6.1 Introduction to binary relations

A binary relation is a way of expressing a relationship between two sets. Mathematically, a ***binary relation*** between two sets A and B is a subset R of $A \times B$. The term *binary* refers to the fact that the relation is a subset of the Cartesian product of *two* sets. The two sets A and B may or may not be equal. For $a \in A$ and $b \in B$, the fact that $(a, b) \in R$ is denoted by aRb.

Suppose, for example, that S is the set of students at a university and C is the set of classes offered by the university. The relation E between $s \in S$ and $c \in C$ indicates whether a student is enrolled in a given class. Thus, sEc if student s is enrolled in class c. A student can be enrolled in more than one class and a class can have more than one student enrolled in it. If a student s is not currently taking any classes, then there is no c such that sEc. Similarly, if a course c is not currently offered by the university, then there is no s such that sEc.

Relations can also be defined on infinite sets. For example, we can define the relation C between $\mathbb{R}$ and $\mathbb{Z}$ to be:

$$xCy \text{ if } |x - y| \leq 1$$

That is, xCy if the distance between real number x and integer y is at most 1.

If two sets, A and B, are finite, then a binary relation R between A and B can be represented by a list of ordered pairs. A relation can also be specified in a more graphical way. In an ***arrow diagram*** of a relation R on sets A and B, the elements of A are listed on the left, the elements of B are listed on the right, and there is an arrow from $a \in A$ to $b \in B$ if aRb.

PARTICIPATION ACTIVITY 6.1.1: Arrow diagram for a relation.

Animation captions:

1. A relation A is defined on a set of people and a set of files. Person p is related to file f under A (pAf) if person p has access to file f.
2. An arrow diagram for A lists the people in a column on the left and the files in a column on the right.
3. Each pair (p, f) in the relation is represented by an arrow pointing from p to f.

A ***matrix representation*** of relation R between A and B is a rectangular array of numbers with |A| rows and |B| columns. Each row corresponds to an element of A and each column corresponds to an element of B. For a ∈ A and b ∈ B, there is a 1 in row a, column b, if aRb. Otherwise, there is a 0.

PARTICIPATION ACTIVITY 6.1.2: Matrix representation for a relation.

Animation captions:

1. A matrix representation for A has a row for each person and a column for each file. (Sue, File B) is represented by a 1 in the row for Sue and the column for File B.
2. Similarly, there is a 1 in the matrix for every pair in the relation.
3. All other entries in the matrix are 0.

PARTICIPATION ACTIVITY 6.1.3: Binary relations between two sets.

Let A = { 2, 3 }. Define the relation R between the set A and $\mathbf{Z}^+$, the set of positive integers:

aRx if x is an integer multiple of a.

1) 3R8
- True
- False

2) 4R8
- True
- False

3) There is an a ∈ A, such that aR7.
- True
- False

4) There is an x ∈ $\mathbf{Z}^+$, such that 2Rx and 3Rx.
- True
- False

5) 2R2
- True
- False

Example 6.1.1: Binary relations to represent resource requirements.

Consider a manufacturing plant that must satisfy an incoming stream of orders. Different orders require different machines in the plant. The requirements for orders can be represented by a relation between the set of all orders and the set of machines in the plant. Order o is related to machine m if o requires m. Representing the requirements as a relation would be a first step in finding an efficient schedule for the orders so that machines are not needed for more than one order at the same time.

It is possible to have a relation between two sets A and B in which A = B. A ***binary relation on a set*** A is a subset of A x A. The set A is called the ***domain*** of the binary relation.

PARTICIPATION ACTIVITY 6.1.4: Binary relations on a set.

Define the relation R on $\mathbf{Z}^+$, the set of positive integers, as follows:

aRb if a and b are relatively prime (that is, the only positive integer that evenly divides both a and b is 1).

1) 8R12
- True
- False

2) 21R20.
- True
- False

3) There is a number such that x ≠ 1 and xRx.
- True
- False

4) There is a number x such that xR7.
- True
- False

An arrow diagram for a relation R on a finite set A requires only one copy of the elements of A. There is an arrow from a to b if aRb. An element can have arrows heading into it and arrows that head out of it. An element that is related to itself is indicated by an arrow called a ***self-loop***. A self-loop leaves the element and then turns around to point to itself again.

PARTICIPATION ACTIVITY 6.1.5: Arrow diagram for a relation on a finite set.

Animation content:

undefined

Animation captions:

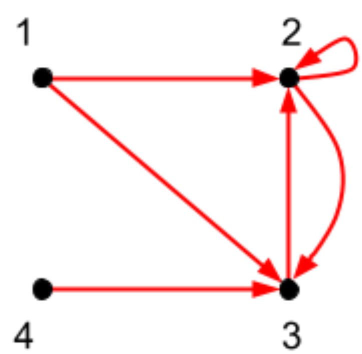
1. A pair (a,b) in the relation is represented by an arrow from a to b.
2. Every pair in the relation is represented by an arrow. The pair (d,d) is represented by a self-loop at d.

PARTICIPATION ACTIVITY 6.1.6: Representations of binary relations on a set.

Let A = {1, 2, 3, 4}. Define a relation R on A:

R = {(1, 2), (1, 3), (2, 2), (2, 3), (3, 4), (4, 3)}

An arrow diagram and matrix representation of the relation are supposed to be represented below, *but there are some mistakes*. Rows of the matrix are numbered 1 through 4 from top to bottom and columns are numbered 1 through 4 from left to right.



$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

1) Which pair is in the arrow diagram but not the original R? Give your answer in the form (a, b).

Check Show answer

2) Which pair is in the original R but not represented in the arrow diagram? Give your answer in the form (a, b).

Check Show answer

3) Which entry in the matrix is a 0 but should be a 1? Give your answer in the form (row-number, column-number). Note that the correct relation is given by the set R.

Check Show answer

4) Which entry in the matrix is a 1 but should be a 0? Give your answer in the form (row-number, column-number). Note that the correct relation is given by the set R.

Check Show answer

CHALLENGE ACTIVITY | 6.1.1: Introduction to binary relations.

460784.3106054.qx3zqy7

Start

Let $A = \{4, -4, 1, -3\}$. Define the relation R between the set A and $\mathbb{Z}$, the set of integers

xRy if $|x - y| \geq 2$

Select all pairs that are in the relation.

- [ ] $(-3, -4)$
- [ ] $(-4, 0)$
- [ ] $(4, 3)$
- [ ] $(1, 3)$

1 2 3

Check Next

Additional exercises

EXERCISE 6.1.1: Matrices to arrow diagrams and sets.

Each matrix below represents a relation. The rows and columns are numbered 1 through 3 or 4.

Give the arrow diagram for each matrix, then express each relation as a set of ordered pairs.

(a)
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Solution

(b)
$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Solution

(c)
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(d) $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

(e) $\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$

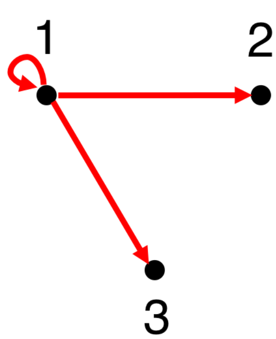
(f) $\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

Solution

EXERCISE 6.1.2: Arrow diagrams to matrices and sets.

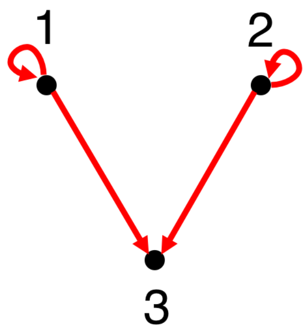
Give the matrix representation for the relation depicted in each arrow diagram. Then express the relation as a set of ordered pairs.

(a)



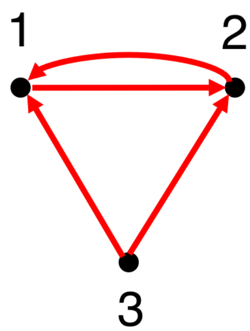
Solution

(b)

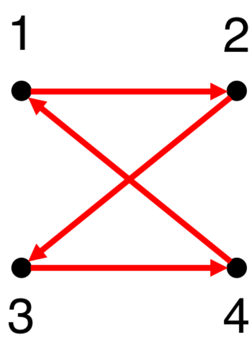


Solution

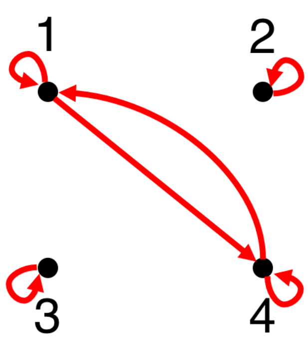
(c)



(d)

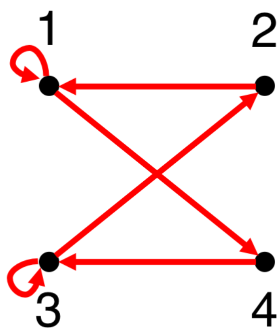


(e)



©zyBooks 02/18/23 02:08 1553027
Beca Zhou
UCII&CSCI6BMeenakshisundaramWinter2023

(f)



EXERCISE 6.1.3: Matrices and arrow diagrams for relations expressed as sets of pairs.

Draw the arrow diagram and the matrix representation for each relation.

(a) Define the set A = {r, o, t, p, c} and B = {proposition, math, proof, discrete}. Define the relation $R \subseteq A \times B$ such that (letter, word) is in the relation if that letter occurs somewhere in the word.

Solution

(b) The domain for relation R is {1, 2, 3, 4}. R = { (1, 2), (3, 4), (2, 3), (3, 2), (2, 1), (3, 1), (4, 3) }.

