## Week 10 at a glance

**We will be learning and practicing to:**

- Clearly and unambiguously communicate computational ideas using appropriate formalism. Translate across levels of abstraction.
  - Defining functions, predicates, and binary relations using multiple representations
  - Determining whether a given binary relation is symmetric, antisymmetric, reflexive, and/or transitive
  - Determining whether a given binary relation is an equivalence relation and/or a partial order
  - Drawing graph representations of relations and functions e.g. Hasse diagram and partition diagram
- Know, select and apply appropriate computing knowledge and problem-solving techniques. Reason about computation and systems. Use mathematical techniques to solve problems. Determine appropriate conceptual tools to apply to new situations. Know when tools do not apply and try different approaches. Critically analyze and evaluate candidate solutions.
  - Using the definitions of the div and mod operators on integers
  - Using divisibility and primality predicates
  - Applying the definition of congruence modulo n and modular arithmetic
- Apply proof strategies, including direct proofs and proofs by contradiction, and determine whether a proposed argument is valid or not.
  - Using proofs as knowledge discovery tools to decide whether a statement is true or false

**TODO:**

Homework assignment 6 (due Thursday June 6, 2024).

Review for Final exam. The test is scheduled for Thursday June 13 11:30a-2:29pm, location REC GYM.

# Week 10 Monday: Equivalence Relations and Partial Orders

## Definitions and representations

A relation is an **equivalence relation** means it is reflexive, symmetric, and transitive.

A relation is a **partial ordering** (or partial order) means it is reflexive, antisymmetric, and transitive.

For a partial ordering, its **Hasse diagram** is a graph representing the relationship between elements in the ordering. The nodes (vertices) of the graph are the elements of the domain of the binary relation. The edges do not have arrow heads. The directionality of the partial order is indicated by the arrangements of the notes. The nodes are arranged so that nodes connected to nodes above them by edges indicate that the relation holds between the lower node and the higher node. Moreover, the diagram omits self-loops and omits edges that are guaranteed by transitivity.

Draw the Hasse diagram of the partial order on the set $\{a, b, c, d, e, f, g\}$ defined as

$$\{(a,a),(b,b),(c,c),(d,d),(e,e),(f,f),(g,g),\\ (a,c),(a,d),(d,g),(a,g),(b,f),(b,e),(e,g),(b,g)\}$$

## Exploring equivalence relations

A **partition** of a set $A$ is a set of non-empty, disjoint subsets $A_1, A_2, \cdots, A_n$ such that

$$A = \bigcup_{i=1}^{n} A_i = \{x \mid \exists i(x \in A_i)\}$$

An **equivalence class** of an element $a \in A$ with respect to an equivalence relation $R$ on the set $A$ is the set

$$\{s \in A \mid (a, s) \in R\}$$

We write $[a]_R$ for this set, which is the equivalence class of $a$ with respect to $R$.

**Fact**: When $R$ is an equivalence relation on a nonempty set $A$, the collection of equivalence classes of $R$ is a partition of $A$.

Also, given a partition $P$ of $A$, the relation $R_P$ on $A$ given by

$$R_P = \{(x, y) \in A \times A \mid x \text{ and } y \text{ are in the same part of the partition } P\}$$

is an equivalence relation on $A$.

*Recall*: We say $a$ is **congruent to** $b$ **mod** $n$ means $(a, b) \in R_{(\mathbf{mod}\ n)}$. A common notation is to write this as $a \equiv b(\mathbf{mod}\ n)$.

We can partition the set of integers using equivalence classes of $R_{(\mathbf{mod}\ 4)}$

$$[0]_{R_{(\mathbf{mod}\ 4)}} =$$
$$[1]_{R_{(\mathbf{mod}\ 4)}} =$$
$$[2]_{R_{(\mathbf{mod}\ 4)}} =$$
$$[3]_{R_{(\mathbf{mod}\ 4)}} =$$
$$[4]_{R_{(\mathbf{mod}\ 4)}} =$$
$$[5]_{R_{(\mathbf{mod}\ 4)}} =$$
$$[-1]_{R_{(\mathbf{mod}\ 4)}} =$$

$$\mathbb{Z} = [0]_{R_{(\mathbf{mod}\ 4)}} \cup [1]_{R_{(\mathbf{mod}\ 4)}} \cup [2]_{R_{(\mathbf{mod}\ 4)}} \cup [3]_{R_{(\mathbf{mod}\ 4)}}$$

Integers are useful because they can be used to encode other objects and have multiple representations. However, infinite sets are sometimes expensive to work with computationally. Reducing our attention to a *partition of the integers* based on congrunce mod $n$, where each part is represented by a (not too large) integer gives a useful compromise where many algebraic properties of the integers are preserved, and we also get the benefits of a finite domain. Moreover, modular arithmetic is well-suited to model any cyclic behavior.

