CS 40
FOUNDATIONS OF CS

[redacted]

Summer 2024

Week 10 Learning goals

- 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.
- Application of equivalence relations to clustering (Netflix example)

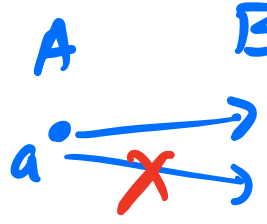
Relations

New idea: A **binary relation** R from A to B is **any** subset of A x B.

Sometimes we will say “a **(binary) relation on A**” where A is a set, which means a relation from A to A, which means a subset of A x A.

$f: A \to B$ → Is it a binary relation?

f is a binary relation s.t.,
∀a∈A, ∃ exactly one b∈B s.t. (a,b)∈f.



Application: Model similarity of RNA strands

An idea important to DNA sequencing – how "close" are two DNA strands?

For today – consider all RNA strands that are no more than "1 edit" away from a strand.

1 character diff

An "edit" for today means replacing one base. (In general, also consider deletions and insertions!)

GAUUACAAUCUCACGAAGGG

|

UCCCA

Application: genomics

Which of these strands are within 1 edit of UCUCA?

A: UCCCA

B: UCCA

C: UCUCA

D: AUCUCA

E: All of the above

Application: genomics

What type of data structure could we use to represent "all pairs of RNA strands that differ by no more than 1 edit"?

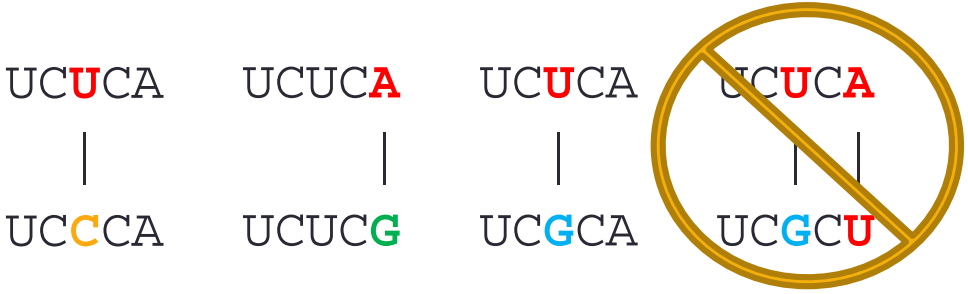
A: A function with domain *S* and codomain *S*

B: A function with domain *S* and codomain *P(S)*

C: A function with domain *S x S* and codomain *{T, F}*

D: A set that is a subset of $S \times S$

E: All of the above



UCUCA → UCCCA

UCUCA → UCACA

can map ≥ 2

UCUCA
1
UCCCA

this pair diff

≤ 1 edit distance

the fn maps a RNA string to a subset of RNA strings.

$f: S \longrightarrow P(S)$

set of all subsets of S.

Each element $X \in P(S)$ as a string of 0 & 1

$X = (0\ 1\ 0\ 1\ 1 \ldots\ldots)$

s

1 denotes whether $s \in X$ or not where $s \in S$

ex: $S = \{$ UAUUC, UUC, AAUUC, UAUAC ... $\}$

RNA string s = UAUUC

$f(s') = X$ → set of RNA strings

$(0\ 1\ 0\ 1\ 1 \ldots)$

s

every RNA $s \in X$ is 1 edit distance away from s!

s' is 1 edit distance away from s.

$S = \{AB, A, BB, B, BA\}$

$X \in P(S)$ $X = \{AB, A\}$

$X = (1\ 1\ 0\ 0\ 0)$

from

Application: genomics

$within1_{TF} : SXS \rightarrow \{T, F\}$

$within1_{TF}(s_1, s_2) =$ T if s_1 & s_2 are within dis 1

$within1_{\mathcal{P}} : S \rightarrow \mathcal{P}(S)$

$within1_{\mathcal{P}}(s_1) =$ set of RNAs s_1 in S s.t s_1 & s_2 are within dis 1.

Let the **binary relation** W_1 be the set of all pairs of RNA strands that are within one edit of one another.

$s_1 \, W_1 \, s_2$

R relation

aRb

What relations have you seen so far?

$a = b$

$a \leq b$

$a \geq b$

$a \mid b$

$a \equiv b \pmod{n}$

$\geq = \{(a,b) \in \mathbb{Z} \times \mathbb{Z} \mid a \text{ is more than } b\}$

$R_{\mod n}$ is a relation on $\mathbb{Z}$

$R_{\mod n} = \{(a,b) \in \mathbb{Z} \times \mathbb{Z} \mid a \bmod n = b \bmod n\}$

$a \; R_{\mod n} \; b$

Binary Relation on Sets :

$A = \{\mathbb{N}, \{1\}, \mathbb{Z}, \{2\}\}$

$A = B$

$|A| = |B|$

$A \subseteq B$

$A \supseteq B$

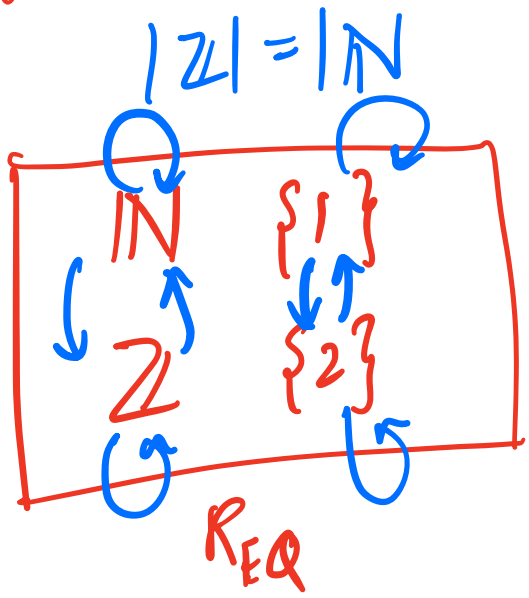
$A \subset B$

$R_{EQ} = \{(x,y) \in A \times A \mid |x| = |y|\}$

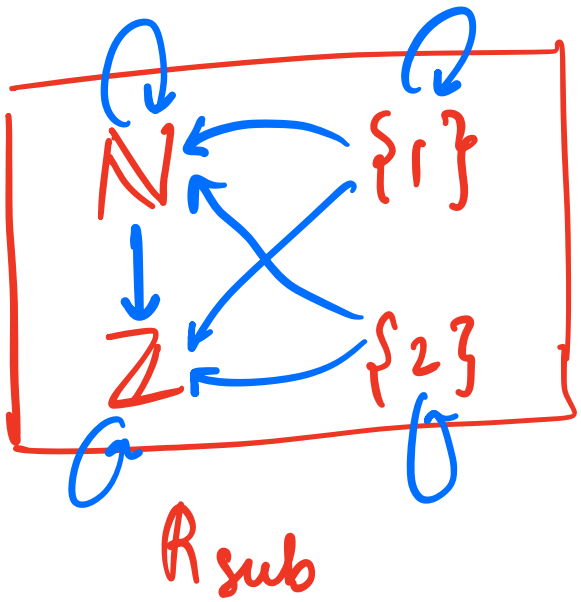
$a \; R_{EQ} \; b$

$R_{sub} = \{(X,Y) \in A \times A \mid X \subseteq Y\}$

Diagrams: ⟶ want to use diagrams to express relationships.