Solution

(c) The domain for relation R is {1, 2, 3, 4}. R = { (1, 2), (2, 1), (3, 3) }.

(d) The domain for relation R is {1, 2, 3}. R = ∅.

Solution

EXERCISE 6.1.4: Arrow diagrams for relations on small finite sets.

Draw the arrow diagram for each relation.

(a) The domain of relation C is $\{0, 1\}^3$. For x, y ∈ $\{0, 1\}^3$, xCy if y can be obtained from x by changing only one 0 to a 1.

(b) The domain of relation P is {2, 4, 8, 16, 32, 64}. For x, y in the domain, xPy if there is a positive integer n such that $x^n = y$.

(c) The domain of relation D is {2, 3, 12, 16, 27, 48}. For x, y in the domain, xDy if y is an integer multiple of x.

(d) The domain for the relation A is the set {2, 5, 7, 8, 11}. For x, y in the domain, xAy if |x-y| is less than 2

Solution

(e) The domain for the relation P is the set {2, 4, 8, 10, 16, 64}. For x, y in the domain, xPy if there is a positive integer n such that $x^n = y$

Solution

(f) The domain of relation H is a group of four friends. For x, y in the domain, xHy if y is at least as tall as x. The table below shows each person in the domain and her height.

Name	Height
Angie	5'0"
Bernice	5'3"
Carmen	5'3"
Deirdre	5'5"

Solution

(g) The domain of relation H is a group of four friends. For x, y in the domain, xHy if y is taller than x. The table below shows each person in the domain and her height.

Name	Height
Angie	5'0"
Bernice	5'3"
Carmen	5'3"
Deirdre	5'5"

6.2 Properties of binary relations

Reflexive and anti-reflexive binary relations

Suppose that R is a binary relation on set A. R is ***reflexive*** if and only if for every $x \in A$, xRx.

Notice that the definition of reflexive is a universal statement. In order for a binary relation to be reflexive, every element in the set must be related to itself. In order to show that a relation is not reflexive, it is only necessary to show that there is a particular $x \in A$ such that xRx is not true.

Suppose that the domain for a relation is a set of people. Person x is related to person y if x has the same biological mother as person y. This relation is reflexive because every person must have the same biological mother as himself or herself.

R is ***anti-reflexive*** if and only if for every x in the domain of R, it is not true that xRx. Irreflexive is an alternative term for anti-reflexive.

The definition of anti-reflexive is also a universal statement. In order for a binary relation to be anti-reflexive every element in the set must not be related to itself. In order to show that a relation is not anti-reflexive, it is only necessary to show that there is a particular x ∈ A such that xRx is true.

Suppose that the domain for a relation is a set of people. Person x is related to person y if x is taller than y. This relation is anti-reflexive because no person can be taller than himself or herself.

If some of the elements in the domain of R are related to themselves and some of the elements are not related to themselves, then R is neither reflexive nor anti-reflexive.

PARTICIPATION ACTIVITY | 6.2.1: Reflexive and anti-reflexive binary relations.

Animation content:

undefined

Animation captions:

1. A relation R on set A is reflexive if every element in A has a self-loop in the arrow diagram for A.
2. A relation R on set A is anti-reflexive if no element in A has a self-loop in the arrow diagram for A.
3. R is not anti-reflexive because aRa and bRb. R is not reflexive because it is not true that cRc, dRd, or eRe.

PARTICIPATION ACTIVITY | 6.2.2: Identifying reflexive and anti-reflexive binary relations.

The domain for each relation described below is the set of all positive real numbers. Select the correct description of the relations.

1) x is related to y if y = x+1.

- Reflexive
- Anti-reflexive
- Neither

2) x is related to y if y = 1/x.

- Reflexive

- Anti-reflexive
- Neither

3) x is related to y if $\lfloor x \rfloor \leq \lfloor y \rfloor$.

- Reflexive
- Anti-reflexive
- Neither

Symmetric binary relations

Suppose that R is a relation on set A. R is ***symmetric*** if and only if for every pair, x and y ∈ A, xRy if and only if yRx.

A relation is symmetric if for every pair of elements x and y in the domain, one of the following situations holds:

- xRy and yRx are both true
- Neither xRy nor yRx is true

The situation that is not allowed in a symmetric relation is for there to be a pair, x and y, such that x is related to y but y is not related to x.

Notice that the definition of symmetric is a universal statement. The criteria is that for every pair, the two elements in the pair are both related to each other or both not related to each other. If there is any pair of elements, x and y, where x is related to y and y is not related to x, then the relation is not symmetric.

Suppose that the domain for a relation is a set of people. Person x is related to person y if x has the same biological mother as person y. This relation is symmetric because x has the same biological mother as y if and only if y has the same biological mother as x. The situation that never happens is that x has the same biological mother as y but y does not have the same biological mother as x.

PARTICIPATION ACTIVITY | 6.2.3: Symmetric binary relations.

Animation content:

undefined

Animation captions:

1. A relation R on set A is symmetric if an arrow from x to y implies that there is an arrow from y to x.

2. The allowed patterns for a pair a $\neq$ b in a symmetric relation is to have aRb and bRa or to have neither aRb nor bRa.
3. Having aRb but not bRa (or vice versa) is not allowed in a symmetric relation.
4. This relation is not symmetric because e is related to a but a is not related to e.

PARTICIPATION ACTIVITY | 6.2.4: Identifying symmetric relations.

The domain for each relation described below is the set of all positive real numbers. Select the correct description of the relations.

1) x is related to y if x/y = 2.

- Symmetric
- Not symmetric

2) x is related to y if y = 1/x.

- Symmetric
- Not symmetric

Anti-symmetric binary relations

Suppose that R is a relation on set A. R is ***anti-symmetric*** if and only if for every pair, x and y ∈ A, if x ≠ y then it can not be the case that xRy and yRx are both true.

A relation is anti-symmetric if for every pair of distinct elements in the domain one of the following situations holds:

- xRy, but it is not true that yRx
- yRx, but it is not true that xRy
- Neither xRy nor yRx is true

The situation that is not allowed in an anti-symmetric relation is for there to be a pair, x and y, such that x ≠ y and xRy and yRx are both true. Notice that the definition of anti-symmetric is a universal statement. If any pair, x and y, in the domain have the forbidden pattern of x ≠ y, xRy and yRx, then the relation is not anti-symmetric. One way to show that a relation is anti-symmetric is to take an arbitrary pair of elements in the domain, x and y, and show that the assumptions xRy and yRx necessarily imply that x = y.

Suppose that the domain for a relation is a set of people. Person x is related to person y if x is taller than person y. This relation is anti-symmetric because it is impossible for there to be two different people, x and y, where x is taller than y and y is taller than x.

A relation is neither symmetric nor anti-symmetric if there is a pair of distinct elements that are related to each other and another pair of elements, x and y, where x is related to y but y is not related to x.

PARTICIPATION ACTIVITY 6.2.5: Anti-symmetric binary relations.

Animation content:

undefined

Animation captions:

1. A relation R on set A is anti-symmetric if an arrow from x to y implies that there is no arrow from y to x.
2. The allowed patterns for a pair, a and b, is for neither aRb nor bRa to be true, or for aRb to be true but not bRa, or for bRa to be true but not aRb.
3. The disallowed pattern is for there to be a pair, a and b, where $a \neq b$ and aRb and bRa are both true.
4. This relation is not anti-symmetric because bRc and cRb are both true.

PARTICIPATION ACTIVITY 6.2.6: Identifying anti-symmetric relations.

The domain for each relation described below is the set of all positive real numbers. Select the correct description of the relations.

1) x is related to y if $\lfloor x \rfloor \leq \lfloor y \rfloor$.

- Symmetric
- Anti-symmetric
- Neither

2) x is related to y if x+y = 1

- Symmetric
- Anti-symmetric
- Neither

3) x is related to y if y = x+1.

- Symmetric
- Anti-symmetric
- Neither

Transitive binary relations

Suppose that R is a relation on set A. R is ***transitive*** if and only if for every three elements, x, y, z ∈ A, if xRy and yRz, then it must also be the case that xRz. Note that in the definition of a transitive relation, the elements, x, y, and z do not necessarily have to be distinct.

The situation that is not allowed in a transitive relation is for there to be an x, y, and z , such that xRy and yRz are true but xRz is not true. Notice that the definition of transitive is a universal statement. If any x, y, and z in the domain have the forbidden pattern of xRy and yRz but not xRz, then the relation is not transitive. If there is no triple x, y, and z that has the forbidden pattern, then the relation is transitive.

Suppose that the domain for a relation is a set of people. Person x is related to person y if x is taller than person y. This relation is transitive because if x is taller than y and y is taller than z, then it must also be the case that x is taller than z.

PARTICIPATION ACTIVITY 6.2.7: Transitive binary relations.

Animation content:

undefined

Animation captions:

1. A relation R on set A is transitive if an arrow from e to a and an arrow a to b implies that there is an arrow from e to b.
2. The disallowed pattern in a transitive relation is to have elements, a, b, and c, where aRb and bRc are true, but aRc is not true.
3. This relation is not transitive because aRb and bRc are true, but aRc is not true.

PARTICIPATION ACTIVITY 6.2.8: Identifying transitive relations.

The domain for each relation described below is the set of all positive real numbers. Select the correct description of the relations.

