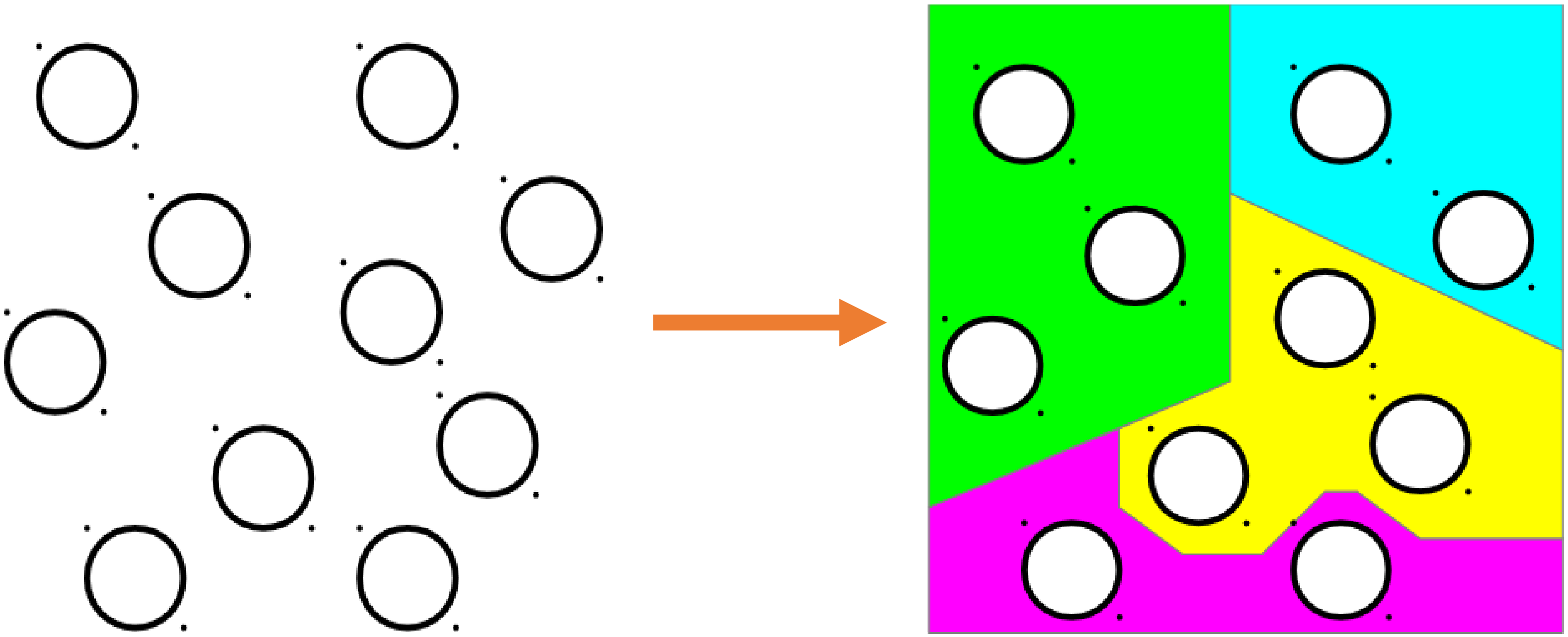
- Such relations split sets into disjoint classes of equivalent elements.

- For example, "x is the same color as y", "x$\leftrightarrow$ *y*"

- To formalize this, first we must understand what does it mean to split a set.

Equivalence Relations---Cont.

- What does it mean to split a set?



- This is called **partitioning** a set.

Partitions

- Given a set S, a partition of S is a set $X \subseteq \wp(S)$ (that is, a set of subsets of S) with the following properties:

    - The union of all sets in X is equal to S.

    - For any $S_1, S_2 \in X$ with $S_1 \neq S_2$,
      we have that $S_1 \cap S_2 = \emptyset$ (S_1 and S_2 are disjoint)

    - $\emptyset \notin X$.