R_{EQ}



R_{sub}

aRa ⟲ a

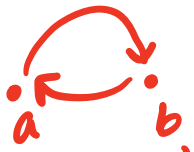
aRb $a \longrightarrow b$

Prop of Binary Relations:

→ Reflexive $\forall a \in A \quad aRa$

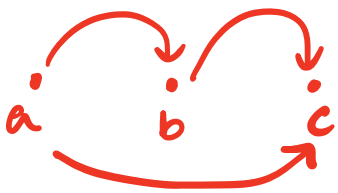
(self loops)

→ Symmetric $\forall a \in A \forall b \in B \quad aRb \rightarrow bRa$



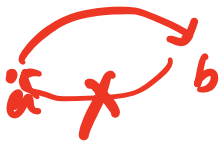
(paired arrows)

→ Transitive $\forall a \in A \forall b \in B \forall c \in C \quad aRb \wedge bRc \rightarrow aRc$



→ (chain collapse)

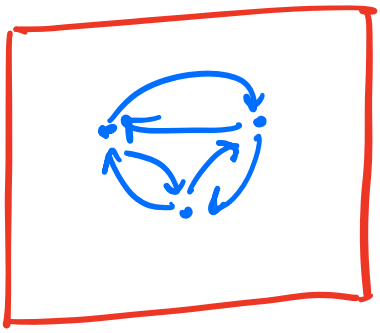
→ Anti Symmetric $\forall a \in A \forall b \in B \quad a \neq b \rightarrow \sim(aRb \wedge bRa)$



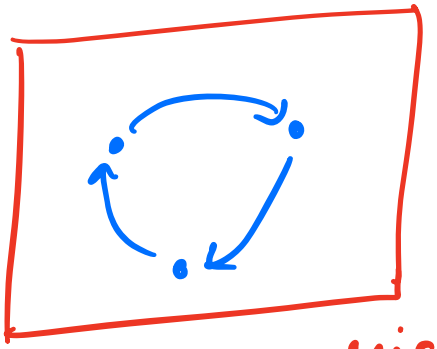
or

$aRb \rightarrow \sim(bRa)$

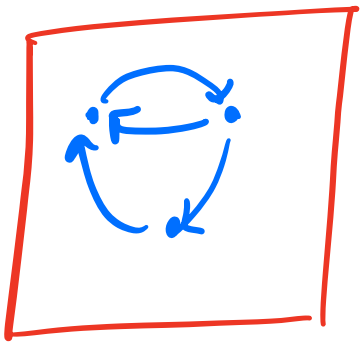
$\wedge \quad bRa \rightarrow \sim(aRb)$



symmetric

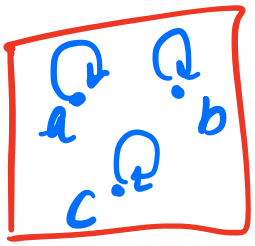


anti symmetric



not sym

& not antisym



both sym & both antisym

An Important Integer Relation(ship)

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

$R_{(\mathbf{mod}\ n)}$ is a relation on the set _____

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

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

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

Example Relations on sets

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_{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, $R_{SUB} =$ _____.

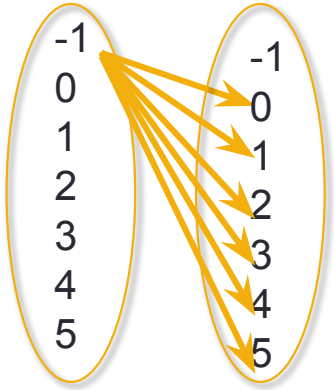
Visualizing relation R from A to B

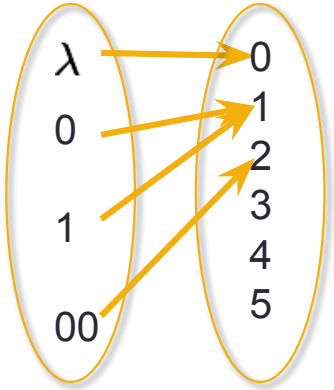
Examples

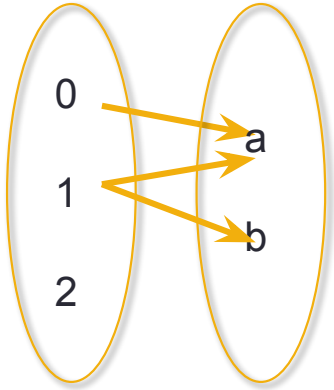
A = B = **Z** R={(x,y) : x < y}

A = {0,1}* B=**N** R={(w, n) : |w|=n}

A = {0,1,2} B={a,b} R={(0,a), (1,a), (1,b)}

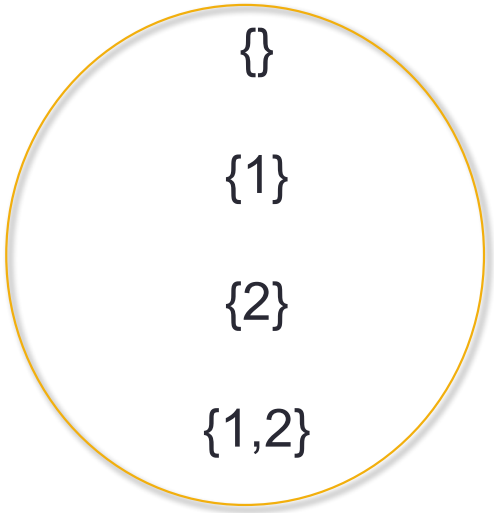


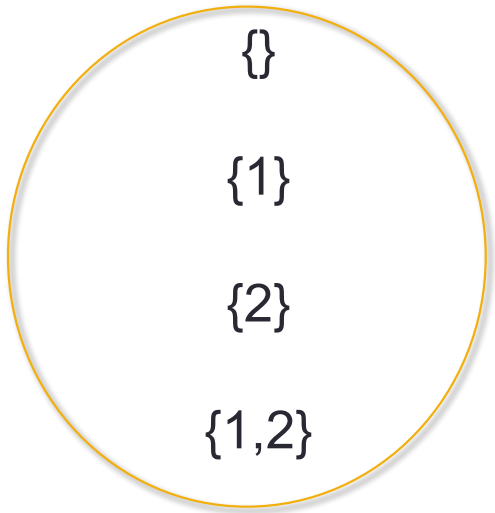




Visualizing relations using arrow diagrams

$$A = \mathcal{P}(\{1, 2\})$$





Properties of relations, R on A *zybook 10.2*

Reflexive, symmetric, transitive , antisymmetric

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**.

Which of these relations are reflexive?

A. W_1 → RNAs dis 1 from each other

B. $R_{(\text{mod } 4)}$ → $a \bmod 4 = b \bmod 4$

C. R_{EQ} → cardinalities are equal

D. R_{SUB} → subset

E. All of the above

Which of these relations are symmetric?

