Relations

Binary Relations

Ordered Pair -- (x,y)

- $(x, y) = (x^*, y^*)$ means that $x = x^*$ and $y = y^*$
- $(x,y) \neq (y,x)$ -- order matters

Cross Product

- $A \times B = \{(a,b) \mid (a \in A) \wedge (b \in B)\}$

Binary Relation

- Any subset R of $A \times B$ is called a *binary relation* on A,B.
- $(a, b) \in R \Leftrightarrow aRb$

Examples

1. $R = \{(a,b) \mid a < b\}$

2. $R = \{(students, courses)\}$

3. $R = \{(f,g) \mid f = O(g)\}$

4. $R = \{(A,B) \mid |A| = |B|\}$

5. Functions: $R = \left\{ \left(a, f(a) \right) \mid a \in A, f(a) \in f(A) \right\}$

    -- All Functions are Relations

    -- NOT All Relations are Functions

Representations

1. Tables

2. Graphs

3. Matrices

Directed Graphs

Analogy

- Graphs $\approx$ Relations
- Directed graphs are pictures of relations
- $uRv \Leftrightarrow$ there is an edge from u to v

Bipartite Graphs

- All edges go from set of vertices A to disjoint set of vertices B
- $R \subset A \times B$ -- general edge relation

Edge Relations

- Contain only topological -- yes/no -- information.
- No other data except connectivity.

<u>**Number of Relations**</u>

Finite Sets

$|A| = m$ and $|B| = n$

$\Rightarrow |A \times B| = |A||B| = mn$

$\Rightarrow$ # relations on $A \times B$ = # subsets of $A \times B = 2^{mn}$

<u>N-Ary Relations</u>

- $R \subset A_1 \times A_2 \times \cdots \times A_n$

- Table = Relational Data Base

- Projections -- Delete some columns

- Joins -- Combine overlapping tables

Relational Data Base

Student	*Homework*	*Midterm*	*Final*	*Grade*
Lydia	90	85	95	A–
Joe	80	85	90	B
Ron	60	45	50	F
Dan	95	98	100	A+
Sally	70	65	75	C

Most Important Relations

1. Equivalence Relations (on $A \times A$)

2. Transitive Closure

3. Partial Order

<u>Equivalence Relations</u>

Properties

1. Reflexive -- aRa

2. Symmetric -- $aRb \Rightarrow bRa$

3. Transitive -- $aRb \ \ and \ \ bRc \Rightarrow aRc$

Examples of Equivalence Relations

1. Rice undergraduates in the same college.

2. People of same height.

3. Computers with same amount of memory.

4. Programs that compute the same function.

5. Horses of the same color

6. Sets of the same cardinality.

7. Propositions that are logically equivalent.

8. Functions in the same complexity class.

Examples of Relations that are NOT Equivalence Relations

1. a is the father of b -- not reflexive, not symmetric

2. a is the brother of b -- not symmetric (sisters)

3. a has at least one parent in common with b -- not transitive

4. $f = O(g)$ -- not symmetric

Reflexive and Symmetric Representations

1. Graphs

2. Matrices

Equivalence Classes for Equivalence Relations

Equivalence Classes

- $[a] = \{x \mid aRx\}$

Properties

- $[a] = [b] \Leftrightarrow aRb$

- $[a] \cap [b] = \phi$ otherwise

Partitions

Definition

1. $A = \cup_{i \in I} A_i$

   -- every element of A lies in some A_i

2. $A_i \cap A_j = \phi \qquad i \neq j$

   -- no element of A lies in more than one A_i

Equivalence Relations $\Leftrightarrow$ Partitions

Theorem: Equivalence Relations $\Leftrightarrow$ *Partitions*

Proof:

$\Rightarrow$: Let $A_a = [a]$.

Then by the properties of equivalence classes the sets A_a form a partition of A.

$\Leftarrow$: Let $\{A_i\}$ be a partition of A, and define

$$aRb \Leftrightarrow \exists i \ a,b \in A_i .$$

Then it is easy to check that R is reflexive, symmetric, and transitive, so R is an equivalence relation.