Partitions

- **Example:**

For example,

let $S = \{1, 2, 3, 4, 5\}$.

Then the following are partitions of S:

$\{ \{1\}, \{2\}, \{3, 4\}, \{5\} \}$ and $\{\{1, 4\}, \{2, 3, 5\}\}$

What about this one?

$\{\{1, 3, 5\}, \{2\}\}$

Partition and Clustering---Cont.

- If you have a set of data, you can often learn something from the data by finding a “good” partition of that data and inspecting the partitions.

- Usually, the term **clustering** is used in data analysis rather than partitioning.

Partition and Clustering---Cont.

- **Question:**

  - What is the relationship between partitioning and equivalence relations?

  - To answer this, we must understand one more concept: **equivalence classes**

Partition and Clustering---Cont.

- **Things to know about Partitions:**
  - Let S be a set and X a partition of S. Then every element $u \in S$ belongs to exactly one set $Y \in X$.
  - If S is a set and X is a partition of S, then for any $u \in S$, we denote by $[u]_X$ the set $Y \in X$ such that $u \in Y$.
- **For example**:
  - If we let S = {1, 2, 3, 4, 5} as before and consider the partition X = {{1, 3, 5}, {2}, {4}}
- Then
  - $[1]_x = \{1, 3, 5\}$,
  - $[2]_x = \{2\}$,
  - $[3]_x = \{1, 3, 5\}$, etc

Equivalence Classes

- Given an equivalence relation R over a set A, for any $x \in A$, the equivalence class of x is the set
- $[x]_R = \{ y \in A \mid xRy \}$
- $[x]_R$ is the set of all elements of A that are related to x by relation R.

Equivalence Classes---Cont.

Can you guess what this set is?

$$\mathbf{X = \{ [x]_R \mid x \in A \}}$$

Equivalence Classes---Cont.

Lemma 1: Let R be an equivalence relation over A, and $X = \{ [x]_R \mid x \in A \}$. Then $\cup X = A$.

Proof: Let R be an equivalence relation over A, and $X = \{ [x]_R \mid x \in A \}$.

We will prove that

(i) $\cup X \subseteq A$ and

(ii) $A \subseteq \cup X$,

from which we can conclude that $\cup X = A$.

(i): Consider any $x \in X$.

By definition of $\cup X$, since $x \in X$, this means that there is some $[y]_R \in X$ such that $x \in [y]_R$.

By definition of $[y]_R$, since $x \in [y]_R$, this means that yRx. Since R is a binary relation over A, this means that $x \in A$. Since our choice of x **was arbitrary**, this shows that if $x \in \cup X$, then $x \in A$. Thus $\cup X \subseteq A$.

Equivalence Classes---Cont.

Lemma 1: Let R be an equivalence relation over A, and $X = \{ [x]_R \mid x \in A \}$. Then $\cup X = A$.

Proof:

(ii): consider any $x \in A$.

Since R is an equivalence relation, R is ***reflexive***, so xRx. Consequently, $x \in [x]_R$.

Since $[x]_R \in \cup X$, we have $x \in X$.

Since our choice of x is arbitrary, this would mean that any $x \in A$ satisfies $x \in \cup X$, so $A \subseteq \cup X$., as required.

Equivalence Classes---Cont.

Conclusion on Lemma 1:

We've established that $\bigcup X = A$, which is one of the three properties required for X to be a partition of A.

There are two more properties to be proved.

Equivalence Classes---Cont.

Lemma: Let R be an equivalence relation over A, and $X = \{ [x]_R \mid x \in A \}$. Then $\emptyset \notin X$.

Proof:

Using the logic of our previous proof, we have that for any $x \in A$, that $x \in [x]_R$.
Consequently, for any $x \in A$, we know $[x]_R \neq \emptyset$.