**Lemma**: For $a, b \in \mathbb{Z}$ and positive integer $n$, $(a, b) \in R_{(\mathbf{mod}\ n)}$ if and only if $n|a - b$.

**Proof**:

**Modular arithmetic**:

**Lemma**: For $a, b, c, d \in \mathbb{Z}$ and positive integer $n$, if $a \equiv b$ ( **mod** $n$) and $c \equiv d$ ( **mod** $n$) then $a + c \equiv b + d$ ( **mod** $n$) and $ac \equiv bd$ ( **mod** $n$). **Informally**: can bring mod "inside" and do it first, for addition and for multiplication.

$(102 + 48)$ **mod** $10 =$ _____

$(7 \cdot 10)$ **mod** $5 =$ _____

$(2^5)$ **mod** $3 =$ _____

**Application: Cycling**

How many minutes past the hour are we at? *Model with* +15 **mod** 60

| **Time:** | 12:00pm | 12:15pm | 12:30pm | 12:45pm | 1:00pm | 1:15pm | 1:30pm | 1:45pm | 2:00pm |
|---|---|---|---|---|---|---|---|---|---|
| **"Minutes past":** | 0 | 15 | 30 | 45 | 0 | 15 | 30 | 45 | 0 |

Replace each English letter by a letter that's fifteen ahead of it in the alphabet *Model with* +15 **mod** 26

| **Original index:** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Original letter:** | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| **Shifted letter:** | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| **Shifted index:** | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

**Application: Cryptography**

**Definition**: Let $a$ be a positive integer and $p$ be a large[1] prime number, both known to everyone. Let $k_1$ be a secret large number known only to person $P_1$ (Alice) and $k_2$ be a secret large number known only to person $P_2$ (Bob). Let the **Diffie-Helman shared key** for $a, p, k_1, k_2$ be $(a^{k_1 \cdot k_2} \textbf{ mod } p)$.

**Idea**: $P_1$ can quickly compute the Diffie-Helman shared key knowing only $a, p, k_1$ and the result of $a^{k_2} \textbf{ mod } p$ (that is, $P_1$ can compute the shared key without knowing $k_2$, only $a^{k_2} \textbf{ mod } p$). Similarly, $P_2$ can quickly compute the Diffie-Helman shared key knowing only $a, p, k_2$ and the result of $a^{k_1} \textbf{ mod } p$ (that is, $P_2$ can compute the shared key without knowing $k_1$, only $a^{k_1} \textbf{ mod } p$). But, any person $P_3$ who knows neither $k_1$ nor $k_2$ (but may know any and all of the other values) cannot compute the shared secret efficiently.

**Key property for *shared* secret**:

$$\forall a \in \mathbb{Z} \forall b \in \mathbb{Z} \forall g \in \mathbb{Z}^+ \forall n \in \mathbb{Z}^+ ((g^a \textbf{ mod } n)^b, (g^b \textbf{ mod } n)^a) \in R_{(\textbf{mod } n)}$$

**Key property for shared *secret***:

There are efficient algorithms to calculate the result of modular exponentiation but there are no (known) efficient algorithms to calculate discrete logarithm.

[1] We leave the definition of "large" vague here, but think hundreds of digits for practical applications. In practice, we also need a particular relationship between $a$ and $p$ to hold, which we leave out here.

# Week 10 Wednesday: Applications

*Recall:*

A relation is an **equivalence relation** means it is reflexive, symmetric, and transitive.

An **equivalence class** of an element $a \in A$ with respect to an equivalence relation $R$ on the set $A$ is the set

$$\{s \in A \mid (a, s) \in R\}$$

We write $[a]_R$ for this set, which is the equivalence class of $a$ with respect to $R$.

A **partition** of a set $A$ is a set of non-empty, disjoint subsets $A_1, A_2, \cdots, A_n$ such that

$$A = \bigcup_{i=1}^{n} A_i = \{x \mid \exists i(x \in A_i)\}$$

**Claim**: For each $a \in U$, $[a]_E \neq \emptyset$.

**Proof**: Towards a _____ consider an arbitrary element $a$ in $U$. We will work to show that $[a]_E \neq \emptyset$, namely that $\exists x \in [a]_E$. By definition of equivalence classes, we can rewrite this goal as

$$\exists x \in U \ (\ (a, x) \in E\ )$$

Towards a _____, consider $x = a$, an element of $U$ by definition. By _____ of $E$, we know that $(a, a) \in E$ and thus the existential quantification has been proved.