A. W_1
B. $R_{(\text{mod } 4)}$
C. R_{EQ}
D. R_{SUB}
E. All of the above

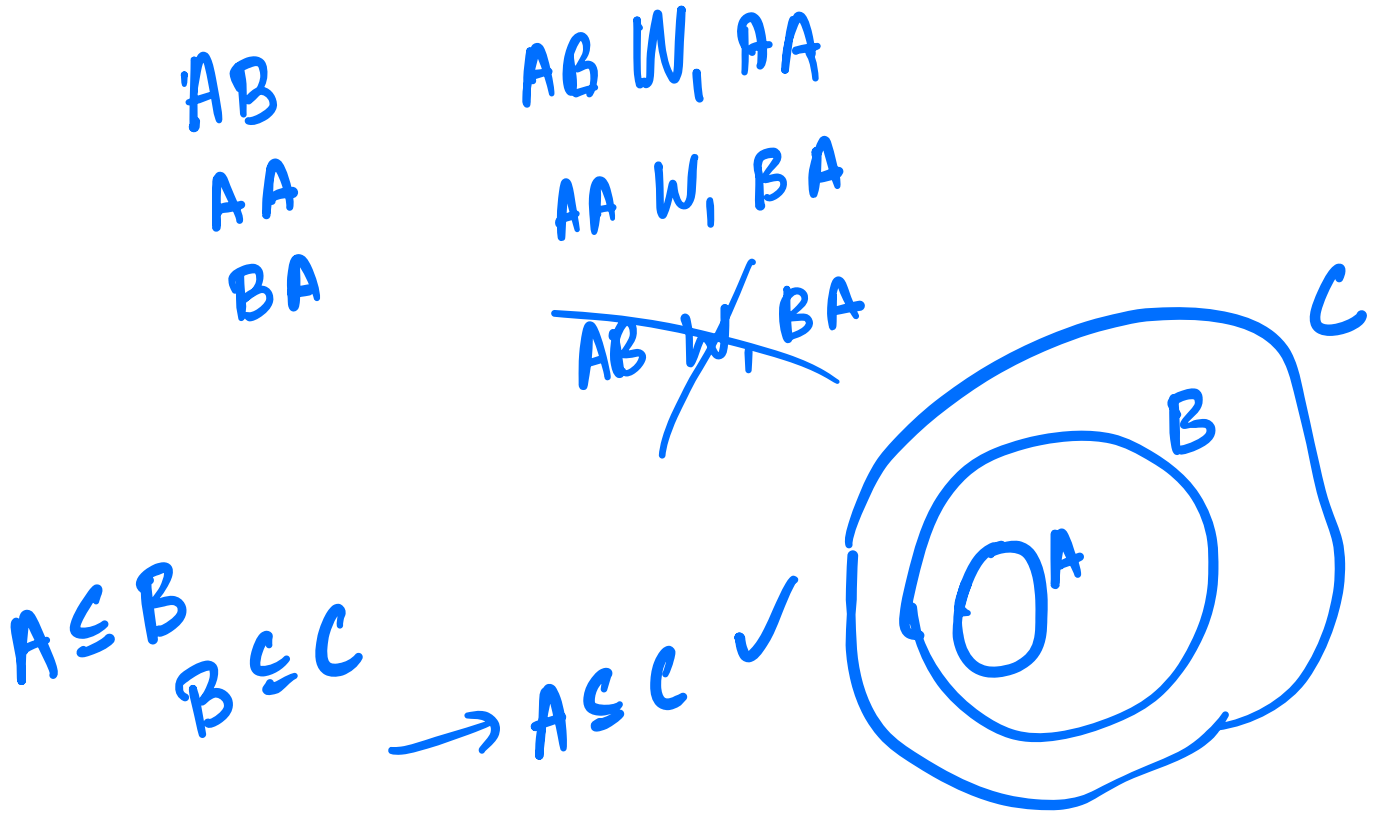
VUCG
VUAG

VUAG W_1 VUCG

VUCG W_1 VUAG

Which of these relations are transitive?

A. W_1
B. $R_{(\text{mod } 4)}$
C. R_{EQ}
D. R_{SUB}
E. All of the above



Which of these relations are antisymmetric?

A. W_1
B. $R_{(\text{mod } 4)}$
C. R_{EQ}
D. R_{SUB}
E. All of the above

$A \neq B \rightarrow \neg(A \subseteq B \wedge B \subseteq A)$

Classifying relations

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

Reflexive, Symmetric & Transitive

(*Zybook 10.3*)A relation is a **partial ordering** (or partial order) means it is

Reflexive, Anti Symmetric & Transitive

Relation on a set A, more generally

Example $A = \mathcal{P}(\{1,2\})$ $\qquad X \; R \; Y$ iff $X \subseteq Y$

Which of the following properties hold for R?

A. Reflexive, i.e. $\forall a(\ (a,a) \in R\ )$

B. Symmetric, i.e. $\forall a \forall b(\ (a,b) \in R \rightarrow (b,a) \in R\ )$

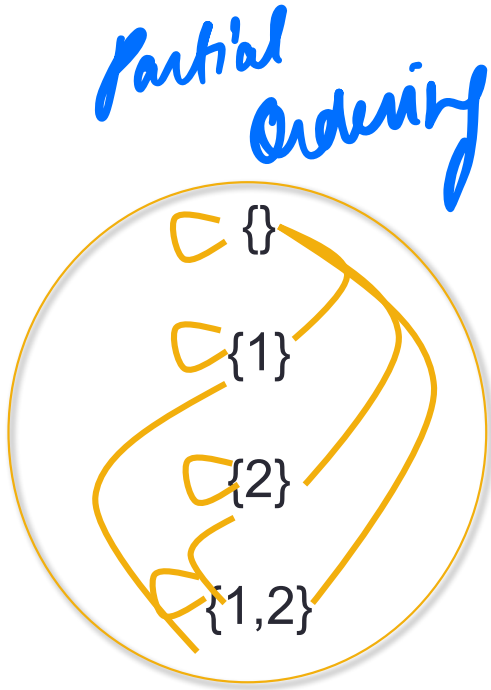
C. Antisymmetric, i.e.

$$\forall a \forall b(\ [(a,b) \in R \wedge (b,a) \in R] \rightarrow a = b\ )$$

D. Transitive, i.e.

$$\forall a \forall b \forall c(\ [(a,b) \in R \wedge (b,c) \in R] \rightarrow (a,c) \in R\ )$$

E. None of the above.

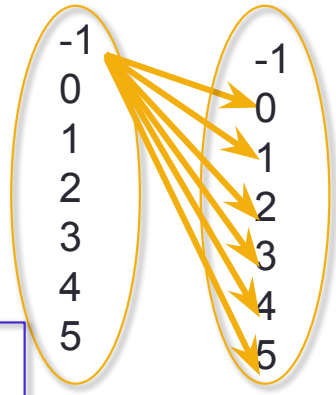


Relation on a set A, more generally

Rosen

Chapter 9

Example **Z** R={(x,y) : x < y}



Which of the following properties hold for R?

A. Reflexive, i.e. $\forall a((a,a) \in R)$
B. Symmetric, i.e. $\forall a \forall b((a,b) \in R \rightarrow (b,a) \in R)$
C. Antisymmetric, i.e. $\forall a \forall b([(a,b) \in R \wedge (b,a) \in R] \rightarrow a = b)$
D. Transitive, i.e. $\forall a \forall b \forall c([(a,b) \in R \wedge (b,c) \in R] \rightarrow (a,c) \in R)$
E. None of the above.

Partial order relations ($\preccurlyeq$) *Rosen Sec 9.6*

- A relation R on set A is a **partial ordering** iff it is **reflexive**, **antisymmetric**, and **transitive**.