Thus $\emptyset \notin X$.

Thus we have shown that every set in X contains at least one element.

Equivalence Classes---Cont.

Lemma: Let R be an equivalence relation over A, and $X = \{ [x]_R \mid x \in A \}$. Then for any two sets $[x]_R, [y]_R \in X$, if $[x]_R \neq [y]_R$, then $[x]_R \cap [y]_R = \emptyset$.

Proof:

Proof is left for you to do as an exercise.

Equivalence Classes---Cont.

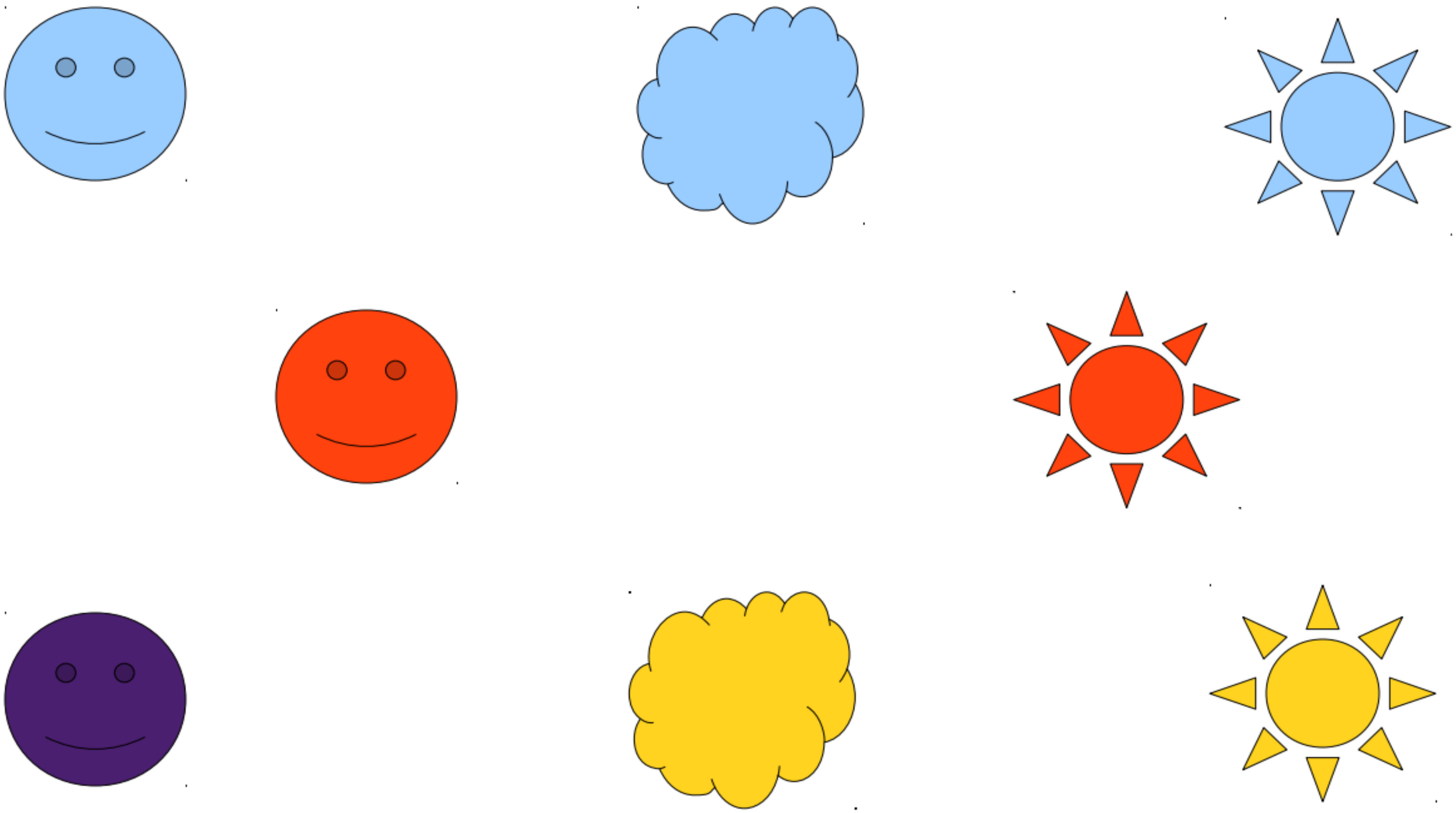
What have we got?

