Week 6 Part B highlights

- Define (binary) relations and give examples.
- Determine and prove whether a given binary relation is symmetric, antisymmetric, reflexive, transitive
- Define and use the congruence modulo m equivalence relation
- Represent equivalence relations as partitions and vice versa.

Definition: When A and B are sets, we say any subset of $A \times B$ is a **binary relation**. A relation R can also be represented as

- A function $f_{TF} : A \times B \to \{T, F\}$ where, for $a \in A$ and $b \in B$, $f_{TF}(\ (a,b)\ ) = \begin{cases} T & \text{when } (a,b) \in R \\ F & \text{when } (a,b) \notin R \end{cases}$
- A function $f_{\mathcal{P}} : A \to \mathcal{P}(B)$ where, for $a \in A$, $f_{\mathcal{P}}(a) = \{b \in B \mid (a,b) \in R\}$

When A is a set, we say any subset of $A \times A$ is a (binary) **relation** on A.

Recall that S is defined as the set of all RNA strands, strings made of the bases in $B = \{\texttt{A}, \texttt{U}, \texttt{G}, \texttt{C}\}$.

Definition: We say that a RNA strand s_1 is "within one edit" of a RNA strand s_2 to mean

$within1_{TF}$: _____ $\to$ _____ $within1_{\mathcal{P}}$: ____ $\to$ _____

$within1_{TF}(s_1, s_2)$ = ________________________ $within1_{\mathcal{P}}(s_1)$ = ____________________________

$W_1 = \{$____________________________________$\}$

Commonly used notation: aRb which means $(a,b) \in R$, for $a \in A$ and $b \in B$.

What relations have you seen so far?

Example binary relations :

Define the relation R_{EQ} on A to be the set of all pairs of sets that have the same size.

R_{EQ} is a relation on the set ______________

In set builder notation, $R_{EQ} =$ ____________________________.

Define the relation $R_{(\textbf{mod } n)}$ on A to be the set of all pairs of integers (a, b) such that $(a \textbf{ mod } n = b \textbf{ mod } n)$.

$R_{(\textbf{mod } n)}$ is a relation on the set ___________

In set builder notation, $R_{(\textbf{mod } n)} =$ ___________________________________.

Then, a is **congruent to** b **mod** n, denoted as $a \equiv b(\textbf{mod } n)$ means $(a, b) \in R_{(\textbf{mod } n)}$.

Some example elements of $R_{(\textbf{mod } 4)}$ are:

Define the relation R_{SUB} to be the set of all pairs of sets where one is the subset of another.

R_{SUB} is a relation on the set ______________

In set builder notation, $R_{SUB} =$ __________________________.

For relation R on a set A, we can represent this relation as a **graph**: a collection of nodes (vertices) and edges (arrows). The nodes of the graph are the elements of A and there is an edge from a to b exactly when $(a, b) \in R$.

Definitions: (*Zybook 10.2*)
A relation R on a set A is called **reflexive** means aRa for every element $a \in A$.
A relation R on a set A is called **symmetric** means bRa whenever aRb, for all $a, b \in A$.
A relation R on a set A is called **antisymmetric** means if $a \neq b$ then $\neg(aRb \wedge bRa)$, for all $a, b \in A$.
A relation R on a set A is called **transitive** means whenever aRb and bRc, then $aRc \in R$, for all $a, b, c \in A$.

Visualize the properties of a relation R on A using arrow diagrams

A relation R on a set A is called **reflexive** means $(a, a) \in R$ for every element $a \in A$.

Informally, every element is related to itself.

Graphically, there are self-loops (edge from a node back to itself) at every node.

A relation R on a set A is called **symmetric** means $(b, a) \in R$ whenever $(a, b) \in R$, for all $a, b \in A$.

Informally, order doesn't matter for this relation.

Graphically, every edge has a paired "backwards" edge so we might as well drop the arrows and think of edges as undirected.

A relation R on a set A is called **transitive** means whenever $(a, b) \in R$ and $(b, c) \in R$, then $(a, c) \in R$, for all $a, b, c \in A$.

Informally, chains of relations collapse.

Graphically, there's a shortcut between any endpoints of a chain of edges.