**Claim**: For each $a \in U$, there is some $b \in U$ such that $a \in [b]_E$.

Towards a _____ consider an arbitrary element $a$ in $U$. By definition of equivalence classes, we can rewrite the goal as

$$\exists b \in U \ (\ (b, a) \in E\ )$$

Towards a _____, consider $b = a$, an element of $U$ by definition. By _____ of $E$, we know that $(a, a) \in E$ and thus the existential quantification has been proved.

**Claim**: For each $a, b \in U$ , ( $(a, b) \in E \ \rightarrow \ [a]_E = [b]_E$ ) and ( $(a, b) \notin E \ \rightarrow \ [a]_E \cap [b]_E = \emptyset$ )

**Corollary**: Given an equivalence relation $E$ on set $U$, $\{[x]_E \mid x \in U\}$ is a partition of $U$.

Last time, we saw that partitions associated to equivalence relations were useful in the context of modular arithmetic. Today we'll look at a different application.

Recall that in a movie recommendation system, each user's ratings of movies is represented as a $n$-tuple (with the positive integer $n$ being the number of movies in the database), and each component of the $n$-tuple is an element of the collection $\{-1, 0, 1\}$.

We call $Rt_5$ the set of all ratings 5-tuples.

Define $d : Rt_5 \times Rt_5 \to \mathbb{N}$ by

$$d(\ (\ (x_1, x_2, x_3, x_4, x_5), (y_1, y_2, y_3, y_4, y_5)\ )\ ) = \sum_{i=1}^{5} |x_i - y_i|$$

Consider the following binary relations on $Rt_5$.

$$E_{proj} = \{(\ (x_1, x_2, x_3, x_4, x_5), (y_1, y_2, y_3, y_4, y_5)\ ) \in Rt_5 \times Rt_5 \ \mid\ (x_1 = y_1) \wedge (x_2 = y_2) \wedge (x_3 = y_3)\}$$

Example ordered pair in $E_{proj}$:

Reflexive? Symmetric? Transitive? Antisymmetric?

$$E_{dist} = \{(u, v) \in Rt_5 \times Rt_5 \ \mid\ d(\ (u, v)\ ) \leq 2\}$$

Example ordered pair in $E_{dist}$:

Reflexive? Symmetric? Transitive? Antisymmetric?

$$E_{circ} = \{(u, v) \in Rt_5 \times Rt_5 \ \mid \ d(\ (\ (0,0,0,0,0)\ , u)\ ) = d(\ (\ (0,0,0,0,0), v\ )\ )\}$$

Example ordered pair in $E_{circ}$:

Reflexive? Symmetric? Transitive? Antisymmetric?