1) x is related to y if y = x+1

- Transitive
- Not transitive

2) x is related to y if y ≥ x+1

- Transitive

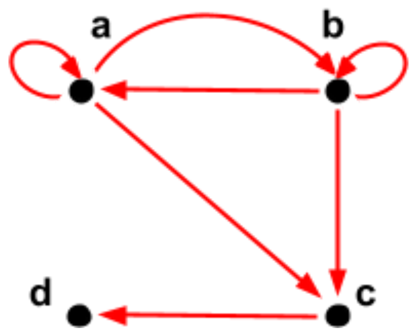
- Not transitive

3) x is related to y if $\lfloor x \rfloor \leq \lfloor y \rfloor$.

- Transitive
- Not transitive

PARTICIPATION ACTIVITY 6.2.9: Recognizing the properties of a relation from an arrow diagram.

The relation R on the set {a, b, c, d} is defined by the arrow diagram below.



1) Is the relation R reflexive, anti-reflexive or neither?

- Reflexive
- Anti-reflexive
- Neither

2) Is the relation R symmetric, anti-symmetric or neither?

- Symmetric
- Anti-symmetric
- Neither

3) Is the relation R transitive?

- Transitive
- Not transitive

PARTICIPATION ACTIVITY 6.2.10: Recognizing the properties of a relation from an arrow diagram.

The arrow diagram below defines a relation Q on the set {x, y}.

1) Is the relation Q transitive?

- Transitive
- Not transitive

Proving and disproving properties of binary relations

Each of the properties of a binary relation is stated as a universal condition. Therefore in order to establish that relation has a property, the condition must be shown to be true for all the elements in the domain. Establishing that a relation does not have a property only requires showing one counter-example: specific elements in the domain which do not satisfy the condition.

For example, consider a relation whose domain is the set of all 4-bit strings. Two strings are related if they have the same first bit or the same last bit or both. For example, 1001 is related to 0101 because the two strings have the same last bit. 1001 is not related to 0100 because neither the first bits nor the last bits of the two strings are the same.

PARTICIPATION ACTIVITY | 6.2.11: Justifying that a relation does or does not have a property.

Animation content:

undefined

Animation captions:

1. Relation R on 4-binary strings. Two strings are related if they have the same first or last bit. The relation R is reflexive: every binary string has the same first bit as itself.
2. R is symmetric because if x has the same first or last bit as y, then y has the same first or last bit as x.
3. R is not transitive. The justification is a counter-example. 1001 is related to 1000 and 1000 is related to 0000. However, 1001 is not related to 0000.

PARTICIPATION ACTIVITY | 6.2.12: Recognizing the properties of a relation: Integer multiples.

The domain of relation D is the set of positive integers. For x, y ∈ $\mathbf{Z}^+$, xDy if x evenly divides y. Positive integer x evenly divides positive integer y if there is another positive integer n such that y = xn.

1) Is the relation D reflexive, anti-reflexive or neither?

- Reflexive
- Anti-reflexive
- Neither

2) Is the relation D symmetric, anti-symmetric or neither?

- Symmetric
- Anti-symmetric
- Neither

3) Is the relation D transitive?

- Transitive
- Not transitive

PARTICIPATION ACTIVITY 6.2.13: Recognizing the properties of a relation: Relatively prime positive integers.

The domain of relation P is the set of positive integers. For x, y ∈ $\mathbf{Z}^+$, xPy if x and y are relatively prime. Two positive integers are relatively prime if the only integer that evenly divides both numbers is 1.

For example, 4 is not relatively prime to 10 because 2 evenly divides 4 and 10. 49 is relatively prime to 15 because 1 is the only positive integer that evenly divides 49 and 15.

1) Select the pair of numbers that xPy is true.

- x = 7 and y = 7
- x = 8 and y = 27
- x = 45 and y = 18

2) Is the relation P reflexive, anti-reflexive or neither?

- Reflexive
- Anti-reflexive
- Neither

3) Is the relation P symmetric, anti-symmetric or neither?

- Symmetric
- Anti-symmetric
- Neither

4) Select the triple of numbers that show that P is not transitive.

- 7, 14, 21
- 7, 11, 15
- 4, 7, 6

Additional exercises

EXERCISE | 6.2.1: Identifying properties of relations.

For each relation, indicate whether the relation is:

- reflexive, anti-reflexive, or neither
- symmetric, anti-symmetric, or neither
- transitive or not transitive

Justify your answer.

(a) The domain of the relation L is the set of all real numbers. For x, y ∈ **R**, xLy if x < y.

Solution

(b) The domain of the relation E is the set of all real numbers. For x, y ∈ **R**, xEy if x ≤ y.

Solution

(c) The domain of relation P is the set of all positive integers. For x, y ∈ $\mathbf{Z}^+$, xPy if there is a positive integer n such that $x^n = y$.

(d) The domain for the relation D is the set of all integers. For any two integers, x and y, xDy if x evenly divides y. An integer x evenly divides y if there is another integer n such that y = xn. (Note that the domain is the set of all integers, not just positive integers.)

(e) The domain for the relation A is the set of all real numbers. xAy if |x - y| ≤ 2.

Solution

(f) The domain for relation R is the set of all real numbers. xRy if x - y is rational. A real number r is rational if there are two integers a and b, such that b ≠ 0 and r = a/b. You can use the fact that the sum of two rational numbers is also rational.

(g) The domain for the relation is **Z**×**Z**. (a, b) is related to (c, d) if a ≤ c and b ≤ d.

Solution

(h) The domain for the relation is **Z**×**Z**. (a, b) is related to (c, d) if a ≤ c or b ≤ d (inclusive or).

(i) The domain for relation T is the set of real numbers. xTy if x + y = 0.

Solution

(j) The domain for relation Z is the set of real numbers. xZy if y = 2x.

(k) The domain for relation T is a group of people. xTy if person y is taller than person x. There are at least two people in the group who are not the same height.

Solution

(l) The domain for relation C is a group of people. xCy if person x is the first cousin of person y (i.e., a parent of person x is a sibling of a parent of person y). You can assume that there at least two people x and y such that x is the first cousin of y. You can also assume that no one has two parents who are siblings of each other.

EXERCISE 6.2.2: Properties of relations - dependence on the domain.

The players on a football team are all weighed on a scale. The scale rounds the weight of every player to the nearest pound. The number of pounds read off the scale for each player is called his measured weight. The domain for each of the following relations below is the set of players on the team. For each relation, indicate whether the relation is:

- reflexive, anti-reflexive, or neither
- symmetric, anti-symmetric, or neither
- transitive or not transitive

Justify your answer.

(a) Player x is related to player y if the measured weight of player x is at least the measured weight of player y. No two players on the team have the same measured weight.

Solution

(b) Player x is related to player y if the measured weight of player x is at least the measured weight of player y. There is at least one pair of players on the team who have the same measured weight. There is also at least one pair of players on the team who have different measured weights.

(c) Player x is related to player y if the measured weight of player x is at least the measured weight of player y. All the players on the team have exactly the same measured weight.

Solution

EXERCISE 6.2.3: Relations that are both reflexive and anti-reflexive or both symmetric and anti-symmetric.

(a) Is it possible to have a relation on the set {a, b, c} that is both reflexive and anti-reflexive? If so, give an example.

(b) Is it possible to have a relation on the set {a, b, c} that is both symmetric and anti-symmetric? If so, give an example.

(c) Is it possible to have a relation on the set {a, b, c} that is neither symmetric nor anti-symmetric? If so, give an example.

(d) Is it possible to have a relation on the set {a, b, c} that is both symmetric and transitive but not reflexive? If so, give an example.

Solution

EXERCISE 6.2.4: Identifying properties of relations - cont.

For each relation R, indicate if the relation is

- Reflexive, anti-reflexive, or neither
- Symmetric, anti-symmetric, or neither
- Transitive or not transitive

(a) The domain of the relation R is {a}. R = {(a, a)}

(b) The domain of the relation R is {a, b, c}. R = {(a, b), (b, c), (a, c)}

Solution

(c) The domain of the relation R is {a, b, c, d}. R = {(a, b), (b, a), (c, d), (d, c)}

(d) The domain of the relation R is {a, b}. R = {(a, b), (b, a), (a, a), (b, b)}

Solution

(e) The domain of the relation R is {a, b, c, d, e, f}. R = {(a, a), (a, b), (b, c), (c, d), (d, e), (e, f), (f, a)}

EXERCISE 6.2.5: Identifying properties of relations on a power set.

X = {a, b, c, d, e}, and P(X) is the power set of X. The domain of all of the relations defined below is P(X).

For each relation, indicate if the relation is

- Reflexive, anti-reflexive, or neither
- Symmetric, anti-symmetric, or neither
- Transitive or not transitive

(a) A is related to B if $|A - B| = 1$

(b) A is related to B if $A \cap B = \emptyset$

Solution

(c) A is related to B if $A \subset B$

Solution

(d) a is an element of X. A is related to B if $B = A \cup \{a\}$

6.3 Directed graphs, paths, and cycles

A ***directed graph*** (or ***digraph***, for short) consists of a pair (V, E). V is a set of ***vertices***, and E, a set of ***directed edges***, is a subset of V × V. An individual element of V is called a ***vertex***. A vertex is typically pictured as a dot or a circle labeled with the name of the vertex. An edge (u, v) ∈ E, is pictured as an arrow going from the vertex labeled u to the vertex labeled v. The vertex u is the ***tail*** of the edge (u, v) and vertex v is the ***head***. If the head and the tail of an edge are the same vertex, the edge is called a ***self-loop***. The graph in the animation below has a self-loop at vertex d.

The ***in-degree*** of a vertex is the number of edges pointing into it. The ***out-degree*** of a vertex is the number of edges pointing out of it.

$$\text{in-degree}(u) = |\{ v \mid (v, u) \in E \}|$$

$$\text{out-degree}(u) = |\{ v \mid (u, v) \in E \}|$$

PARTICIPATION ACTIVITY 6.3.1: Directed graphs.