A relation R on a set A is called **antisymmetric** means $\forall a \in A\, \forall b \in A\ (\ (\ (a, b) \in R \wedge (b, a) \in R\ ) \rightarrow a = b\ )$

Informally, the relation has directionality.

Graphically, can organize the nodes of the graph so that all non-self loop edges go up.

When the domain is $\{a, b, c, d, e, f, g, h\}$ define a relation that is **not reflexive** and is **not symmetric** and is **not transitive**.

When the domain is $\{a, b, c, d, e, f, g, h\}$ define a relation that is **not reflexive** but is **symmetric** and is **transitive**.

When the domain is $\{a, b, c, d, e, f, g, h\}$ define a relation that is **symmetric** and is **antisymmetric**.

Relation	Reflexive? (Why?)	Symmetric (Why?)	Transitive? Why?	Antisymmetric? Why?
W_1				
$R_{(\mathbf{mod}\ 4)}$				
R_{EQ}				
R_{SUB}				

(*Zybook 10.4*) A relation is an **equivalence relation** means it is________________________________

(*Zybook 10.3*)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 whose nodes (vertices) are the elements of the domain of the binary relation and which are located such that nodes connected to nodes above them by (undirected) 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

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

Summary: binary relations can be useful for organizing elements in a domain. Some binary relations have special properties that make them act like some familiar relations. Equivalence relations (reflexive, symmetric, transitive binary relations) “act like” equals. Partial orders (reflexive, antisymmetric, transitive binary relations) “act like” less than or equals to.

Exploring equivalence relations

(*Zybook 10.3*) A **partition** of a set A is a set (family) $P \subseteq \mathcal{P}(A)$ such that the following conditions hold

(*Zybook 10.4*) An **equivalence class** of an element $a \in A$ with respect to an equivalence relation R on the set A is the set of all elements in A that are related to a. We denote it as $[a]_R$

$$[a]_R = \underline{\hspace{4cm}}$$

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

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

Some examples of elements of $[5]_{R_{(\textbf{mod } 4)}}$ are: ________________________________

Some examples of elements of $[9]_{R_{(\textbf{mod } 4)}}$ are: ________________________________

Some examples of elements of $[6]_{R_{(\textbf{mod } 4)}}$ are: ________________________________

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

$$\begin{aligned}
[0]_{R_{(\textbf{mod } 4)}} &= \\
[1]_{R_{(\textbf{mod } 4)}} &= \\
[2]_{R_{(\textbf{mod } 4)}} &= \\
[3]_{R_{(\textbf{mod } 4)}} &= \\
[4]_{R_{(\textbf{mod } 4)}} &= \\
[5]_{R_{(\textbf{mod } 4)}} &= \\
[-1]_{R_{(\textbf{mod } 4)}} &=
\end{aligned}$$

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

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.

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.

Conventions for today: We will use $U = \{r_1, r_2, \cdots, r_t\}$ to refer to an arbitrary set of user ratings (we'll pick some specific examples to explore) that are a subset of Rt_5. We will be interested in creating partitions $C_1, \cdots, C_m$ of U. We'll assume that each user represented by an element of U has a unique ratings tuple.

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

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

Your idea: equivalence relations! You offer your manager three great options on Rt_5:

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

Describe the idea behind E_{proj}

Example ordered pair in E_{proj}:

Reflexive? Symmetric? Transitive? Antisymmetric?

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

Describe the idea behind E_{dist}

Example ordered pair in E_{dist}:

Reflexive? Symmetric? Transitive? Antisymmetric?

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

Describe the idea behind E_{circ}

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

$$
\begin{array}{l}
\{ \\
\quad [\ (0,0,0,0,0)\ ]_E \\
\quad ,[\ (0,0,0,0,1)\ ]_E \\
\quad ,[\ (0,0,0,1,1)\ ]_E \\
\quad ,[\ (0,0,1,1,1)\ ]_E \\
\quad ,[\ (0,1,1,1,1)\ ]_E \\
\quad ,[\ (1,1,1,1,1)\ ]_E \\
\}
\end{array}
$$

How many elements are in each part of the partition?

Proofs related to equivalence relations

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.