Examples on the set of integers: "less than or equal", "greater than or equal"

Example on the set of positive integers: divisibility

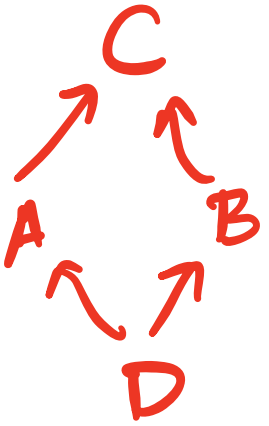
Example on power set of a set: subset inclusion

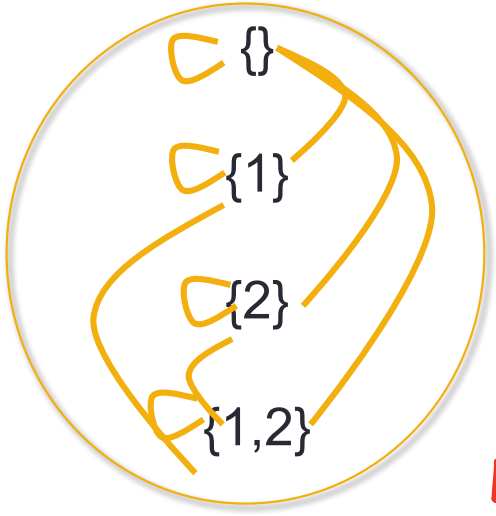
Example on set of binary strings: "is a prefix of"

Hasse Diagram

Graph of a partial order where

1. Don't include self-loops.
2. Don't include arcs that are implied by transitivity.
3. Use location instead of arrowheads: all edges point up.

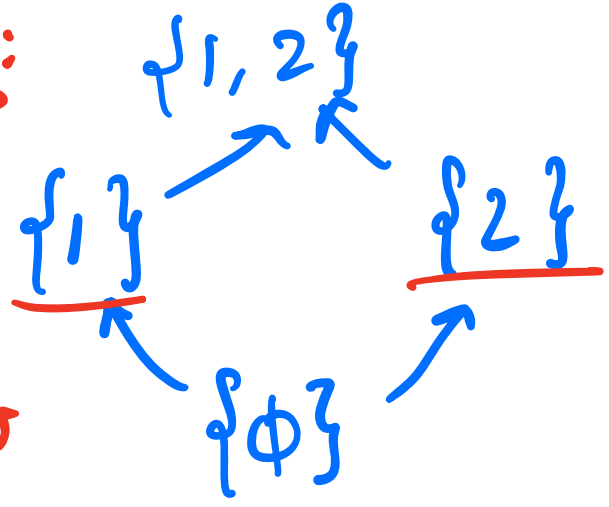




incomparable:

{1} ∦ {2}

not related to



P(S)

R = ⊆

Application: genomics

An idea important to genomics – are two RNA (or DNA) strands "close" ? For today – consider all RNA strands that are no more than "1 edit" away from a strand. An "edit" means applying one of the procedures *mutation* or *insertion* or *deletion.*

Mut with domain $S \times S$ is defined by, for $s_1 \in S$ and $s_2 \in S$,

$$Mut(s_1, s_2) = \exists k \in \mathbb{Z}^+ \exists b \in B(\ mutation(s_1, k, b) = s_2\ )$$

Ins with domain $S \times S$ is defined by, for $s_1 \in S$ and $s_2 \in S$,

$$Ins(s_1, s_2) = \exists k \in \mathbb{Z}^+ \exists b \in B(\ insertion(s_1, k, b) = s_2\ )$$

Del with domain $S \times S$ is defined by, for $s_1 \in S$ and $s_2 \in S$,

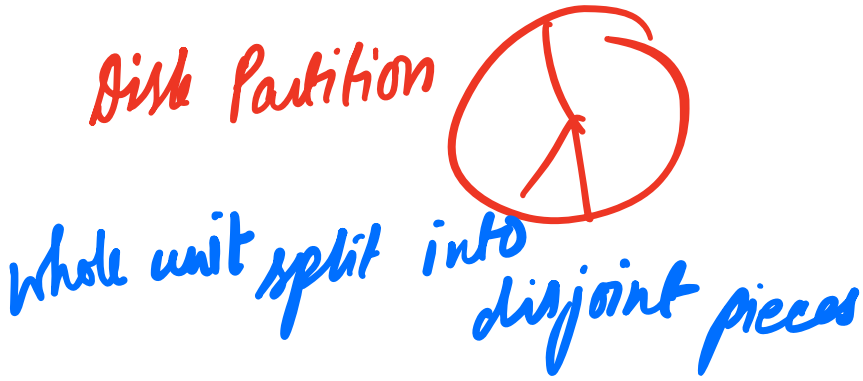
$$Del(s_1, s_2) = \exists k \in \mathbb{Z}^+(\ deletion(s_1, k) = s_2\ )$$

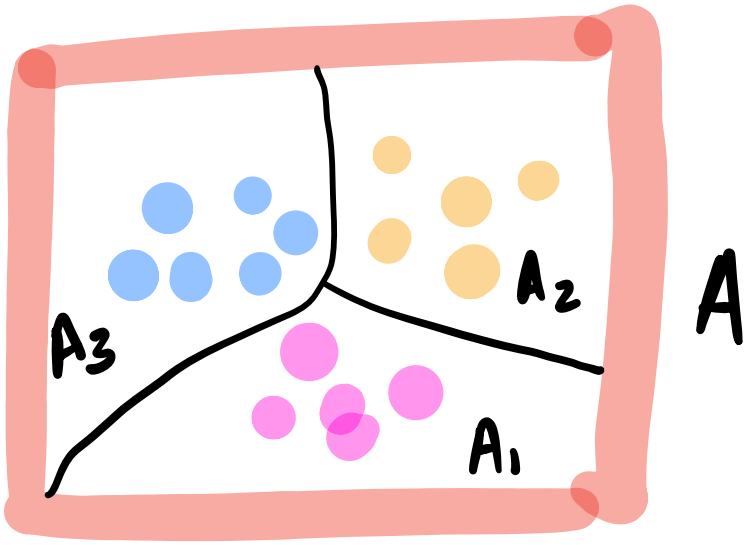
Definition: We say that a RNA strand s_1 is "within one edit" of a RNA strand s_2 to mean

$$Mut(s_1, s_2) \vee Mut(s_2, s_1) \vee Ins(s_1, s_2) \vee Ins(s_2, s_1) \vee Del(s_1, s_2) \vee Del(s_2, s_1)$$