- The union of all sets in X is equal to A.
- Any two non-equal sets in X are disjoint.
- X does not contain the empty set.

Thus X is a partition of A.

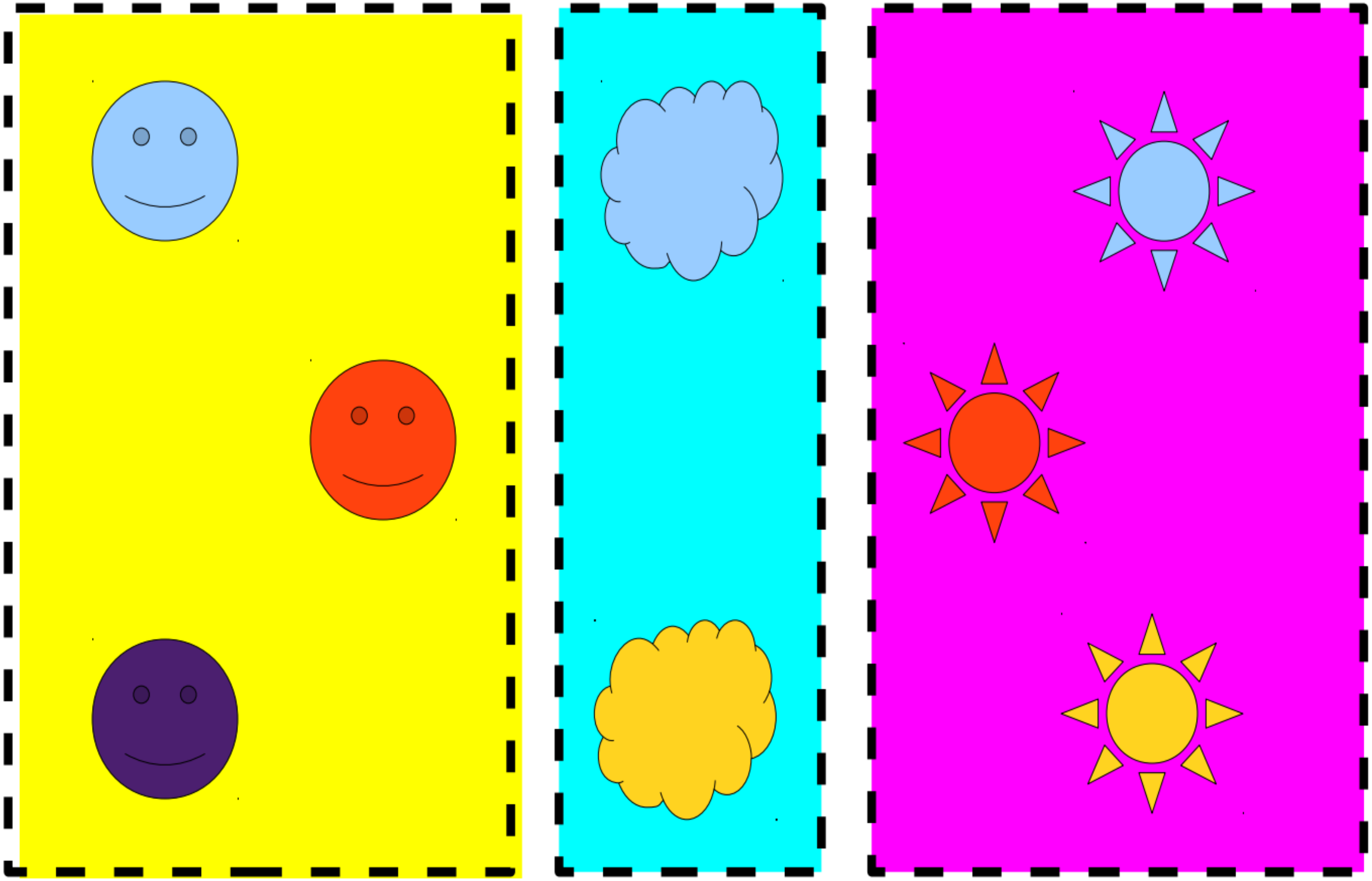
Hence **equivalence relations** **<u>split sets into disjoint classes of equivalent elements</u>**.