Animation captions:

1. Each edge in a directed graph is drawn as an arrow. Edge (a,b) is drawn as an arrow from vertex a to vertex b. Edge (d,d) is a self-loop at d.
2. The tail of edge (a,b) is a and the head is b.
3. The in-degree of c is 2 because there are two edges pointing into c.
4. The out-degree of b is 1 because there is one edge pointing out of b.
5. The only edge pointing into d is the self-loop (d,d), so both the in-degree and out-degree of d are 1.

The definition of a digraph should sound familiar because a digraph is the same mathematically as a relation on the set V. The set of edges E is a subset of V × V and is therefore a relation on the set V. Using the notation of relations, uEv if and only if there is a directed edge from vertex u to vertex v. A picture of a digraph is the same as the arrow diagram for the relation E.

Directed graphs are used in many different applications in computer science. They can be used, for example, to represent communication links in a network, dependencies between tasks in a large computation, or the relationships between users in a social network.

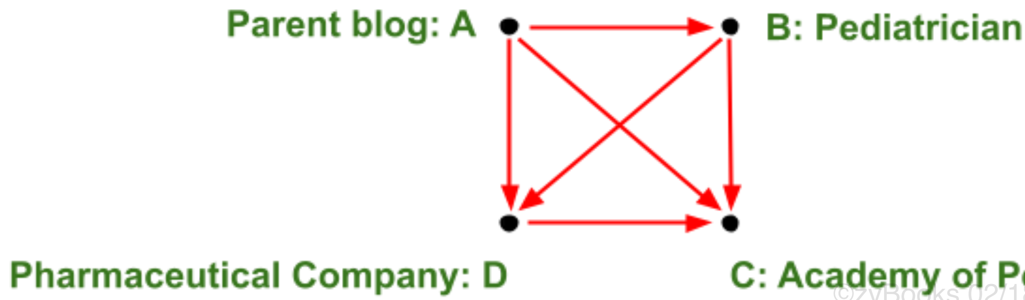
Example 6.3.1: The internet as a directed graph.

The internet can be viewed as a very large directed graph G = (V, E). The vertex set is the set of all urls. The hyperlink relationships can be expressed as directed edges: (u, v) ∊ E if there is a hyperlink from the page u to v. The structure of this graph is used to assist in a search. For example, a web page with a large in-degree is likely to be a more authoritative source and therefore should appear earlier in a list of hits returned by a search engine.

Consider for example a search on the keywords "children", "allergy", and "pollen". There are likely thousands of pages that include these three words. Which pages are likely to be of greatest interest to the user? Below is a sample of a few hits that might arise:

- A: A blog entry by a parent whose child has an allergy.
- B: An article written by a local pediatrician on children's allergies.
- C: An article on the web pages of the American Academy of Pediatrics.
- D: An article on the web pages of a pharmaceutical company selling allergy medication for children.

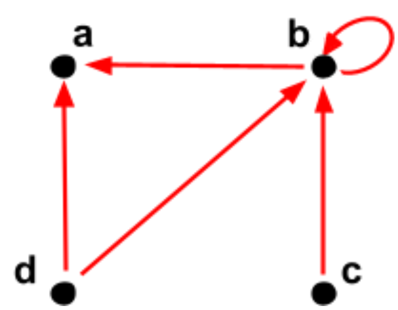
How can a computer rank these sources? In the case that there are thousands of hits, ranking is even more important because only a few will be seen by the users. Define a directed graph whose vertex set is the set of all hits. (X, Y) is an edge in the graph if page X has a hyperlink to page Y. Suppose the graph looks like:



The real algorithms used by search companies are beyond the scope of this material, but hyperlinks are a critical component of the process. Search engines like Google assign a numerical score to each web page and rank the page according to their score. The score depends on the in-degree of a page as well as the importance of the pages who have directed edges into that page. One might guess from the structure above that a helpful ranking of the pages would look like:

1. C: An article on the web pages of the American Academy of Pediatrics
2. D: An article on the web pages of a pharmaceutical company selling allergy medication for children.
3. B: An article written by a local pediatrician on children's allergies
4. A: A blog entry by a parent whose child has an allergy.

PARTICIPATION ACTIVITY | 6.3.2: In-degree and out-degree.



1) Select the ordered pair which is an edge in the graph.

- (b, c)
- (c, b)
- (c, c)

2) In the graph above, which vertex has the largest in-degree?

- a
- b

- c
- d

3) In the graph above, which vertex has the smallest out-degree?

- a
- b
- c
- d

Walks in directed graphs

Consider a directed graph corresponding to airline flights. The vertex set is the set of airports used by the airline and a directed edge from airport u to airport v represents a direct flight from u to v. A person wishing to travel from Tallahassee to Peoria would naturally prefer to take a direct flight. However, if the airline does not have a direct flight, he or she may be able to make the trip from Tallahassee to Peoria by a series of direct flights. The flight graph can be used to determine when such a trip is possible.

In the language of graphs, we are asking whether it is possible to travel from one vertex to another by a series of hops along the directed edges of the graph.

Definition 6.3.1: A walk in a directed graph.

A ***walk*** from v_0 to v_l in a directed graph G is a sequence of alternating vertices and edges that starts and ends with a vertex.

$$\langle v_0, (v_0, v_1), v_1, (v_1, v_2), v_2, \ldots, v_{l-1}, (v_{l-1}, v_l), v_l \rangle$$

Each edge in the sequence appears after its tail and before its head:
$\ldots, v_{i-1}, (v_{i-1}, v_i), v_i, \ldots$

Since the edges in a walk are completely determined by the vertices, a walk can also be denoted by the sequence of vertices:

$$\langle v_0, v_1, \ldots, v_l \rangle.$$

The sequence of vertices is a walk only if $(v_{i-1}, v_i) \in E$ for each i = 1, 2,...,l. Two consecutive vertices $\langle \ldots, v_{i-1}, v_i, \ldots \rangle$ in a walk represent an occurrence of the edge (v_{i-1}, v_i) in the walk.

The ***length of a walk*** is l, the number of edges in the walk.

An ***open walk*** is a walk in which the first and last vertices are not the same. A ***closed walk*** is a walk in which the first and last vertices are the same.

PARTICIPATION ACTIVITY | 6.3.3: Walks in directed graphs.

Animation content:

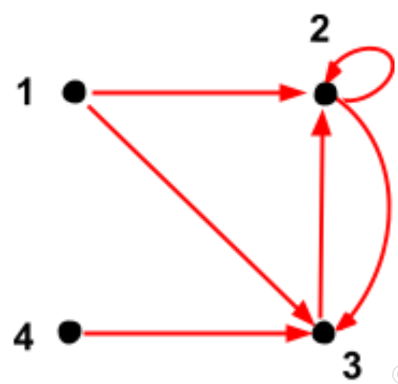
undefined

Animation captions:

1. ⟨a, b, c, b, d⟩ is a walk in the graph because (a, b), (b, c), (c, b), and (b, d) are all edges in the graph.
2. ⟨a, b, c, b, d⟩ is an open walk because the first and last vertices are not the same.
3. The walk ⟨a, b, c, b, d⟩ has length 4 because there are four edges in the walk.
4. ⟨b, d, c, b, c, b⟩ is a walk in the graph because (b, d), (d, c), (c, b) and (b, c) are all edges in the graph. The walk is closed because the first and last vertices are the same.
5. The length of walk ⟨b, d, c, b, c, b⟩ is 5 because there are 5 edges in the walk, including repetitions.
6. ⟨d, d⟩ is a closed walk of length 1.
7. ⟨a⟩ is a closed walk of length 0.

PARTICIPATION ACTIVITY | 6.3.4: Identifying open and closed walks.

The next four questions refer to the directed graph shown:



1) The sequence ⟨ 1, 2, 4 ⟩ is an open walk.

- True
- False

2) The sequence ⟨ 2, 3, 2, 2 ⟩ is a closed walk.

- True
- False

3) The sequence ⟨ 1, 2, 3, 2 ⟩ is an open walk.

- True
- False

4) The sequence ⟨ 2, 2, 2 ⟩ is a closed walk.

- True
- False

Trails, circuits, paths, and cycles

In many contexts, walks that do not repeat vertices or edges are preferable. For example, an airline trip that arrived at the same airport twice would be inefficient.

- A ***trail*** is a walk in which no edge occurs more than once.
- A ***path*** is a walk in which no vertex occurs more than once.
- A ***circuit*** is a closed trail.
- A ***cycle*** is a circuit of length at least 1 in which no vertex occurs more than once, except the first and last vertices which are the same.

⟨a, c, d, a⟩ is a cycle because the only repeated vertices are the first and the last, a. The circuit ⟨a, c, a, d, a⟩ is not a cycle because the vertex a appears in the middle of the circuit as well as at the beginning and the end. The circuit ⟨b, c, d, c, b⟩ is also not a cycle because the vertex c is repeated in the middle of the circuit.

PARTICIPATION ACTIVITY 6.3.5: Trails, circuits, paths, and cycles.

Animation captions:

1. ⟨a, b, c, b, d⟩ is a walk in the graph because (a, b), (b, c), (c, b), and (b, d) are all edges in the graph.
2. No edge occurs more than once. So the open walk ⟨a, b, c, b, d⟩ is a trail.
3. Vertex b is reached twice, so this trail is not a path.
4. ⟨b, d, c, b⟩ is a walk in the graph because (b, d), (d, c), and (c, b) are all edges.

5. The closed walk ⟨b, d, c, b⟩ is a circuit because no edge occurs more than once. The circuit is a cycle because only the first and last vertices are repeated.