Equivalence classes partition a set

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





$\forall A_i, \quad A_i \neq \phi$

$\forall A_i, A_j, i \neq j \quad A_i \cap A_j = \phi$

$\bigcup A_i = A$

Equivalence classes

$[5]_{R_{(\mathbf{mod}\ 4)}}$

$[9]_{R_{(\mathbf{mod}\ 4)}}$

$[6]_{R_{(\mathbf{mod}\ 4)}}$

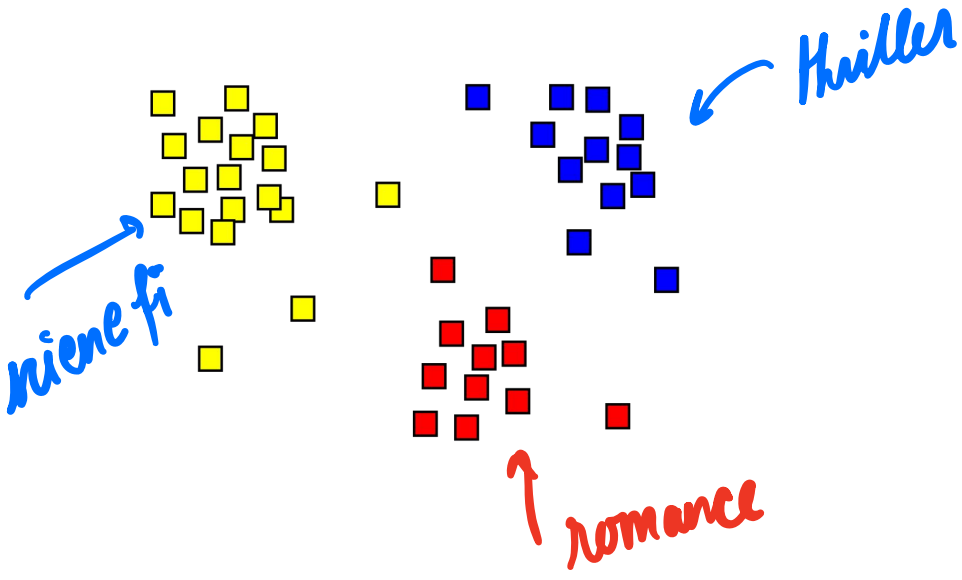
Clustering

Post links for motivation

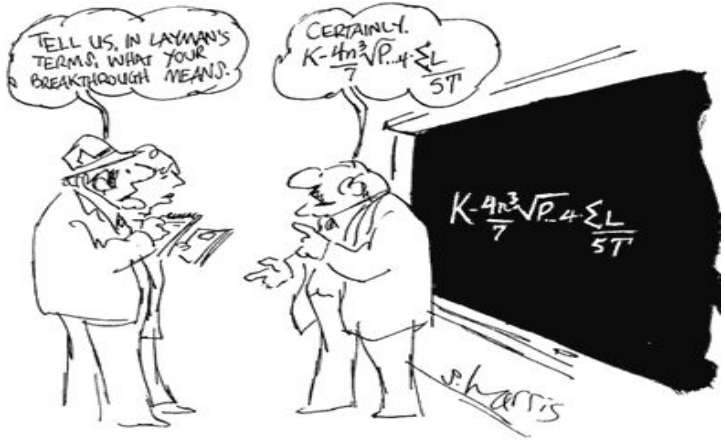
NETFLIX

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, with the following constraints:

area of research



Multiple Representations



Representing user ratings on NETFLIX

Person	Fyre	Frozen II	Picard	Ratings written as a 3-tuple
P_1	✗	•	✓	$(-1, 0, 1)$
P_2	✓	✓	✗	$(1, 1, -1)$
P_3	✓	✓	✓	$(1, 1, 1)$
P_4	•	✗	✓	$(0, -1, 1)$

Multiple Representations

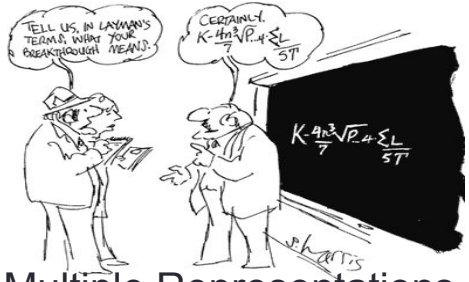
✗ Did not like: represent with -1

• No preference: represent with 0

✓ Liked: represent with 1

Representing user ratings or NETFLIX

Person	Fyre	Frozen II	Picard	Ratings written as a 3-tuple
P_1	✗	•	✓	$(-1, 0, 1)$
P_2	✓	✓	✗	$(1, 1, -1)$
P_3	✓	✓	✓	$(1, 1, 1)$
P_4	•	✗	✓	$(0, -1, 1)$

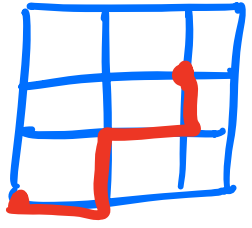


Multiple Representations

Definition: The set of movie ratings over n movies is R_n, where each element of R_n is a n-tuple with each entry in the tuple one of $\{-1, 0, 1\}$. The distance between two ratings is defined by d:

$$d(\ (x_1, \ldots, x_n), (y_1, \ldots, y_n)\ ) = \sum_{1 \le i \le n} |x_i - y_i|$$

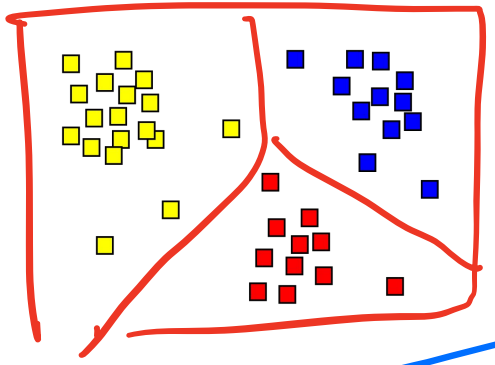
Manhattan



✗ Did not like: represent with -1
• No preference: represent with 0
✓ Liked: represent with 1

Idea: Partition the set of users

$U = \{r_1, r_2, \cdots, r_t\}$ is a set of user ratings, and $U \subseteq R_5$. Assume that each user represented by an element of U has a unique ratings tuple. A candidate clustering is $C_1, \cdots, C_m$ that is a **partition** of U: set of non-empty, disjoint subsets of U whose union equals U. We compare candidate clusterings by computing a metric, e.g. min cluster density or average cluster density, where density relates number of ratings in a cluster with the maximum distance between them.



cluster

A partition P is a set (family) $P \subseteq \mathcal{P}(U)$ s.t:

(i) $\forall C \in P \quad C \neq \emptyset$ → clusters are non empty

(ii) $\forall x \in U \quad \exists C \ (x \in C)$ → every element ∈ cluster

(iii) $\forall C_1 \in P \ \forall C_2 \in P \quad C_1 \neq C_2 \rightarrow C_1 \cap C_2 \neq \emptyset$

→ No two diff clusters ∩.

Equivalence classes

Definition: A binary relation E on U is an **equivalence relation** means it is reflexive, symmetric, and transitive.

$\forall x \in U($ xEx $)$, $\forall x \in U\, \forall y \in U\, ($ $xEy \to yEx$ $)$, and $\forall x \in U\, \forall y \in U\, \forall z \in U\, ($ $xEy \wedge yEz \to xEz$ $)$

An **equivalence class** of an element $x \in U$ for an equivalence relation E on the set U is the set

Equivalence class:

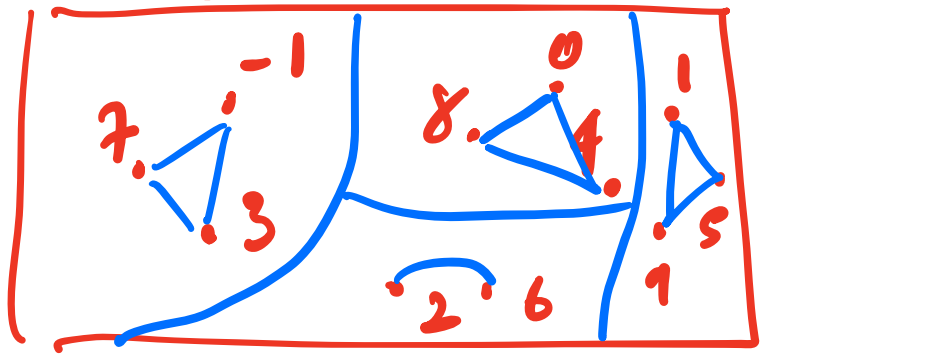
$$[a_E] = \{ s \in U \mid sEa \}$$

$a \in U$ → $[-1\ R\ (\mathrm{mod}\ 4)] = \{-1, 3, 7\}$

mod 4

$[3_E] = \{-1, 3, 7\}$

$E = R \bmod 4$ on $\{-1, \dots, 9\}$



Equivalence classes:

$[-1]_{R\,(\mathrm{mod}\,4)} = \{-1, 3, 7\}$

$[3]_{R\,(\mathrm{mod}\,4)} = \{-1, 3, 7\}$

$[2]_{R\,(\mathrm{mod}\,4)} = \{2, 6\}$

$\vdots$

$\{[1]_E, [0]_E, [1]_E, [2]_E, [3]_E, \ldots, [9]_E\}$

$= \{[-1], [0], [2], [1]\}$

↓ defines a partition of U

↓ each of these are equivalence classes.