QED

Functions on Equivalence Classes

Subtlety

- Let $f([a]) = g(a)$

- To show f is well-defined, must show that

  $aRb \Rightarrow g(a) = g(b)$

3. If $aRb \Rightarrow g(a) = g(b)$, then we say that

   g respects equivalence classes

Example

Rice Undergraduates

- $aRb \Leftrightarrow a \text{ and } b \text{ are in the same college}$

Functions

- $f\left([Mary]\right) = Mary's\ Last\ Name$

  -- f does not respect equivalence classes

- $f\left([Mary]\right) = Mary's\ College$

  -- f respects equivalence classes

Closures

Closure

- Smallest relation $S \supset R$ with property P

Reflexive Closure

- $S = R \cup \Delta$, where $\Delta = \{(a,a)\}$

Symmetric Closure

- $S = R \cup R^{-1}$, where $(b,a) \in R^{-1} \Leftrightarrow (a,b) \in R$

Transitive Closure

- $S = R^{*}$ (see next lecture)

Transitive Closure

<u>Composition</u>

Functions

- If $f : A \to B$ and $g : B \to C$, then $g \circ f : A \to C$
- $(g \circ f)(a) = g\big(f(a)\big)$

Relations

- If $R \subset A \times B$ and $S \subset B \times C$, then $S \circ R \subset A \times C$
- $a(S \circ R)c \Leftrightarrow \exists b \in B$ such that $a\,R\,b$ and $b\,S\,c$

Examples of Composition

Definitions

- $a\,R\,b$ means $b =$ parent of a
- $b\,S\,c$ means $b =$ sibling of c

Composition

- $a(S \circ R)c$ means

Examples of Composition

Definitions

- $a\,R\,b$ means b = parent of a
- $b\,S\,c$ means b = sibling of c

Composition

- $a(S \circ R)c$ means c = aunt/uncle of a
- $a(R \circ R)c$ means

Examples of Composition

Definitions

- $a R b$ means $b =$ parent of a
- $b S c$ means $b =$ sibling of c

Composition

- $a(S \circ R)c$ means $c =$ aunt/uncle of a
- $a(R \circ R)c$ means $c =$ grandparent of a
- $a(R^{-1} \circ S \circ R)c$ means

<u>Examples of Composition</u>

Definitions

- $a\,R\,b$ means b = parent of a
- $b\,S\,c$ means b = sibling of c

Composition

- $a(S \circ R)c$ means c = aunt/uncle of a
- $a(R \circ R)c$ means c = grandparent of a
- $a(R^{-1} \circ S \circ R)c$ means c = cousin of a

Composition and Matrix Multiplication

Notation

- M = Matrix for R
- N = Matrix for S

Composition

- $M * N$ = Matrix for $S \circ R$
- $*$ = Boolean Matrix Multiplication

  -- $+$ = or

  -- $\times$ = and

Powers and Closure

Powers of a Relation -- Recursive Definition

- $R^0 = I$ (Identity)
- $R^1 = R$
- $R^2 = R \circ R$
- $R^{n+1} = R \circ R^n = \underbrace{R \circ \cdots \circ R}_{n+1 \ \text{factors}}$

Explicit Definition

- $a\, R^n\, b \Leftrightarrow a = x_0\, R\, x_1\, R\, x_2 \cdots x_{n-1}\, R\, x_n = b$ (by induction on n)

Transitive Closures

- $R^+ = \cup_{k \geq 1} R^k$
- $R^* = \cup_{k \geq 0} R^k$

Closures

Transitive Closure

- R^+ = transitive closure of R
- $a\, R^+\, b \Leftrightarrow a = x_0\, R\, x_1\, R\, x_2 \cdots x_{n-1}\, R\, x_n = b \qquad n \geq 1$

Reflexive and Transitive Closure

- R^* = transitive closure of R
- $a\, R^*\, b \Leftrightarrow a = x_0\, R\, x_1\, R\, x_2 \cdots x_{n-1}\, R\, x_n = b \qquad n \geq 0$
- R^* is often called just the *transitive closure*

Observations

- $*$ means 0 or more
- $+$ means 1 or more
- R^* is reflexive
- R^+ need not be reflexive

Examples

1. $R = \{(a,b) \mid a \text{ is a parent of } b\}$

-- $R^+ = ?$

-- $R^* = ?$

2. $R = \{(a,b) \mid a \text{ shares a common border with } b\}$

-- $R^+ = ?$

-- $R^* = ?$

3. $R = \{(a,b) \mid \text{computer } a \text{ is connected to computer } b\}$

-- $R^+ = ?$

-- $R^* = ?$

4. $R = \{(a,b) \mid \text{instruction } a \text{ precedes instruction } b\}$

-- $R^+ = ?$

-- $R^* = ?$

More Examples

Graphs

- $\rightarrow$ means edge
- $\rightarrow^*$ means path

Trees

- $\rightarrow$ means child
- $\rightarrow^*$ means descendant

Computers

- $\Rightarrow$ means can get from one configuration (instantaneous description, snapshot) to another in one move (1 machine cycle, 1 instruction)
- $\Rightarrow^*$ means an entire computation