PARTICIPATION ACTIVITY | 6.3.6: Trails, circuits, paths, and cycles.

The sequences below are all walks in a graph. Select the correct description for each sequence.

1) ⟨ 1, 2, 3, 2, 1 ⟩

- Neither a circuit nor a cycle.
- A circuit but not a cycle.
- A circuit and a cycle.

2) ⟨ 2, 1, 3, 4 ⟩

- Neither a trail nor a path.
- A trail but not a path.
- A trail and a path.

3) ⟨ 2, 3, 2 ⟩

- Neither a circuit nor a cycle.
- A circuit but not a cycle.
- A circuit and a cycle.

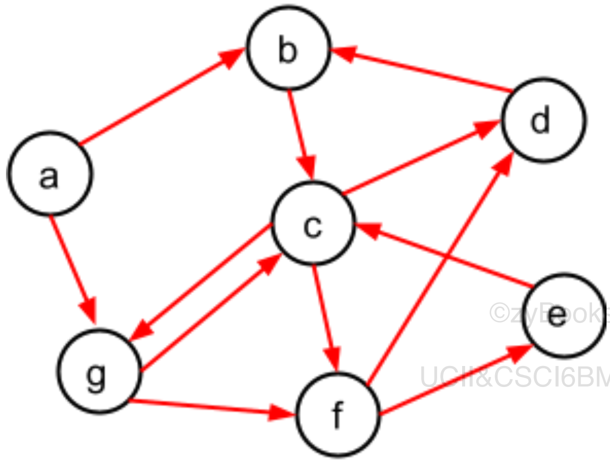
4) ⟨ 1, 4, 3, 1, 2, 5, 1 ⟩

- Neither a circuit nor a cycle.
- A circuit but not a cycle.
- A circuit and a cycle.

Additional exercises

EXERCISE | 6.3.1: Directed graph definitions.

The diagram below shows a directed graph.



(a) What is the in-degree of vertex d?

Solution ⌄

(b) What is the out-degree of vertex c?

Solution ⌄

(c) What is the head of edge (b, c)?

Solution ⌄

(d) What is the tail of edge (g, f)?

Solution ⌄

(e) List all the self-loops in the graph.

Solution ⌄

(f) Is ⟨a, g, f, c, d⟩ a walk in the graph? If so, is it an open or closed walk? Is it a trail, path, circuit, or cycle?

Solution ⌄

(g) Is ⟨a, g, f, d, b⟩ a walk in the graph? If so, is it an open or closed walk? Is it a trail, path, circuit, or cycle?

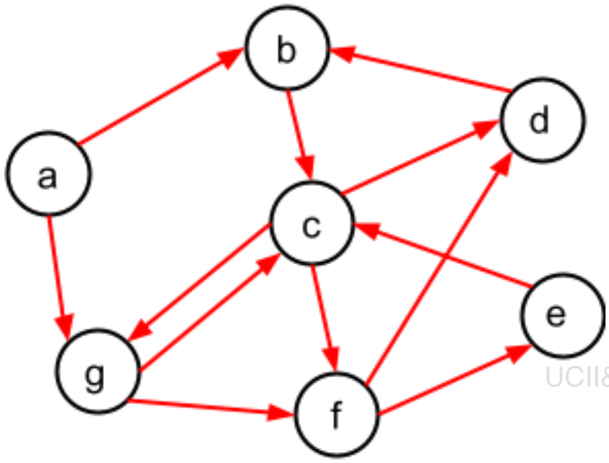
Solution ⌄

(h) Is ⟨c, g, f, e⟩ a circuit in the graph? Is it a cycle?

Solution ⌄

EXERCISE | 6.3.2: Directed graph definitions, cont.

The diagram below shows a directed graph.



(a) Is $\langle$d, b, c, g, c, f, e, c, d$\rangle$ a circuit in the graph? Is it a cycle?

Solution

(b) What is the longest cycle in the graph?

Solution

(c) Give an example of a cycle of length 4.

(d) Give an example of a path of length 5.

Solution

(e) Is there a path of length 3 from vertex d to vertex f? If so, give an example.

Solution

(f) Is there a path of length 3 from vertex a to vertex c? If so, give an example.

Solution

(g) Give an example of an open trail of length 4 that is not a path.

(h) Give an example of a circuit of length 5 that is not a cycle.

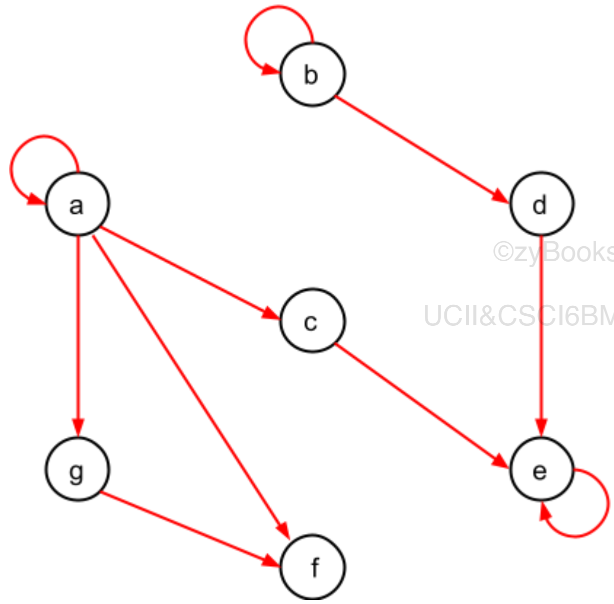
(i) Give an example of a circuit of length 6 that is not a cycle.

Solution

(j) Is it true that for each pair of vertices there is a path from one to the other?

EXERCISE 6.3.3: Directed graph definitions, cont.

The diagram below shows a directed graph.



(a) Which vertex has the largest in-degree? What is the in-degree for that vertex?

(b) Which vertex has the largest out-degree? What is the out-degree for that vertex?

Solution

(c) List all the self-loops in the diagram.

(d) Is the sequence ⟨b, d, e, e⟩ a walk in the graph? If it is, is it an open walk?

(e) Is the sequence ⟨a, c, f, g⟩ a walk in the graph? If so, is it an open or closed walk? Is it a trail, path, circuit, or cycle?

Solution

(f) Is the sequence ⟨a, a, c, e, e⟩ a walk in the graph? If so, is it an open or closed walk? Is it a trail, path, circuit, or cycle?

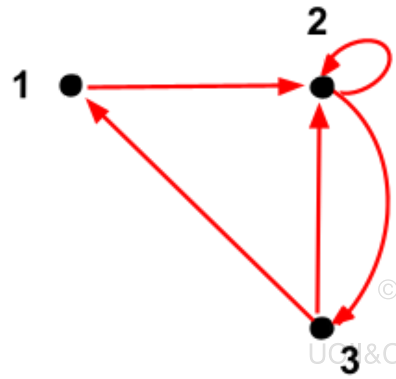
Solution

(g) Give another example of an open trail of length 4 that is not a path.

Solution

EXERCISE 6.3.4: Finding walks of a specific length.

The questions below refer to the directed graph shown:



(a) List all the numbers x, such that there is a trail in the graph of length x. Justify your answer.

Solution

(b) List all the numbers x, such that there is a path in the graph of length x. Justify your answer.

Solution

(c) List all the numbers x, such that there is a circuit in the graph of length x. Justify your answer.

Solution

(d) List all the numbers x, such that there is a cycle in the graph of length x. Justify your answer.

Solution

6.4 Composition of relations

There is a one-to-one correspondence between directed graphs and binary relations in that the arrow diagram for a binary relation is a directed graph. Similarly the edge set of a directed graph defines a binary relation on the set of vertices of that graph. If a directed graph G has a walk of length k from vertex a to vertex b, then what does that say about the binary relation corresponding to G? Before defining the counterpart of a walk in a binary relation, we need to first define the composition of two relations.

The ***composition*** of relations R and S on set A is another relation on A, denoted S o R. The pair (a, c) ∈ S o R if and only if there is a b ∈ A such that (a, b) ∈ R and (b, c) ∈ S.

The ordering of the relations R and S in the expression "S o R" may seem unusual because it is natural to read an expression from left to right. However, the ordering defined here for composition of relations is consistent with the ordering defined for composition of functions in which R is applied first and then S. The following animation illustrates the composition of two relations:

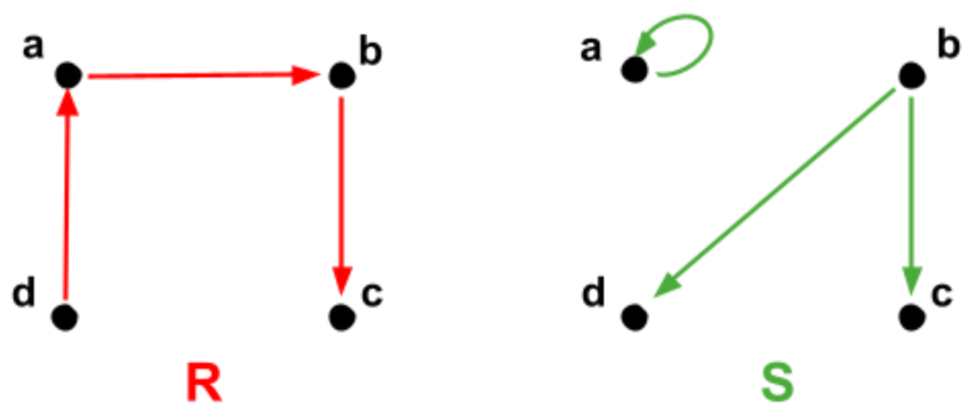
PARTICIPATION ACTIVITY | 6.4.1: Composition of relations.