The partition of $Rt_5$ defined by ____ is

{ { $(-1,-1,-1,-1,-1), (-1,-1,-1,-1,0), (-1,-1,-1,-1,1), (-1,-1,-1,0,-1), (-1,-1,-1,0,0), (-1,-1,-1,0,1), (-1,-1,-1,1,-1), (-1,-1,-1,1,0), (-1,-1,-1,1,1)$ },
{ $(-1,-1,0,-1,-1), (-1,-1,0,-1,0), (-1,-1,0,-1,1), (-1,-1,0,0,-1), (-1,-1,0,0,0), (-1,-1,0,0,1), (-1,-1,0,1,-1), (-1,-1,0,1,0), (-1,-1,0,1,1)$ },
{ $(-1,-1,1,-1,-1), (-1,-1,1,-1,0), (-1,-1,1,-1,1), (-1,-1,1,0,-1), (-1,-1,1,0,0), (-1,-1,1,0,1), (-1,-1,1,1,-1), (-1,-1,1,1,0), (-1,-1,1,1,1)$ },
{ $(-1,0,-1,-1,-1), (-1,0,-1,-1,0), (-1,0,-1,-1,1), (-1,0,-1,0,-1), (-1,0,-1,0,0), (-1,0,-1,0,1), (-1,0,-1,1,-1), (-1,0,-1,1,0), (-1,0,-1,1,1)$ },
{ $(-1,0,0,-1,-1), (-1,0,0,-1,0), (-1,0,0,-1,1), (-1,0,0,0,-1), (-1,0,0,0,0), (-1,0,0,0,1), (-1,0,0,1,-1), (-1,0,0,1,0), (-1,0,0,1,1)$ },
{ $(-1,0,1,-1,-1), (-1,0,1,-1,0), (-1,0,1,-1,1), (-1,0,1,0,-1), (-1,0,1,0,0), (-1,0,1,0,1), (-1,0,1,1,-1), (-1,0,1,1,0), (-1,0,1,1,1)$ },
{ $(-1,1,-1,-1,-1), (-1,1,-1,-1,0), (-1,1,-1,-1,1), (-1,1,-1,0,-1), (-1,1,-1,0,0), (-1,1,-1,0,1), (-1,1,-1,1,-1), (-1,1,-1,1,0), (-1,1,-1,1,1)$ },
{ $(-1,1,0,-1,-1), (-1,1,0,-1,0), (-1,1,0,-1,1), (-1,1,0,0,-1), (-1,1,0,0,0), (-1,1,0,0,1), (-1,1,0,1,-1), (-1,1,0,1,0), (-1,1,0,1,-1)$ },
{ $(-1,1,1,-1,-1), (-1,1,1,-1,0), (-1,1,1,-1,1), (-1,1,1,0,-1), (-1,1,1,0,0), (-1,1,1,0,1), (-1,1,1,1,-1), (-1,1,1,1,0), (-1,1,1,1,1)$ },
{ $(0,-1,-1,-1,-1), (0,-1,-1,-1,0), (0,-1,-1,-1,1), (0,-1,-1,0,-1), (0,-1,-1,0,0), (0,-1,-1,0,1), (0,-1,-1,1,-1), (0,-1,-1,1,0), (0,-1,-1,1,1)$ },
{ $(0,-1,0,-1,-1), (0,-1,0,-1,0), (0,-1,0,-1,1), (0,-1,0,0,-1), (0,-1,0,0,0), (0,-1,0,0,1), (0,-1,0,1,-1), (0,-1,0,1,0), (0,-1,0,1,1)$ },
{ $(0,-1,1,-1,-1), (0,-1,1,-1,0), (0,-1,1,-1,1), (0,-1,1,0,-1), (0,-1,1,0,0), (0,-1,1,0,1), (0,-1,1,1,-1), (0,-1,1,1,0), (0,-1,1,1,1)$ },
{ $(0,0,-1,-1,-1), (0,0,-1,-1,0), (0,0,-1,-1,1), (0,0,-1,0,-1), (0,0,-1,0,0), (0,0,-1,0,1), (0,0,-1,1,-1), (0,0,-1,1,0), (0,0,-1,1,1)$ },
{ $(0,0,0,-1,-1), (0,0,0,-1,0), (0,0,0,-1,1), (0,0,0,0,-1), (0,0,0,0,0), (0,0,0,0,1), (0,0,0,1,-1), (0,0,0,1,0), (0,0,0,1,1)$ },
{ $(0,0,1,-1,-1), (0,0,1,-1,0), (0,0,1,-1,1), (0,0,1,0,-1), (0,0,1,0,0), (0,0,1,0,1), (0,0,1,1,-1), (0,0,1,1,0), (0,0,1,1,1)$ },
{ $(0,1,-1,-1,-1), (0,1,-1,-1,0), (0,1,-1,-1,1), (0,1,-1,0,-1), (0,1,-1,0,0), (0,1,-1,0,1), (0,1,-1,1,-1), (0,1,-1,1,0), (0,1,-1,1,1)$ },
{ $(0,1,0,-1,-1), (0,1,0,-1,0), (0,1,0,-1,1), (0,1,0,0,-1), (0,1,0,0,0), (0,1,0,0,1), (0,1,0,1,-1), (0,1,0,1,0), (0,1,0,1,-1)$ },
{ $(0,1,1,-1,-1), (0,1,1,-1,0), (0,1,1,-1,1), (0,1,1,0,-1), (0,1,1,0,0), (0,1,1,0,1), (0,1,1,1,-1), (0,1,1,1,0), (0,1,1,1,1)$ },
{ $(1,-1,-1,-1,-1), (1,-1,-1,-1,0), (1,-1,-1,-1,1), (1,-1,-1,0,-1), (1,-1,-1,0,0), (1,-1,-1,0,1), (1,-1,-1,1,-1), (1,-1,-1,1,0), (1,-1,-1,1,1)$ },
{ $(1,-1,0,-1,-1), (1,-1,0,-1,0), (1,-1,0,-1,1), (1,-1,0,0,-1), (1,-1,0,0,0), (1,-1,0,0,1), (1,-1,0,1,-1), (1,-1,0,1,0), (1,-1,0,1,1)$ },
{ $(1,-1,1,-1,-1), (1,-1,1,-1,0), (1,-1,1,-1,1), (1,-1,1,0,-1), (1,-1,1,0,0), (1,-1,1,0,1), (1,-1,1,1,-1), (1,-1,1,1,0), (1,-1,1,1,1)$ },
{ $(1,0,-1,-1,-1), (1,0,-1,-1,0), (1,0,-1,-1,1), (1,0,-1,0,-1), (1,0,-1,0,0), (1,0,-1,0,1), (1,0,-1,1,-1), (1,0,-1,1,0), (1,0,-1,1,1)$ },
{ $(1,0,0,-1,-1), (1,0,0,-1,0), (1,0,0,-1,1), (1,0,0,0,-1), (1,0,0,0,0), (1,0,0,0,1), (1,0,0,1,-1), (1,0,0,1,0), (1,0,0,1,1)$ },
{ $(1,0,1,-1,-1), (1,0,1,-1,0), (1,0,1,-1,1), (1,0,1,0,-1), (1,0,1,0,0), (1,0,1,0,1), (1,0,1,1,-1), (1,0,1,1,0), (1,0,1,1,1)$ },
{ $(1,1,-1,-1,-1), (1,1,-1,-1,0), (1,1,-1,-1,1), (1,1,-1,0,-1), (1,1,-1,0,0), (1,1,-1,0,1), (1,1,-1,1,-1), (1,1,-1,1,0), (1,1,-1,1,1)$ },
{ $(1,1,0,-1,-1), (1,1,0,-1,0), (1,1,0,-1,1), (1,1,0,0,-1), (1,1,0,0,0), (1,1,0,0,1), (1,1,0,1,-1), (1,1,0,1,0), (1,1,0,1,-1)$ },
{ $(1,1,1,-1,-1), (1,1,1,-1,0), (1,1,1,-1,1), (1,1,1,0,-1), (1,1,1,0,0), (1,1,1,0,1), (1,1,1,1,-1), (1,1,1,1,0), (1,1,1,1,1)$ }
}