Closures

Matrix Definition

- M = Matrix for the relation R
- $R^{+} = \sum_{k \geq 1} M^{k}$
- $R^{*} = \sum_{k \geq 0} M^{k}$
- Matrix multiply and add = boolean multiply and add

Graph Definition

- $G = (V, E)$
  - -- $V = A$ (set on which R is defined)
  - -- $E = \{a \to b \mid a R b\}$
- $a R^{+} b \Leftrightarrow a \to x_1 \to \cdots \to x_n \to b \qquad n \geq 1$
- $a R^{*} b \Leftrightarrow a \to x_1 \to \cdots \to x_n \to b \qquad n \geq 0$

Simple Theorems on Transitivity

Theorem 1: *R is transitive if and only if $R \supset R^n$ for all $n \geq 1$.*

Proof: $\Rightarrow$: By induction on n.

$\Leftarrow$: $R \supset R^n \Rightarrow R \supset R^2 \Rightarrow R$ *transitive*

Theorem 2:

1. R^* is reflexive
2. R^+, R^* are transitive
3. $R^+, R^* \supset R$

Proof: Obvious from Definitions

Fundamental Theorem

Theorem 3: R^* is the smallest reflexive and transitive relation that contains R.

In particular, $R^* = \cap Q$, where the intersection is over all reflexive and transitive relations Q that contain R.

Proof: By Theorem 2, R^* is clearly a reflexive and transitive relation that contains R. Now suppose that Q is any reflexive and transitive relation that contains R. Then

$$a\,R^*\,b \Rightarrow a = x_0\,R\,x_1\,R\,x_2 \cdots x_{n-1}\,R\,x_n = b$$

$$\Rightarrow a = x_0\,Q\,x_1\,Q\,x_2 \cdots x_{n-1}\,Q\,x_n = b$$

$$\Rightarrow a\,Q\,b$$

because Q is reflexive and transitive.

Hence $Q \supset R^*$. QED

Relations on Finite Sets

Theorem 4: *Let $|A| = n$, and let R be a relation on A. If there is a path in R from a to b, then there is a path in R from a to b of length at most n ($n-1$ if $a \neq b$).*

Proof: Remove cycles. Pigeonhole principle.

Corollary: $|A| = n \Rightarrow R^* = R \cup R^2 \cup \cdots \cup R^n$

Partial Order

Orders

Partial Order

- Reflexive -- $a R a$
- Antisymmetric -- $a R b$ and $b R a \Rightarrow a = b$
- Transitive -- $a R b$ and $b R c \Rightarrow a R c$

Note: There may be elements that are NOT comparable

Total Order

- Partial order where every two elements are comparable

Well Order

- Total order where every nonempty subset has a smallest element
- Induction works only on well ordered sets
- Base case = smallest element

Examples

- $\{\mathbf{N}, \leq\}$

- $\{\mathbf{Z}, \leq\}$

- $\{P(S), \supset\}$

- $\{\mathbf{Z}^{+}, |\}$

- *Orders on* $N \times N$
  - -- Lexicographic -- $(a,b) < (c,d) \Leftrightarrow a < c \;\; or \;\; a = c \;\; and \;\; b < d$
  - -- Product -- $(a,b) < (c,d) \Leftrightarrow a < c \; and \;\; b < d$