Theorem: Given a equivalence relation E on set U, $\mathcal{F} = \{ [x]_E \mid x \in U \}$ is a partition of U

$[x]_E$ → equivalence classes

Proof: Each point in the definition of a partition.

✓ (i) $\forall C \in \mathcal{F}, \quad C \neq \phi$

Why is $\forall x, [x]_E \neq \phi$

Recall $[x]_E = \{ y \in U \mid y E x \}$

Since $x E x \quad \therefore \quad x \in [x]_E$

$\therefore [x]_E \neq \phi$

(ii) $\forall x \in U, \exists C \in \tilde{F}$ $x \in C$

Let x be an arbitrary element in U,

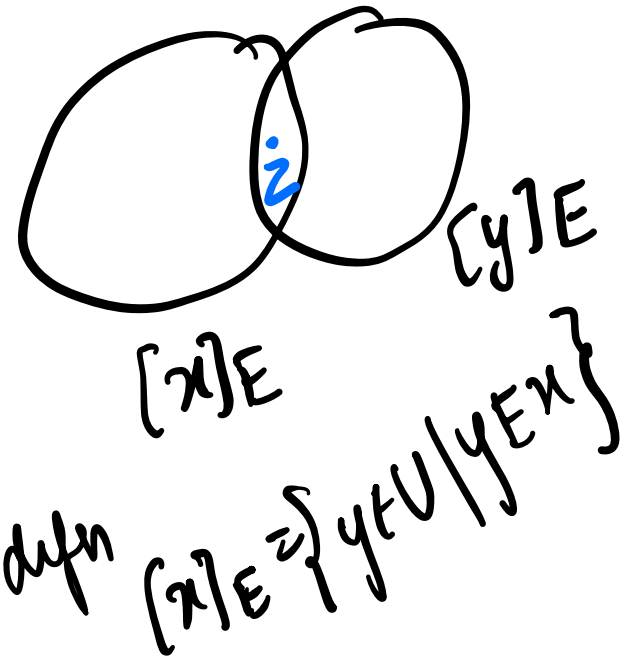
we know that

$x E x \Rightarrow x \in [x]_E$ & $[x]_E \in \tilde{F}$

Therefore, $[x]_E$ is a witness for x.

(iii) $\forall [x]_E \in \tilde{F}$ $\forall [y]_E \in \tilde{F}$ $[x]_E \neq [y]_E$

$\longrightarrow [x]_E \cap [y]_E \neq \phi$



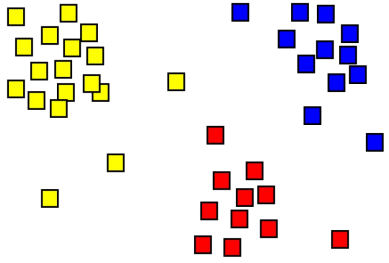
Proof by contrapositive

defn $[x]_E = \{y \in U \mid y E x\}$

Idea: Partitions from relations

$U = \{r_1, r_2, \cdots, r_t\}$ is a set of user ratings, and $U \subseteq R_5$. Assume that each user represented by an element of U has a unique ratings tuple. A candidate clustering is $C_1, \cdots, C_m$ that is a **partition** of U: set of non-empty, disjoint subsets of U whose union equals U. We compare candidate clusterings by computing a metric, e.g. min cluster density or average cluster density, where density relates number of ratings in a cluster with the maximum distance between them.

from first class we used sim metric



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

Set of nonempty disjoint subsets of U whose union is U

Which of these is a partition of $\{1,2,3,4\}$

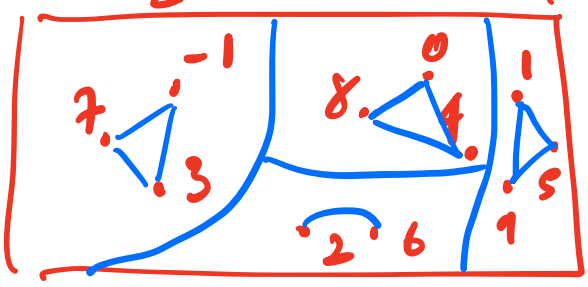
A) $P_1 = \{\{1,2\},\{3,2\},\{4\}\}$ ← not disjoint

B) $P_2 = \{\emptyset, \{1,2,3\},\{4\}\}$ ✗

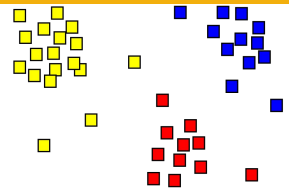
✓ C) $P_3 = \{\{1,2\},\{3,4\}\}$

D) $P_4 = \{\{1,5\},\{2\},\{3,4\}\}$ ← extra

$E = R \bmod 4$ on $\{-1,\dots,9\}$



Partitions from relations



Definition: A binary relation E on U is an **equivalence relation** means it is reflexive, symmetric, and transitive.

$\forall x \in U($ xEx $)$, $\forall x \in U\, \forall y \in U\, ($ $xEy \to yEx$ $)$, and $\forall x \in U\, \forall y \in U\, \forall z \in U\, ($ $xEy \wedge yEz \to xEz$ $)$

An **equivalence class** of an element $x \in U$ for an equivalence relation E on the set U is the set

$$[x]_E = \{s \in U | (x, s) \in E\}$$

↳ another way to say sEx

The set of equivalence classes of E is $\{[x]_E \mid x \in U\}$.

A. $\{[x]_E \mid x \in U\} \in U$

B. $\{[x]_E \mid x \in U\} \subseteq U$ ✗ $[x]_E \subseteq U$

C. $\{[x]_E \mid x \in U\} \in \mathcal{P}(U)$ ✗ $[x]_E \in \mathcal{P}(U)$

D. $\{[x]_E \mid x \in U\} \subseteq \mathcal{P}(U)$ ✓

E. None of the above.

$U = \{-1, \ldots, 9\}$

$\{[x]_E \mid x \in U\}$

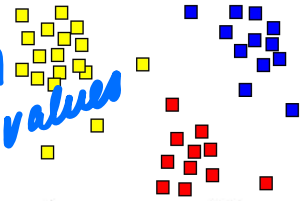
$= \{[-1], [0], [2], [1]\}$

$= \{\{-1, 3, 7\}, \{8, 0, 4\}, \{2, 6\}, \{1, 5, 9\}\}$

Relations on U

Which of these are eq. relations

words: two tuples are related if they agree on at least 3 values



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

$E_{dist} = \{(u, v) \in U \times U \mid d(u, v) \leq 2\}$ Not transitive

$E_{circ} = \{(u, v) \in U \times U \mid d((0,0,0,0,0), u) = d((0,0,0,0,0), v)\}$ distance from origin is equal

$$d((x_1, \ldots, x_n), (y_1, \ldots, y_n)) = \sum_{1 \leq i \leq n} |x_i - y_i|$$

Which of these options is true?

A. $((0, 1, -1, 0, 1), (1, 1, -1, 0, 0)) \in E_{proj}$

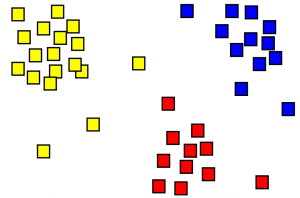
B. $((0, 1, -1, 0, 1), (1, 1, -1, 0, 0)) \in E_{dist}$ diff in 2 bits

C. $((0, 1, -1, 0, 1), (1, 1, -1, 0, 0)) \in E_{circ}$

D. More than one of the above.

E. None of the above.

Relations on U



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

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

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

Which of these relations is **not** an equivalence relation?