The partition of $Rt_5$ defined by $E =$ ____ is

$$\{$$
$$[\ (0,0,0,0,0)\ ]_E$$
$$,[\ (0,0,0,0,1)\ ]_E$$
$$,[\ (0,0,0,1,1)\ ]_E$$
$$,[\ (0,0,1,1,1)\ ]_E$$
$$,[\ (0,1,1,1,1)\ ]_E$$
$$,[\ (1,1,1,1,1)\ ]_E$$
$$\}$$

How many elements are in each part of the partition?

**Scenario**: Good morning! You're a user experience engineer at Netflix. A product goal is to design customized home pages for groups of users who have similar interests. Your manager tasks you with designing an algorithm for producing a clustering of users based on their movie interests, so that customized homepages can be engineered for each group.

Your idea: equivalence relations!

$$E_{id} = \{(\ (x_1, x_2, x_3, x_4, x_5), (x_1, x_2, x_3, x_4, x_5)\ ) \mid (x_1, x_2, x_3, x_4, x_5) \in Rt_5\}$$

*Describe how each homepage should be designed ...*

$$E_{proj} = \{(\ (x_1, x_2, x_3, x_4, x_5), (y_1, y_2, y_3, y_4, y_5)\ ) \in Rt_5 \times Rt_5 \ \mid \ (x_1 = y_1) \wedge (x_2 = y_2) \wedge (x_3 = y_3)\}$$

*Describe how each homepage should be designed ...*

$$E_{circ} = \{(u, v) \in Rt_5 \times Rt_5 \ \mid \ d(\ (\ (0, 0, 0, 0, 0)\ , u)\ ) = d(\ (\ (0, 0, 0, 0, 0), v\ )\ )\}$$

*Describe how each homepage should be designed ...*

# Week 10 Friday: Review and Advice

Convert $(2A)_{16}$ to

- binary (base ____)

- decimal (base ____)

- octal (base ____)

- ternary (base ____)

The bases of RNA strands are elements of the set $B = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}$. The set of RNA strands $S$ is defined (recursively) by:

| | |
|---|---|
| Basis Step: | $\mathtt{A} \in S, \mathtt{C} \in S, \mathtt{U} \in S, \mathtt{G} \in S$ |
| Recursive Step: | If $s \in S$ and $b \in B$, then $sb \in S$ |

where $sb$ is string concatenation.

Each of the sets below is described using set builder notation. Rewrite them using the roster method.