- *Order on Strings* Σ^{*}
  - -- Lexicographic order = Dictionary order

- *Graphs and Trees*
  - -- Subgraphs and Subtrees

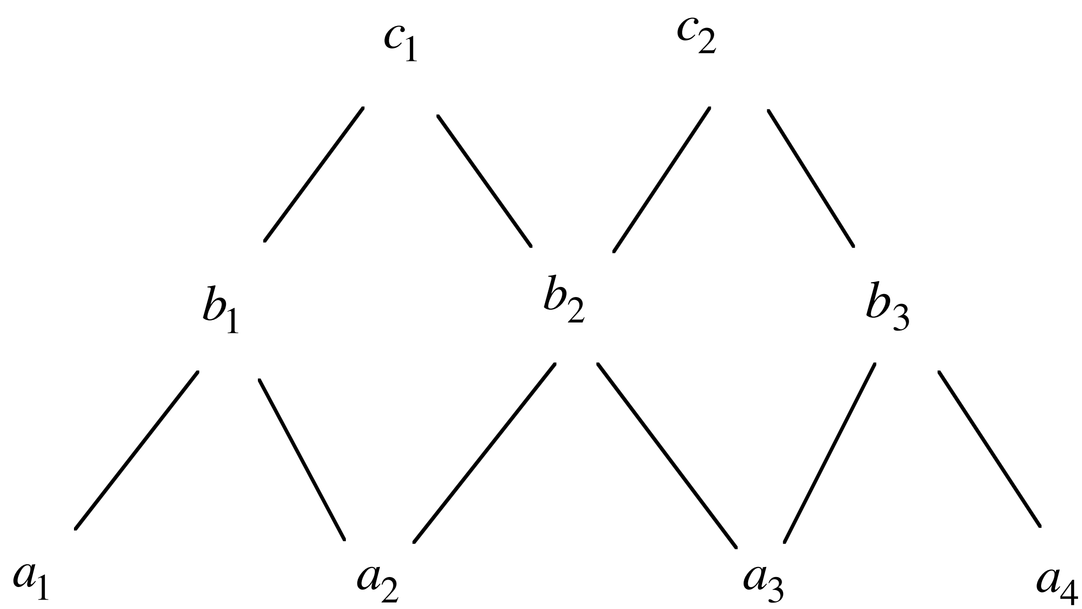
Hasse Diagrams

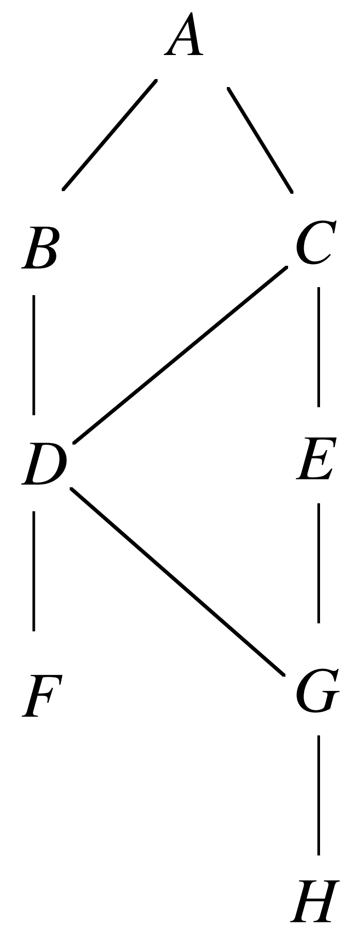
- Graphical representation of a poset

- Relation graph without reflexive and transitive edges

- See pictures

Hasse Diagrams





<u>Definitions</u>

Maximal and Minimal Elements

- in the set
- not unique

Greatest (Maximum) and Least (Minimum) Elements

- in the set
- unique

Upper and Lower Bounds

- not necessarily in the set
- not unique

Lub and Glb

- not necessarily in the set
- unique

Examples

Hasse Diagram

- Maximal and Minimal Elements
- Upper and Lower Bounds

$\{P(S), \supset\}$

- Greatest = S Least = ϕ

$\{Z, \leq\}$

- no greatest or least element

$\{(0,1), \leq\}$

- no greatest or least element

Lattices

Definition

- Poset where every pair has a lub and a glb

Examples

- Total Orders

- $\{\boldsymbol{Z}^{+}, |\}$

  -- $\text{glb}(a,b) = \text{gcd}(a,b)$

  -- $\text{lub}(a,b) = lcm(a,b)$

- $\{P(S), \supset\}$

  -- $\text{glb}(A,B) = A \cap B$

  -- $\text{lub}(A,B) = A \cup B$

- $\{(N \times N,\ Product\}$

  -- $\text{glb}\left\{(a,c), b,d)\right\} = \left(\min(a,c), \min(b,d)\right)$

  -- $\text{lub}\left\{(a,c), b,d)\right\} = \left(\max(a,c), \max(b,d)\right)$

Topological Sort

Purpose

- Convert a partial order into a total order

Applications

- Scheduling -- Engineering
- Hidden Surface Algorithms -- Computer Graphics

Topological Sort on Finite Sets

Lemma

- Every finite poset has a minimal element
- Proof by induction on $|S|$

Algorithm

- Choose a minimal element a of S
- Choose a minimal element b of $S - \{a\}$
- Continue until all elements of S are exhausted
- Rank elements in order chosen

Result of Topological Sort is NOT Unique!

- Examples -- Hasse diagrams

Scheduling Comp 280 for Spring 2011