Animation captions:

1. R and S are relations on set A. (a, c) ∈ S o R because there is a vertex d such that (a,d) ∈ R and (d,c) ∈ S.
2. Since (c,d) ∈ R and (d,c) ∈ S, (c,c) ∈ S o R.
3. Since (b,b) ∈ R and (b,d) ∈ S, (b,d) ∈ S o R.
4. Since (b,b) ∈ R and (b,a) ∈ S, (b,a) ∈ S o R.

PARTICIPATION ACTIVITY | 6.4.2: Composition of relations.

Consider the following two relations on the set {a, b, c, d}:



1) Is (a, d) in S o R?

- Yes
- No

2) Is (b, d) in S o R?

- Yes
- No

3) Is (b, c) in S o R?

- Yes
- No

4) Is (d, a) in S o R?

- Yes

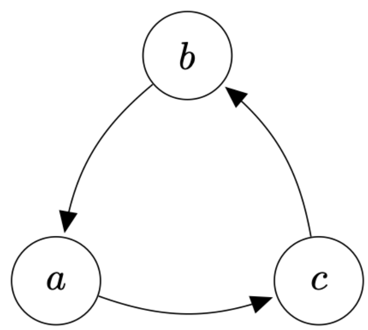
No

CHALLENGE ACTIVITY | 6.4.1: Composition of relations.

460784.3106054.qx3zqy7

Start

Given the relation U below, use ordered pair notation to express the relation U o U.



U

U o U = { Ex: (a, b), (b, c) }

1 2 3 4

Check Next

Additional exercises

EXERCISE | 6.4.1: Composition of relations expressed as a set of pairs.

Here are two relations defined on the set {a, b, c, d}:

S = { (a, b), (a, c), (c, d), (c, a) }
R = { (b, c), (c, b), (a, d), (d, b) }

Write each relation as a set of ordered pairs.

(a) S o R

(b) R o S

(c) S o S

Solution

(d) R o R

Solution

EXERCISE 6.4.2: Composition of relations on the real numbers.

Here are four relations defined on **R**, the set of real numbers:

$R_1 = \{ (x, y): x \leq y \}$
$R_2 = \{ (x, y): x > y \}$
$R_3 = \{ (x, y): x < y \}$
$R_4 = \{ (x, y): x = y \}$

Describe each relation below. (Hint: each of the answers will be one of the relations R_1 through R_4 or the relation **R**x**R**.)

(a) R_1 o R_2

(b) R_4 o R_1

(c) R_1 o R_1

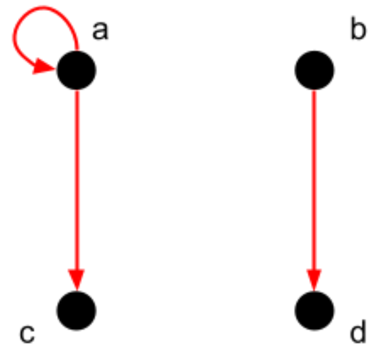
Solution

(d) R_3 o R_1

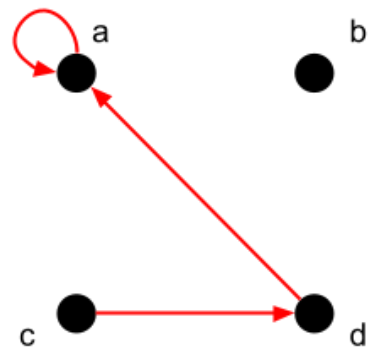
Solution

EXERCISE 6.4.3: Composition of relations and arrow diagrams.

The arrow diagrams for relations A and B are shown below. Both relations have the domain {a, b, c, d}.



A



B

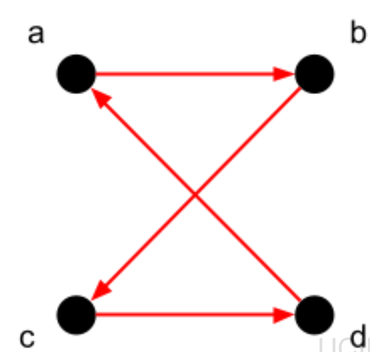
(a) Draw the arrow diagram for $B \circ A$.

Solution

(b) Draw the arrow diagram for $A \circ B$.

EXERCISE 6.4.4: Composition of relations and arrow diagrams, cont.

Below is the arrow diagram for relation S with the domain {a, b, c, d}. Define relation T to be $S \circ S$.



S

(a) Express relation T as a set of related pairs.

Solution

(b) Draw the arrow diagram for $S \circ T$.

Solution

EXERCISE 6.4.5: Composition and relation properties.

For the following statements, provide a proof if the statement is true or give a counterexample if the statement is false. S and R are binary relations over the same domain.

(a) If S and R are both reflexive, then $S \circ R$ is reflexive.

(b) If S is reflexive, then $S \circ S$ is reflexive.

Solution

(c) If S and R are both anti-reflexive, then $S \circ R$ is anti-reflexive.

(d) If S is anti-reflexive, then $S \circ S$ is anti-reflexive.

Solution

(e) If S and R are both anti-symmetric, then $S \circ R$ is anti-symmetric.

Solution

(f) If S is anti-symmetric, then $S \circ S$ is anti-symmetric.

(g) If S and R are both symmetric, then $S \circ R$ is symmetric.

(h) If S is symmetric, then $S \circ S$ is symmetric.

Solution

(i) If S and R are both transitive, then $S \circ R$ is transitive.

Solution

(j) If S is transitive, then $S \circ S$ is transitive.

6.5 Graph powers and the transitive closure

A relation on a set can be composed with itself. For example, consider a relation P on a set of people that expresses parent-child relationships. xPy means that x is the parent of y. x(P o P)z holds if there is a person y such that x is the parent of y and y is the parent of z. In other words, x(P o P)z means that x is a grandparent of z.

In the directed graph corresponding to the relation P o P, there is an edge from x to z if there is a vertex y such that there is an edge from x to y in P and an edge from y to z in P. Thus, the directed graph corresponding to P o P represents all walks of length 2 in P. More generally:

$R^1 = R$
$R^k = R \text{ o } R^{k-1}$, for all $k \geq 2$

The edge set E of a directed graph G can be viewed as a relation. E^k is the relation E composed with itself k times. The graph G^k is defined to be the directed graph whose edge set is E^k and is called the k^{th} ***power of G***. G^k expresses walk relationships between vertices in the following natural way:

Theorem 6.5.1: The Graph Power Theorem.

Let G be a directed graph. Let u and v be any two vertices in G. There is an edge from u to v in G^k if and only if there is a walk of length k from u to v in G.

The proof is given at the end of this section. The following animation gives some intuition about why the theorem holds using a small example. The notation ⟨ a, b, *, c ⟩ used in the animation indicates a path of length 3 in which the first vertex is a, the second vertex is b, the fourth vertex is c, and the third vertex is unknown.

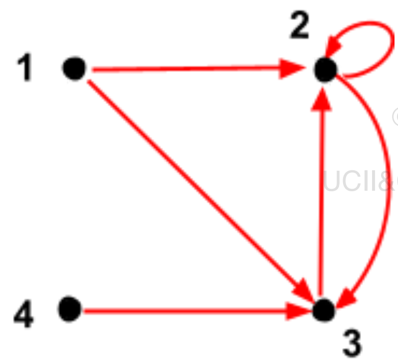
PARTICIPATION ACTIVITY 6.5.1: Graph powers and walks.

Animation captions:

1. The walk ⟨a, d, a⟩ in G implies that (a,a) is an edge in G^2.
2. The walk ⟨b, a, d⟩ in G implies that (b,d) is an edge in G^2.
3. The walk ⟨c, b, a⟩ in G implies that (c,a) is an edge in G^2.
4. ⟨c, d, a⟩ is a walk in G, but (c,a) is already present in G^2.
5. The walk ⟨d, a, d⟩ in G implies that (d,d) is an edge in G^2.
6. G^3 represents walks of length 3 in G and is obtained by composing G with G^2.
7. The walk ⟨a,*,a⟩ in G (represented by edge (a,a) in G^2) and the edge (a,d) in G imply an edge (a,d) in G^3.
8. The walk ⟨b,*,d⟩ in G (represented by edge (b, d) in G^2) and the edge (d, a) in G imply an edge (b, a) in G^3.
9. The walk ⟨c,*,a⟩ in G (represented by edge (c, a) in G^2) and the edge (a, d) in G imply an edge (c, d) in G^3.
10. The walk ⟨d,*,d⟩ in G (represented by edge (d, d) in G^2) and the edge (d, a) in G imply an edge (d, a) in G^3.

PARTICIPATION ACTIVITY | 6.5.2: Graph powers.

The figure below shows directed graph G:



1) Which one is not an edge in G^2?
- (1, 3)
- (1, 2)
- (4, 3)

2) How many edges are in G^1?
- 0
- 6
- 4

3) Is (4, 2) an edge in G^3?
- Yes
- No

The transitive closure

The union of G^k for all $k \geq 1$ (denoted G^+) represents reachability by walks of any positive length in G. In taking the union of graphs, there is only one copy of the vertex set and the union is taken over the edge sets of the respective graphs.

$$G^+ = G^1 \cup G^2 \cup G^3 \cup G^4 \ldots.$$

(u, v) is an edge in G^+ if vertex v can be reached from vertex u in G by a walk of any length. While the expression given above for $G^+$is an infinite union, if the vertex set is finite, then only graph powers up to |V| are required. Let G be a graph on a finite vertex set with n vertices. Then

$$G^+ = G^1 \cup G^2 \cup G^3 \cup \ldots \cup G^n$$

The same definition holds for a relation R. Let R be a relation on a finite domain with n elements. Then

$$R^+ = R^1 \cup R^2 \cup R^3 \cup \ldots \cup R^n$$

The relation R^+ is called the ***transitive closure of R*** and is the smallest relation that is both transitive and includes all the pairs from R. In other words, any relation that contains all the pairs from R and is transitive must include all the pairs in R^+. If G is a directed graph, then G^+ is called the ***transitive closure of G***. The animation below shows how the graph G^+(with 4 vertices) is determined from the graphs G^1 through G^4.