- $\{s \in S \mid$ the leftmost base in $s$ is the same as the rightmost base in $s$ and $s$ has length $3\}$

- $\{s \in S \mid$ there are twice as many $\mathtt{A}$s as $\mathtt{C}$s in $s$ and $s$ has length $1\}$

Certain sequences of bases serve important biological functions in translating RNA to proteins. The following recursive definition gives a special set of RNA strands: The set of RNA strands $\hat{S}$ is defined (recursively) by

| | |
|---|---|
| Basis step: | $\mathtt{AUG} \in \hat{S}$ |
| Recursive step: | If $s \in \hat{S}$ and $x \in R$, then $sx \in \hat{S}$ |

where $R = \{\mathtt{UUU}, \mathtt{CUC}, \mathtt{AUC}, \mathtt{AUG}, \mathtt{GUU}, \mathtt{CCU}, \mathtt{GCU}, \mathtt{UGG}, \mathtt{GGA}\}$.

Each of the sets below is described using set builder notation. Rewrite them using the roster method.

- $\{s \in \hat{S} \mid s$ has length less than or equal to $5\}$

- $\{s \in S \mid$ there are twice as many $\mathtt{C}$s as $\mathtt{A}$s in $s$ and $s$ has length $6\}$

Let $W = \mathcal{P}(\{1, 2, 3, 4, 5\})$. Consider the statement

$$\forall A \in W \ \forall B \in W \ \forall C \in W \ ((A \cap B = A \cap C) \to (B = C))$$

Translate the statement to English. Negate the statement and translate this negation to English. Decide whether the original statement or its negation is true and justify your decision.

The set of linked lists of natural numbers $L$ is defined by

| | |
|---|---|
| Basis step: | $[] \in L$ |
| Recursive step: | If $l \in L$ and $n \in \mathbb{N}$, then $(n, l) \in L$ |

The function $length : L \to \mathbb{N}$ that computes the length of a list is

| | |
|---|---|
| Basis step: | $length([]) = 0$ |
| Recursive step: | If $l \in L$ and $n \in \mathbb{N}$, then $length((n, l)) = 1 + length(l)$ |

Prove or disprove: the function *length* is onto.

Prove or disprove: the function *length* is one-to-one.

Suppose $A$ and $B$ are sets and $A \subseteq B$:

**True or False**? If $A$ is infinite then $B$ is finite.

**True or False**? If $A$ is countable then $B$ is countable.

**True or False**? If $B$ is infinite then $A$ is finite.

**True or False**? If $B$ is uncountable then $A$ is countable.

Compute the last digit of

$$(42)^{2024}$$

*Extra* Describe the pattern that helps you perform this computation and prove it using mathematical induction.

# Looking forward

## Tips for future classes from the CSE 20 TAs and tutors

- In class
  - Go to class
  - There's usually a space for skateboards/longboards/eboards to go at the front or rear of the lecture hall
  - If you have a flask water bottle please ensure that its secured during a lecture and it cannot fall - putting on the floor often leads to it falling since people sometimes cross your seats.
  - Take notes - it's much faster and more effective to note-take in class than watch recordings after, particularly if you do so longhand
  - Resist the urge to sit in the back. You will be able to focus much better sitting near the front, where there are fewer screens in front of you to distract from the lecture content
  - If you bring your laptop to class to take notes / access class materials, sit towards the back of the room to minimize distractions for people sitting behind you!
  - Don't be afraid to talk to the people next to you during group discussions. Odds are they're as nervous as you are, and you can all benefit from sharing your thoughts and understanding of the material
  - Certain classes will podcast the lectures, just like Zoom archives lecture recordings, at podcast.ucsd.edu . If they aren't podcasted, and you want to record lectures, ask your professor for consent first
- Office hours, tutor hours, and the CSE building
  - Office hours are a good place to hang out and get work done while being able to ask questions as they come up
  - Get to know the CSE building: CSE B260, basement labs, office hours rooms
- Libraries and on-campus resources
  - Look up what library floors are for what, how to book rooms: East wing of Geisel is open 24/7 (they might ask to see an ID if you stay late), East Wing of Geisel has chess boards and jigsaw puzzles, study pods on the 8th floor, free computers/wifi
  - Know Biomed exists and is usually less crowded
  - Most libraries allow you to borrow whiteboards and markers (also laptops, tablets, microphones, and other cool stuff) for 24 hours
  - Take advantage of Dine with a prof / Coffee with a prof program. It's legit free food / coffee once per quarter.
  - When planning out your daily schedule, think about where classes are, how much time will they take, are their places to eat nearby and how you can schedule social time with friends to nearby areas
  - Take into account the distances between classes if they are back to back