A. E_{proj}

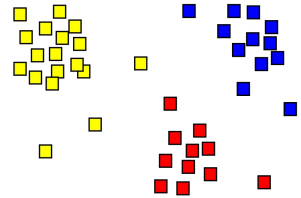
B. E_{dist}

C. E_{circ}

D. More than one of the above.

E. None of the above.

Relations on U



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

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

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

The partition of U defined by _____ is:

(i) partition of U defined by E_{proj} → partition by groups of first 3 values

(ii) " " E_{circ} → partition by distance from origin

Simplicity consider $U \in (\{0,1\})^5$

$[00000]_{E_{proj}} \in U$

00100
01000
01100
10000

$[00000]_{E_{proj}} = \{(00000), (00001), (00010), (00011)\}$

$[00000]_{E_{proj}} = [00001]_{E_{proj}}$

Relations on U

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

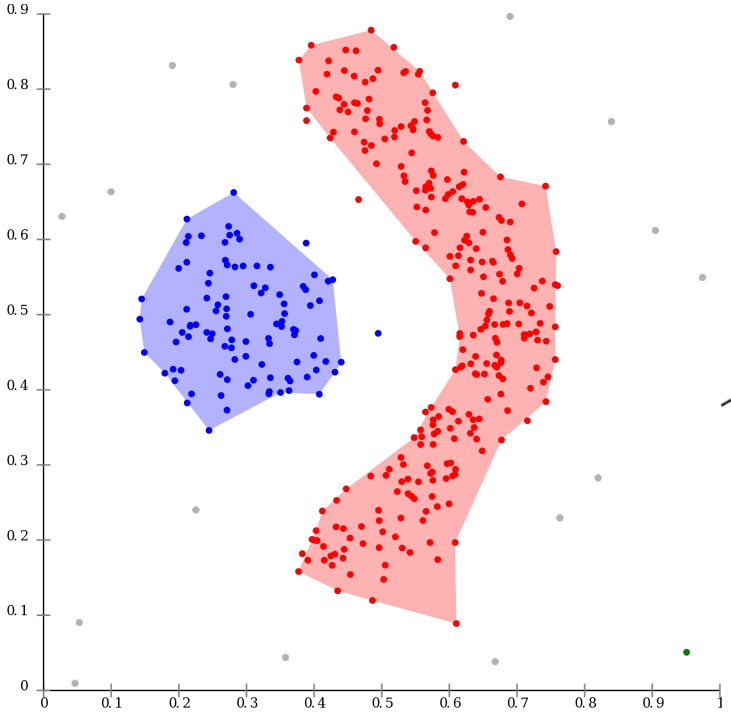
$$E_{dist} = \{(u, v) \in U \times U \ \mid \ d(u, v) \leq 2\}$$

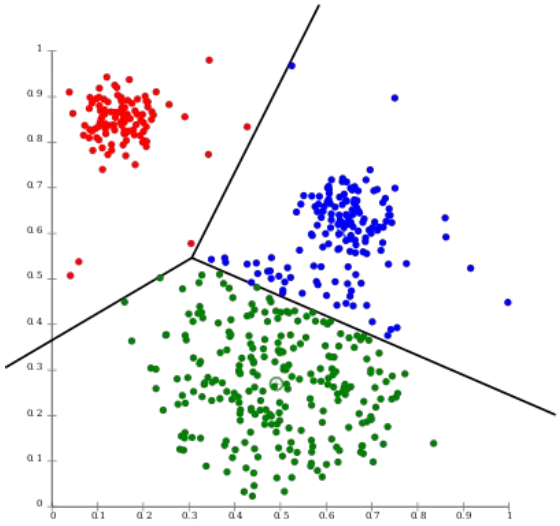
$$E_{circ} = \{(u, v) \in U \times U \ \mid \ d((0, 0, 0, 0, 0), u) = d((0, 0, 0, 0, 0), v)\}$$

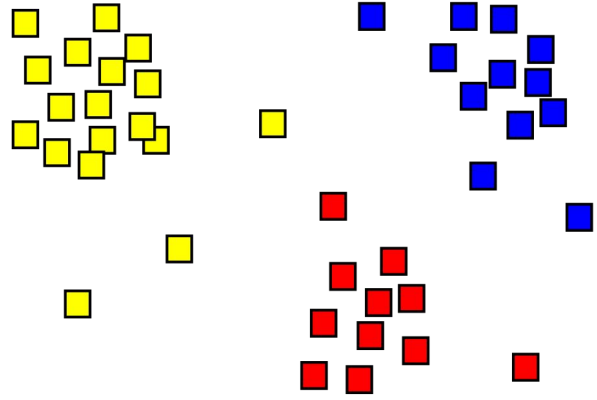
The partition of U defined by _____ is:

Generating Clusters (Efficiently)

https://en.wikipedia.org/wiki/K-means_clustering
https://en.wikipedia.org/wiki/Hierarchical_clustering
Clustering is a big, active research area!







What would you like us to review?

Application (Concept)

A. Numbers (algorithms, sets, base expansions, proof strategies)
B. Movie Ratings (symbolic statements, relations and partitions)
C. RNA strands/Linked lists (functions, recursion and induction)
D. Sets & Cardinality (functions and proof strategies)
E. Something else

Main takeaway

- Equivalence relations let us group “similar” elements. Generalize the idea of equality to many other application domains

- Equivalence classes partition sets (similar elements are in the same group)

Modular exponentiation

Rosen p. 254

Calculate $3^8 \bmod 7$

Approach 1: Directly

$3^1 \bmod 7 =$

$3^2 \bmod 7 =$

$3^3 \bmod 7 =$

$3^4 \bmod 7 =$

$3^5 \bmod 7 =$

$3^6 \bmod 7 =$

$3^7 \bmod 7 =$

$3^8 \bmod 7 =$

Modular exponentiation

Rosen p. 254

Calculate 3^8 **mod** 7

Modular Exponentation; Algorithm 5 in Section 4.2 (page 254)

```
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}
```

Approach 2: Using Algorithm 5

$b =$ ____, $n =$ _____, $k =$ ____, $m =$ ____

i	a_i	x	*power*
		1	b **mod** $m =$
0			
1			
2			
3			

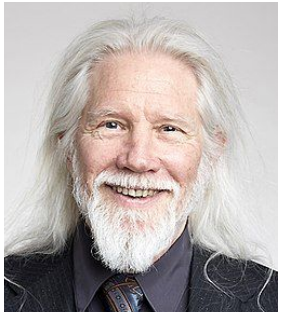
Application: Cryptography

How do parties communicate private key securely across insecure channel?

- Caesar cipher: need to agree on shift
- AES: private key, needs to be generated for each session

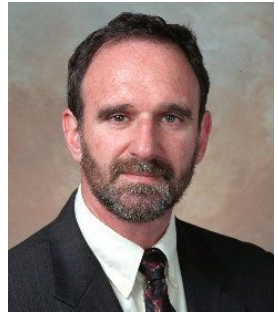
Application: Cryptography

Rosen, 4.6, p302



a: public

p: public



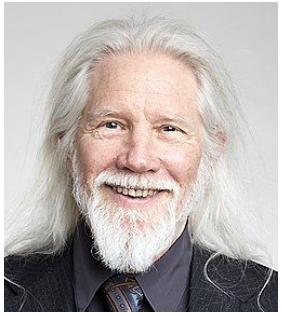
k1: secret

k2: secret

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 $(a^{k_1 \cdot k_2} \textbf{ mod } p)$.