PARTICIPATION ACTIVITY 6.5.3: The transitive closure of G as a union of graph powers.

Animation captions:

1. G has 4 vertices. Therefore G^+ is the union of graphs G, G^2, G^3, and G^4. Every edge in G, G^2, G^3, and G^4 becomes an edge in G^+.

There is an alternative way to find the transitive closure of a graph or relation that does not require computing the powers directly. The process repeatedly looks for three elements, x, y, z, such that (x, y) and (y, z) are pairs in the relation but (x, z) is not in the relation. If there is such a triplet of elements, then the pair (x, z) is added to the relation. The process eventually ends when there is no such triplet of elements. The procedure outlined below will eventually terminate with R^+:

Figure 6.5.1: Procedure to find the transitive closure of a relation R on a set A.

Repeat the following step until no pair is added to R:

- If there are three elements $x, y, z \in A$ such that $(x, y) \in R$, $(y, z) \in R$ and $(x, z) \notin R$, then add (x, z) to R.

The following animation illustrates the process described above:

PARTICIPATION ACTIVITY 6.5.4: Finding the transitive closure of a relation.

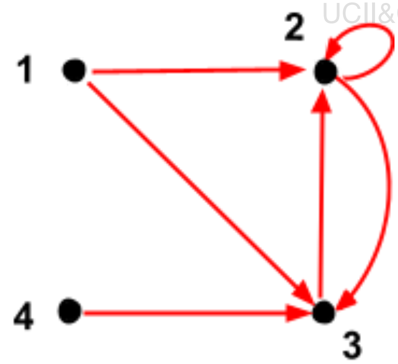
Animation captions:

1. To find the transitive closure of relation R, start with the pairs in R and add pairs. Since (a,c) and (c,d) are present, (a,d) is added.

2. Since (b,c) and (c,d) are present, (b,d) is added. After (a,e), (b,e), and (c,e) are added, there are no more walks ⟨x, *, y⟩ without the presence of edge (x,y).

PARTICIPATION ACTIVITY 6.5.5: The transitive closure of a graph.

The figure below shows directed graph G:



1) Is (1, 3) an edge in G^+?
- Yes
- No

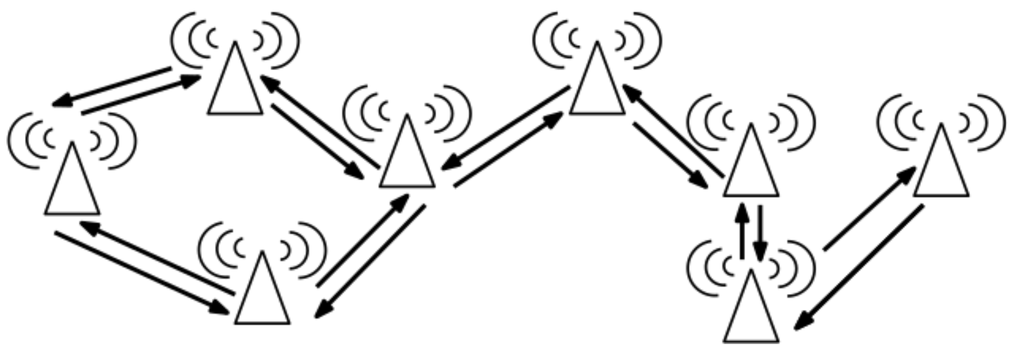
2) Is (3, 3) an edge in G^+?
- Yes
- No

3) What is the in-degree of vertex 4 in G^+?
- 3
- 2
- 0

4) What is the in-degree of vertex 3 in G^+?
- 1
- 3
- 4
- 2

Example 6.5.1: Connectivity in sensor networks.

A sensor network consists of a set of sensors distributed over a geographical area. The sensors themselves are typically small, low-cost devices with a limited amount of power that fades over time with use. Sensor networks are used in many industrial and military applications to monitor processes, machinery, or people. Typically, a sensor is only able to transmit a message to other nearby sensors. The radius of a sensor's range may depend in part on its remaining power. The communication graph for the network has a vertex set corresponding to the sensors in the network with a directed edge from sensor x to sensor y if x can send a message directly to y. Messages can also be transmitted through the network along a path: x sends a message to y, y transmits the message to z, and so on. One of the goals in the design of sensor networks is to maintain connectivity as long as possible. In order to test connectivity, it is important to be able to answer questions like: Is there a path from x to y in the network along the directed edges of the communication graph? This question can be answered for all pairs of sensors by computing G^+ where G is the communication graph.



Induction overview for the proof of the Graph Power Theorem

The proof of the Graph Power Theorem requires a technique called induction which is covered in more depth elsewhere. Here is a brief introduction to induction followed by a proof of the theorem. The theorem states that for any k ≥ 1:

there is an edge (u, v) in G^k $\leftrightarrow$ there is a walk from u to v of length k in G.

The proof must show that the theorem holds for an infinite sequence values for k: k = 1, 2, ... ***Induction*** starts by showing that a theorem is true for k = 1.Then an inductive proof shows that for any k > 1, if the theorem is true for k - 1, then the theorem also holds for k. The fact that the Graph Power Theorem is true for k = 1 follows almost immediately from the definitions.

Proof 6.5.1: Proof of the Graph Power Theorem.

Theorem: Let G be a directed graph. Let u and v be any two vertices in G. There is an edge from u to v in G^k if and only if there is a walk of length k from u to v in G.

Proof.

By induction on k.

Base case: k = 1. G^1 = G, by definition. Moreover an edge (u, v) is a walk ⟨u, v⟩ of length 1. Therefore, there is an edge (u, v) in G^1 if and only if there is a walk ⟨u, v⟩ of length 1 in G.

Inductive step: Prove that for k > 1, if it is true that:

there is an edge (u, v) in G^{k-1} $\leftrightarrow$ there is a walk from u to v of length k - 1 in G

then it is true that:

there is an edge (u, v) in G^k $\leftrightarrow$ there is a walk from u to v of length k in G

E, E^{k-1}, and E^k are the edge sets for G, G^{k-1}, and G^k. By definition, E^k is obtained by composing E and E^{k-1}: $E^k = E \circ E^{k-1}$. By definition of composition, there is an edge (u, v) in G^k if and only if there is a vertex x, such that (u, x) is an edge in G^{k-1} and (x, v) is an edge in G. By the assumption that the theorem holds for k - 1, there is an edge (u, x) in G^{k-1} if and only if there is a walk from u to x of length k - 1 in G. The edge (x, v) in G is a walk of length 1 in G: ⟨x, v⟩.

We have shown that there is an edge (u, v) in G^k if and only if there is a vertex x such that there is a walk of length k-1 from u to x and a walk of length 1 from x to v in G.

It remains to show that there is an x such that there is a walk of length k-1 from u to x and a walk of length 1 from x to v in G if and only if there is a walk of length k in G from u to v. Each direction of the "if and only if" is proven separately.

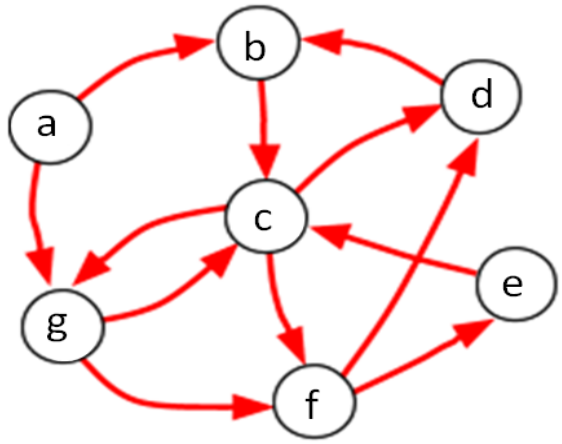
Suppose that ⟨u, ..., x⟩ is a walk of length k-1 and ⟨x, v⟩ is a walk of length 1 in G. The walks ⟨u, ... , x⟩ and ⟨x, v⟩ can be put together to form a walk ⟨u, ..., x, v⟩. The length of the walk from u to v is the sum of the lengths of walks from u to x and from x to v: (k - 1) + 1 = k.

Now suppose that there is a walk of length k from u to v in G. Let x be the second-to-last vertex reached on the walk: ⟨u, ... x, v⟩. Then ⟨u, ..., x⟩ must be a walk of length k - 1 from u to x in G and (x, v) must be an edge in G. ■

Additional exercises

EXERCISE | 6.5.1: Edges of graph powers.

The diagram below shows a directed graph G.



(a) Is (a, b) in G^2?

Solution

(b) Is (b, e) in G^3?

Solution

(c) Is (g, g) in G^3?

(d) Is (g, g) in G^4?

(e) Is (b, b) in G^3?

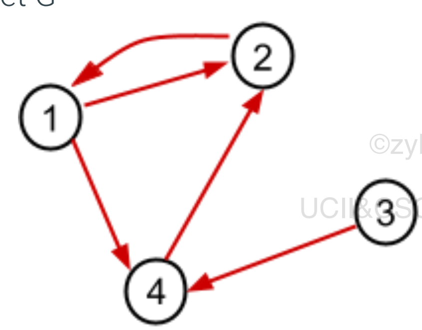
Solution

(f) Is (b, d) in G^5?