Equivalence Classes---Cont.



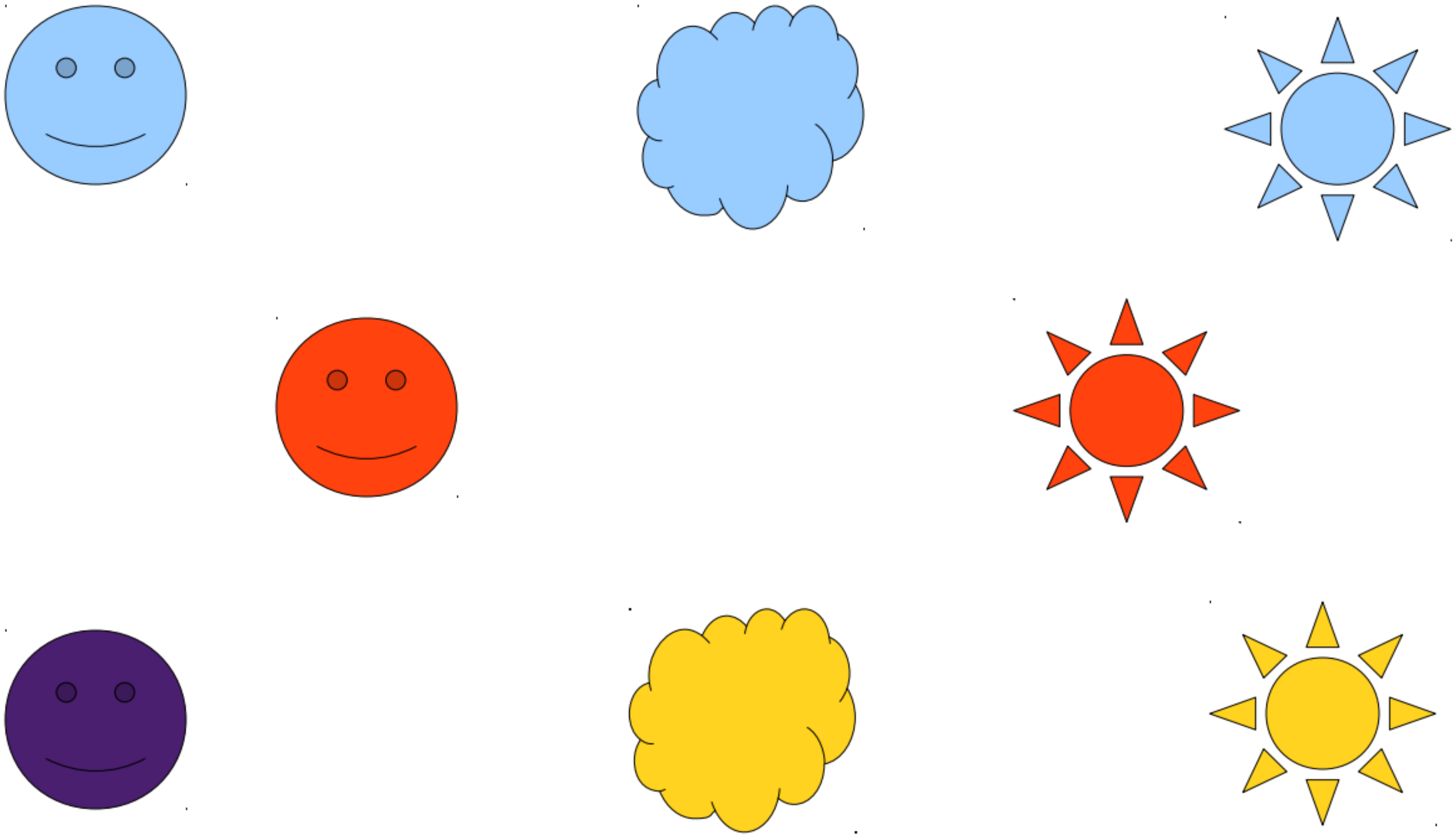
Let R be “has the same shape as”