- Final exams
  - Don't forget your university card during exams. Physical version is best for ID checks.
  - Look up seating assignments for exams and go early to make sure you're in the right place (check the exits to make sure you're reading it the right way)
  - Know where your exam is being held (find it on a map at least a day beforehand). Finals are often in strange places that take a while to find

# Review Quiz

1. Binary relations.

   (a)

   Assume there are five movies in the database, so that each user's ratings can be represented as a 5-tuple. We call $Rt_5$ the set of all ratings 5-tuples. Consider the binary relation on the set of all 5-tuples where each component of the 5-tuple is an element of the collection $\{-1, 0, 1\}$

   $$G_1 = \{(u, v) \mid \text{the ratings of users } u \text{ and } v \text{ agree about the first movie in the database}\}$$

   $$G_2 = \{(u, v) \mid \text{the ratings of users } u \text{ and } v \text{ agree about at least two movies}\}$$

   i. **True** or **False**: The relation $G_1$ holds of $u = (1, 1, 1, 1, 1)$ and $v = (-1, -1, -1, -1, -1)$
   ii. **True** or **False**: The relation $G_2$ holds of $u = (1, 0, 1, 0, -1)$ and $v = (-1, 0, 1, -1, -1)$
   iii. **True** or **False**: $G_1$ is reflexive; namely, $\forall u\ (\ (u, u) \in G_1\ )$
   iv. **True** or **False**: $G_1$ is symmetric; namely, $\forall u\ \forall v\ (\ (u, v) \in G_1 \to (v, u) \in G_1\ )$
   v. **True** or **False**: $G_1$ is transitive; namely, $\forall u\ \forall v\ \forall w(\ ((u, v) \in G_1 \wedge (v, w) \in G_1) \to (u, w) \in G_1\ )$
   vi. **True** or **False**: $G_2$ is reflexive; namely, $\forall u\ (\ (u, u) \in G_2\ )$
   vii. **True** or **False**: $G_2$ is symmetric; namely, $\forall u\ \forall v\ (\ (u, v) \in G_2 \to (v, u) \in G_2\ )$
   viii. **True** or **False**: $G_2$ is transitive; namely, $\forall u\ \forall v\ \forall w(\ ((u, v) \in G_2 \wedge (v, w) \in G_2) \to (u, w) \in G_2\ )$

   (b)

   Consider the binary relation on $\mathbb{Z}^+$ defined by $\{(a, b) \mid \exists c \in \mathbb{Z}(b = ac)\}$. Select all and only the properties that this binary relation has.

   i. It is reflexive.
   ii. It is symmetric.
   iii. It is transitive.
   iv. It is antisymmetric.

2. Equivalence relations.

   (a)

   Recall that the binary relation $EQ_{\mathbb{R}}$ on $\mathcal{P}(\mathbb{R})$ is

   $$\{(X_1, X_2) \in \mathcal{P}(\mathbb{R}) \times \mathcal{P}(\mathbb{R}) \mid |X_1| = |X_2|\}$$

   and $R_{(\mathbf{mod}\ n)}$ is the set of all pairs of integers $(a, b)$ such that $(a\ \mathbf{mod}\ n = b\ \mathbf{mod}\ n)$.
   Select all and only the correct items.

   i. $(\mathbb{Z}, \mathbb{R}) \in EQ_{\mathbb{R}}$
   ii. $(0, 1) \in EQ_{\mathbb{R}}$
   iii. $(\emptyset, \emptyset) \in EQ_{\mathbb{R}}$
   iv. $(-1, 1) \in R_{(\mathbf{mod}\ 2)}$

v. $(1, -1) \in R_{(\mathbf{mod}\ 3)}$

vi. $(4, 16, 0) \in R_{(\mathbf{mod}\ 4)}$

(b)

Fill in the blanks in the following proof that, for any equivalence relation $R$ on a set $A$,

$$\forall a \in A\ \forall b \in A\ ((a, b) \in R \leftrightarrow [a]_R \cap [b]_R \neq \emptyset)$$

**Proof**: Towards a **(a)**_____, consider arbitrary elements $a$, $b$ in $A$. We will prove the biconditional statement by proving each direction of the conditional in turn.