Application: Cryptography

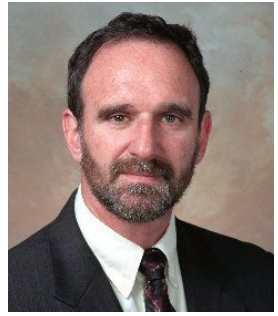
Rosen, 4.6, p302



k1: secret

a: public

p: public



k2: secret

Suppose that Alice and Bob want to share a common key. The protocol follows these steps, where the computations are done in $\mathbf{Z}_p$.

(1) Alice and Bob agree to use a prime p and a primitive root a of p.

(2) Alice chooses a secret integer k_1 and sends a^{k_1} **mod** p to Bob.

(3) Bob chooses a secret integer k_2 and sends a^{k_2} **mod** p to Alice.

(4) Alice computes $(a^{k_2})^{k_1}$ **mod** p.

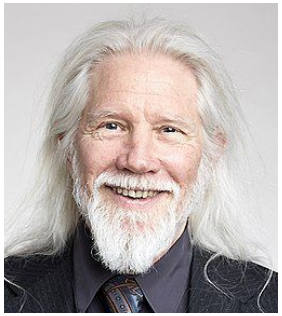
(5) Bob computes $(a^{k_1})^{k_2}$ **mod** p.

At the end of this protocol, Alice and Bob have computed their shared key, namely

$$(a^{k_2})^{k_1} \bmod p = (a^{k_1})^{k_2} \bmod p.$$

Application: Cryptography

Rosen, 4.6, p302



a: public

p: public

k1: secret

k2: secret

key cryptosystem they need to share a common key. The protocol we will describe is known as the **Diffie-Hellman key agreement protocol**, after Whitfield Diffie and Martin Hellman, who described it in 1976. However, this protocol was invented in 1974 by Malcolm Williamson in secret work at the British GCHQ. It was not until 1997 that his discovery was made public.

Application: Cryptography

Multiplication is computationally cheap, but
Factoring is computationally expensive
Given a 400 digit number that is a product of two 200 digit primes, can't efficiently find these primes. *Basis of security of RSA*

Modular exponentiation is computationally cheap, but
Discrete logarithm is computationally expensive:
Given a 300 digit prime and the result of exponentiation mod this prime, can't efficiently find the logarithm, i.e. find k when given a^k mod p
Basis of security of Diffie-Hellman

Peter Shor 1994 https://ieeexplore.ieee.org/document/365700

Review

- Review and practice with key concepts and examples from the quarter

What would you like us to cover first?

Application (Concept)

A. Numbers (algorithms, sets, base expansions, proof strategies)
B. Movie Ratings (symbolic statements, relations and partitions)
C. RNA strands (sets and definitions)
D. Linked lists (functions and recursion and induction)
E. Sets & Cardinality (functions and proof strategies)

Numbers: algorithms

```
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}
```

$b =$ ____, $n =$ _____, $k =$ ____, $m =$ ____

i	a_i	x	$power$
		1	b **mod** $m =$
0			
1			
2			
3			

Numbers: sets

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

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

, which is similar to the previous statement but about intersection. Which of the following statements is logically equivalent to its **negation?** (Select all and only that apply.)

(a) $\forall A \in W\ \forall B \in W\ \forall C \in W\ \big((A \cap B = A \cap C)\ \wedge\ (B \neq C)\big)$

(b) $\neg\forall A \in W\ \forall B \in W\ \forall C \in W\ \neg\big((A \cap B = A \cap C)\ \wedge\ (B \neq C)\big)$

(c) $\exists A \in W\ \exists B \in W\ \exists C \in W\ \big((A \cap B = A \cap C)\ \rightarrow\ \neg(B = C)\big)$

(d) $\exists A \in W\ \exists B \in W\ \exists C \in W\ \big((A \cap B = A \cap C)\ \wedge\ \neg(B = C)\big)$

Numbers: Base expansions

Convert $(2A)_{16}$ to …

A. binary (base ?)
B. decimal (base ?)
C. octal (base ?)
D. ternary (base ?)
E. All of the above

Numbers: Every integer n >= 2 is a product of primes

Ratings

In this question, we will translate statements about movie ratings using a **new** notion of distance between pairs of n-tuples where n is a positive integer:

$$d_3(\ (x_1, \ldots, x_n), (y_1, \ldots, y_n)\ ) = \sum_{i=1}^{n} |x_i - y_i|$$

In class, we discussed the application of n-tuples to movie recommendations. 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 $\{-1, 0, 1\}$.

Assume there are five movies in the database, so that each user's ratings can be represented as a 5-tuple. We use the following definition:

- R is the set of all ratings, that is, the set of all 5-tuples where each component of the 5-tuple is an element of the collection $\{-1, 0, 1\}$.

Fill in the blank and translate:

(1) The minimum distance between any two distinct ratings is _____

(2) The maximum distance between any two distinct ratings is _____

Ratings

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

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

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

$$d(\ (x_1, \ldots, x_n), (y_1, \ldots, y_n)\ ) = \sum_{1 \leq i \leq n} |x_i - y_i|$$

Prove that ____ is / is not _____ (reflexive / symmetric / transitive)

RNA strands

Each of the sets below is described using set builder notation. Rewrite them using the roster method. For example, the set described in set builder notation as

$$\{s \in S \mid \text{the leftmost base in } s \text{ is } \texttt{A} \text{ and } s \text{ has length } 2\}$$

is described using the roster method by

$$\{\texttt{AA}, \texttt{AC}, \texttt{AG}, \texttt{AU}\}.$$

Justifications aren't required for credit for this question, but it's good practice to think about how you would explain why your answer is correct.

(a) $\{s \in S \mid s \text{ has length } 2\}$

(b) $\{s \in S \mid \text{the leftmost base in } s \text{ is the same as the rightmost base in } s \text{ and } s \text{ has length } 3\}$

(c) $\{s \in S \mid \text{the bases in } s \text{ appear in alphabetical order}\}$[1]

(d) $\{s \in S \mid \text{there are twice as many } \texttt{A}\text{s as } \texttt{C}\text{s in } s \text{ and } s \text{ has length } 1\}$

RNA strands

The bases of RNA are elements of the set $B = \{\texttt{A}, \texttt{C}, \texttt{G}, \texttt{U}\}$. Certain sequences of bases serve important biological functions in translating RNA to proteins. The following recursive definition gives a special set of RNA strands that share certain biochemical properties.

Definition This set of RNA strands $\hat{S}$ is defined (recursively) by:

Basis Step:	$\texttt{AUG} \in \hat{S}$
Recursive Step:	If $s \in \hat{S}$ and $x \in R$, then $sx \in \hat{S}$

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

Each of the sets below is described using set builder notation. Rewrite them using the roster method. Justifications aren't required for credit for this question, but it's good practice to think about how you would explain why your answer is correct.

(a) $\{s \in \hat{S} \mid s \text{ has length less than or equal to } 5\}$

(b) $\{s \in \hat{S} \mid \text{there are twice as many } \texttt{C}\text{s as } \texttt{A}\text{s in } s \text{ and } s \text{ has length } 6\}$

Linked lists

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 $(n, l) \in L$

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

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

(a) Prove or disprove that the function *length* is onto.

(b) Prove or disprove that the function *length* is one-to-one.

Sets & Cardinality

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

A. If A is infinite then B is finite.
B. If A is countable then B is countable.
C. If B is infinite then A is finite.
D. If B is uncountable then A is uncountable.
E. None of the above.

Diagonalization: $|\mathbb{N}| \neq |\mathcal{P}(\mathbb{N})|$