Equivalence Classes---Cont.



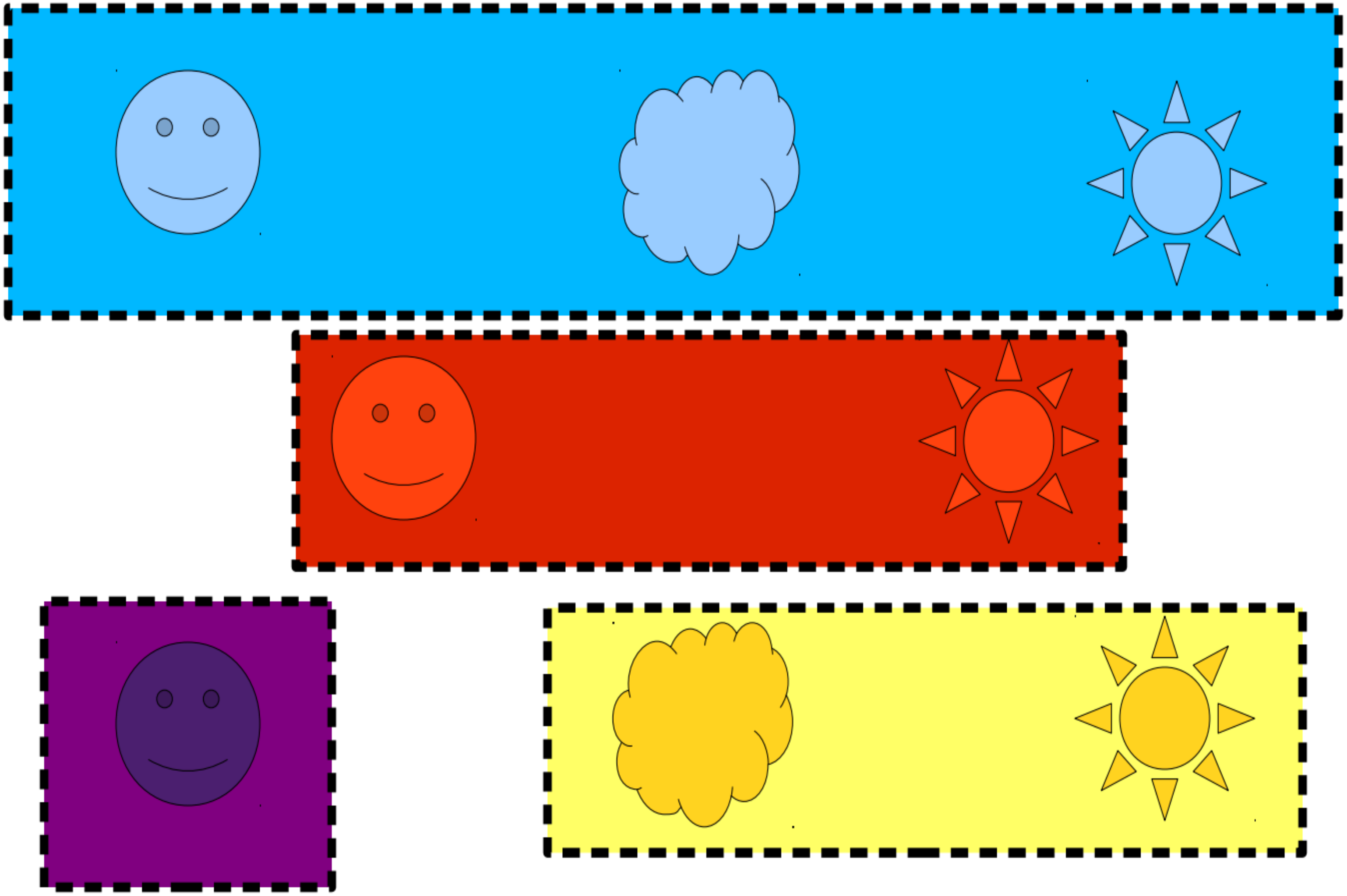
Let R be "has the same shape as"