**Goal 1**: we need to show $(a, b) \in R \rightarrow [a]_R \cap [b]_R \neq \emptyset$ *Proof of Goal 1*: Assume towards a **(b)**_____ that $(a, b) \in R$. We will work to show that $[a]_R \cap [b]_R \neq \emptyset$. Namely, we need an element that is in both equivalence classes, that is, we need to prove the existential claim $\exists x \in A\ (x \in [a]_R \wedge x \in [b]_R)$. Towards a **(c)**_____, consider $x = b$, an element of $A$ by definition. By **(d)**_____ of $R$, we know that $(b, b) \in R$ and thus, $b \in [b]_R$. By assumption in this proof, we have that $(a, b) \in R$, and so by definition of equivalence classes, $b \in [a]_R$. Thus, we have proved both conjuncts and this part of the proof is complete.

**Goal 2**: we need to show $[a]_R \cap [b]_R \neq \emptyset \rightarrow (a, b) \in R$ *Proof of Goal 2*: Assume towards a **(e)**_____ that $[a]_R \cap [b]_R \neq \emptyset$. We will work to show that $(a, b) \in R$. By our assumption, the existential claim $\exists x \in A\ (x \in [a]_R \wedge x \in [b]_R)$ is true. Call $w$ a witness; thus, $w \in [a]_R$ and $w \in [b]_R$. By definition of equivalence classes, $w \in [a]_R$ means $(a, w) \in R$ and $w \in [b]_R$ means $(b, w) \in R$. By **(f)**_____ of $R$, $(w, b) \in R$. By **(g)**_____ of $R$, since $(a, w) \in R$ and $(w, b) \in R$, we have that $(a, b) \in R$, as required for this part of the proof.

Consider the following expressions as options to fill in the two proofs above. Give your answer as one of the numbers below for each blank a-c. You may use some numbers for more than one blank, but each letter only uses one of the expressions below.

| | |
|---|---|
| i exhaustive proof | vi proof by contrapositive |
| ii proof by universal generalization | vii proof by contradiction |
| iii proof of existential using a witness | viii reflexivity |
| iv proof by cases | ix symmetry |
| v direct proof | x transitivity |

3. Partial orders.

(a) Consider the partial order on the set $\mathcal{P}(\{1, 2, 3\})$ given by the binary relation $\{(X, Y) \mid X \subseteq Y\}$

i. How many nodes are in the Hasse diagram of this partial order?

ii. How many edges are in the Hasse diagram of this partial order?

(b) Consider the binary relation on $\{1, 2, 4, 5, 10, 20\}$ defined by $\{(a, b) \mid \exists c \in \mathbb{Z}(b = ac)\}$.

i. How many nodes are in the Hasse diagram of this partial order?

ii. How many edges are in the Hasse diagram of this partial order?

4. Equivalence classes and partitions.

(a)

Select all and only the correct statements about an equivalence relation $E$ on a set $U$:

i. $E \in U \times U$
ii. $E = U \times U$
iii. $E \subseteq U \times U$
iv. $\forall x \in U\ ([x]_E \in U)$
v. $\forall x \in U\ ([x]_E \subseteq U)$
vi. $\forall x \in U\ ([x]_E \in \mathcal{P}(U))$
vii. $\forall x \in U\ ([x]_E \subseteq \mathcal{P}(U))$

(b) Select all and only the partitions of $\{1, 2, 3, 4, 5\}$ from the sets below.

i. $\{1, 2, 3, 4, 5\}$
ii. $\{\{1, 2, 3, 4, 5\}\}$
iii. $\{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}\}$
iv. $\{\{1\}, \{2, 3\}, \{4\}\}$
v. $\{\{\emptyset, 1, 2\}, \{3, 4, 5\}\}$

5. Modular exponentiation.

Modular exponentiation is required to carry out the Diffie-Helman protocol for computing a shared secret over an unsecure channel.

Consider the following algorithm for fast exponentiation (based on binary expansion of the exponent).

Modular Exponentation

```
1   procedure modular exponentiation(b: integer;
2                    n = (a_{k-1}a_{k-2}...a_1a_0)_2, m: positive integers)
3   x := 1
4   power := b mod m
5   for i:= 0 to k-1
6     if a_i = 1 then x:= (x · power) mod m
7     power := (power · power) mod m
8   return x {x equals b^n mod m}
```

(a) If we wanted to calculate $3^8$ **mod** 7 using the modular exponentation algorithm above, what are the values of the parameters $b$, $n$, and $m$? (Write these values in usual, decimal-like, mathematical notation.)

(b) Give the output of the *modular exponentiation* algorithm with these parameters, i.e. calculate $3^8$ **mod** 7. (Write these values in usual, decimal-like, mathematical notation.)