Equivalence Classes---Cont.



Let T be “is the same **color** as”

Equivalence Classes---Cont.



Let T be "is the same **color** as"

Order Relations

Equivalence relations help us group objects with similar properties

Order relations allow us to rank objects against each other

For instance, we could take the relation $<$ over $\mathbb{N}$, where $x < y$ means "x is smaller than y.

Order Relations --- Strict Orders

- The relation $<$ over $\mathbb{N}$, where $x < y$ means "***x* is smaller than *y***".
- The relation "***x* is not as tasty as *y***," giving an ordering over different types of food
- The relation "***x* is smaller than *y***" over different buildings

These are examples of strict orders.

Order Relations --- Properties of Strict Orders

- **Irreflexive:** No object will be strictly worse/smaller than itself.

- **Transitive:** Let's suppose that $x < y$ and $y < z$.
  From this, we can conclude that $x < z$ (though not always true: think about Rock, Paper, Scissors – but mostly true)

- **Asymmetric:** if x runs faster than y, we know for certain that y does not run faster than x

Order Relations --- Strict Orders

Formal Definition

- A binary relation R over a set A is called a strict order if R is **irreflexive**, **asymmetric**, and **transitive**.

Partial Orders

- Strict orders do not give us a nice way to talk about relations like $\leq$ or $\subseteq$

- These relations are examples of Partial orders

- They still rank objects against one another, but are slightly more forgiving than $<$ and $\subset$

  - the relation < is “x is strictly less than y”
  - The relation $\leq$ is “ x is not greater than y”

Partial Orders

- As with equivalence relations and strict orders, we will also define partial orders by looking for properties shared by all partial orders

- Since, partial orders are a “soft” version of strict order, it would be nice to see which properties of strict orders carry over to partial orders.

- Properties of strict order: Irreflexivity, Transitivity, and Asymmetry

  - None of the partial orders are irreflexive ($x < x$, vs. $x \leq x$)
  - All of them are transitive
  - Asymmetry is a bit complex

Partial Orders

- Asymmetry – Think about $\leq$
  - Sometimes it acts asymmetrically (e.g. $37 \leq 100$),
  - but it is not always the case ($100 \leq 100$)
- If you will look at some other partial orders such as $\subseteq$ you will see the same pattern
- In fact, we can generalize this as follows

Partial Orders

- Partial orders are antisymmetric

  - A binary relation is called antisymmetric if $for\ any\ x\ and\ y\ \in A, if\ x\ \neq y,\ then\ if\ xRy, we\ have\ \cancel{yRx}$

  - Equivalently, a binary relation R over a set A is antisymmetric if $for\ any\ x, y\ \in A, if\ xRy\ and\ yRx, then\ x \neq y$