Solution

EXERCISE | 6.5.2: Drawing graph powers.

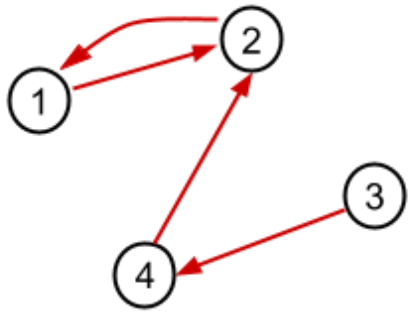
(a) The drawing below shows a graph G. Draw G^2, G^3, and G^4. Then take the union of all of the graphs (including G) to get G^+



EXERCISE | 6.5.3: Finding the transitive closure of a graph.

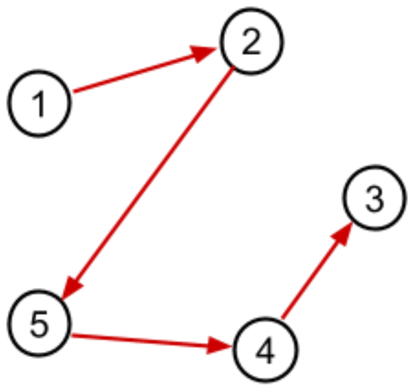
Draw the transitive closure of each graph.

(a)

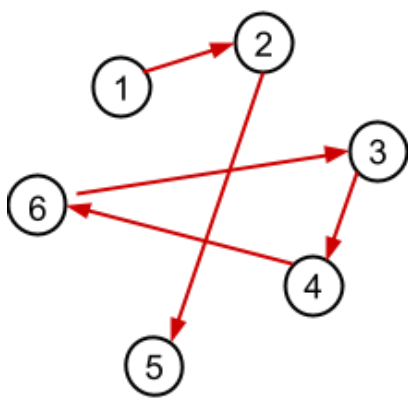


Solution

(b)



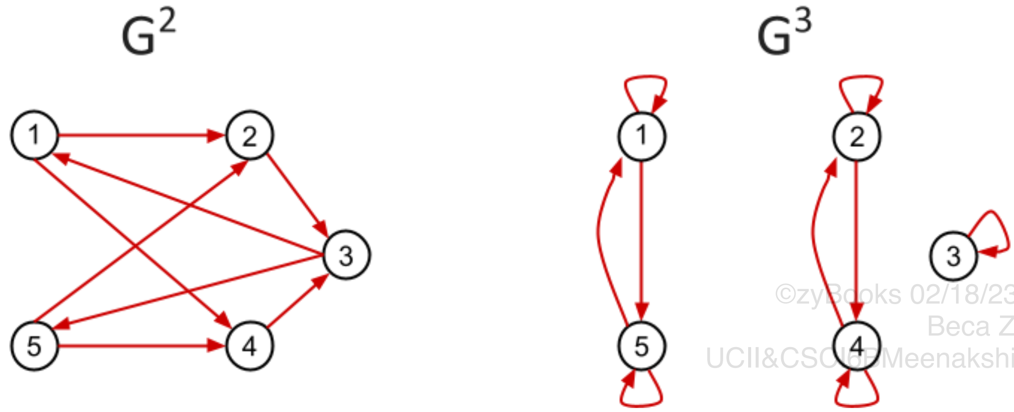
(c)



Solution

EXERCISE | 6.5.4: Inferring facts about a graph from its graph powers.

The drawing below shows G^2 and G^3 for a graph G.



Use the information provided in G^2 and G^3 to answer the following questions about G.

(a) Is there a walk of length 3 from vertex 4 to vertex 5 in G?

(b) Is there any closed walk of length 2 in G?

(c) Is there any closed walk of length 3 in G?

Solution

(d) Is there a walk of length 4 from vertex 2 to vertex 1 in G?

(e) Is there a walk of length 5 from vertex 2 to vertex 3 in G?

Solution

(f) Is there a walk from vertex 2 to vertex 4 in G?

Solution

(g) Which vertices can be reached from vertex 3 in G by a walk of length 2?

Solution

(h) Which vertices have a walk of length 3 to vertex 1 in G?

Solution

(i) Is it possible to infer the out-degree of vertex 4 in G^+ from the information given?

EXERCISE 6.5.5: Properties of relations and the transitive closure.

(a) Consider a digraph G in which each vertex has an in-degree of at least one. Suppose that the relation defined by the edges of G is symmetric. Is G^+ reflexive? Why or why not?

EXERCISE 6.5.6: Composition and relation properties, cont.

(a) Show that if G is reflexive then every edge in G^k is also an edge in G^{k+1}.

(b) Show that if G is reflexive then $G^+ = G^n$.

6.6 Matrix multiplication and graph powers

An n × m ***matrix*** over a set S is an array of elements from S with n rows and m columns. Each element in a matrix is called an ***entry***. The entry in row i and column j of matrix A is denoted by $A_{i,j}$. A matrix is called a ***square matrix*** if the number of rows is equal to the number of columns.

Figure 6.6.1: Examples of matrices.

$$\begin{pmatrix} 1 & 3 \\ 3 & -5 \\ -2 & -2 \end{pmatrix} \qquad \begin{pmatrix} 1.1 & 3.0 & -5.4 \\ -2.2 & -2.1 & 1.0 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

3 x 2 matrix over **Z** 2 x 3 matrix over **R** 2 x 2 matrix over {0, 1}

PARTICIPATION ACTIVITY 6.6.1: Matrix basics.

$$A = \begin{pmatrix} 4 & 7 \\ 2 & 3 \\ 7 & 9 \end{pmatrix} \qquad B = \begin{pmatrix} 2 & 5 & 5 \\ 1 & -2 & 9 \end{pmatrix} \qquad C = \begin{pmatrix} 4 & 7 \\ 2 & 3 \end{pmatrix}$$

1) Which matrix is a 2 × 3 matrix?
Type A, B, or C.

Check Show answer

2) Which matrix is a square matrix?

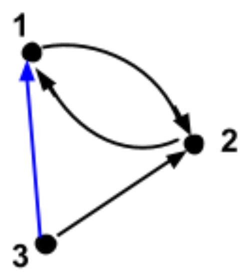
Check Show answer

3) What is the value of $A_{3,2}$?

Check Show answer

A directed graph G with n vertices can be represented by an n × n matrix over the set {0, 1} called the ***adjacency matrix*** for G. If matrix A is the adjacency matrix for a graph G, then $A_{i,j}$ = 1 if there is an edge from vertex i to vertex j in G. Otherwise, $A_{i,j}$ = 0.

Figure 6.6.2: A directed graph and the corresponding adjacency matrix.

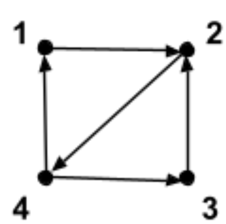


$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

The directed graph's edge, highlighted in blue, goes from vertex 3 to vertex 1. The corresponding entry in the adjacency matrix shows a 1 in row 3, column 1.

PARTICIPATION ACTIVITY 6.6.2: Adjacency matrix for a directed graph.

The figure below shows a directed graph and its corresponding matrix A with two missing entries.



$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ ? & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & ? & 0 \end{pmatrix}$$

1) What is the correct value for $A_{2,1}$?

- 0

○ 1

2) What is the correct value for $A_{4,3}$?

○ 0

○ 1

If mathematical operations, such as addition and multiplication, are defined on a set S, then matrix addition and multiplication can be defined for matrices over the set S. A ***Boolean matrix*** is a matrix whose entries are from the set {0, 1}. Boolean addition and multiplication are used in adding and multiplying entries of a Boolean matrix. This material presents matrix addition and multiplication for square Boolean matrices because those operations can be used to compute the transitive closure of a graph.

Matrix multiplication and addition can also be defined for general rectangular matrices over other sets such as the integers or the real numbers. Matrix multiplication and addition over the real numbers are useful operations in many contexts such as scientific computing and computer graphics.

The product of two matrices, A and B, is well defined only if the number of columns in A is equal to the number of rows in B. Each entry of the matrix A × B is computed by taking the dot product of a row of A and a column of B. If A and B are n × n matrices, the ***dot product*** of row i of A and column j of B is the sum of the product of each entry in row i from A with the corresponding entry in column j from B:

$$A_{i,1}B_{1,j} + A_{i,2}B_{2,j} + \cdots + A_{i,n}B_{n,j}$$

PARTICIPATION ACTIVITY 6.6.3: Illustration of the Boolean matrix dot product.

Animation captions:

1. The dot product is computed by multiplying each entry of a row with the matching entry in a column and then taking the sum.

PARTICIPATION ACTIVITY 6.6.4: Computing dot products of rows and columns of Boolean matrices.

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \qquad B = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

1) What is the dot product of row 3 of matrix A and column 2 of matrix B?

○ 1

○ 2

2) What is the dot product of row 2 of matrix A and column 1 of matrix B?

○ 0

○ 1

If A and B are n × n matrices over the integers, then the ***matrix product*** of A and B, denoted AB or A⋅B, is another n × n matrix such that $(AB)_{i,j}$ is the result of taking the dot product of row i of matrix A and column j of matrix B.

PARTICIPATION ACTIVITY | 6.6.5: Illustration of Boolean matrix multiplication.

Animation captions:

1. A × B = C. The entry in row 1, column 1 of C is the dot product of row 1 of A and column 1 of B.
2. The entry in row 1, column 2 of C is the dot product of row 1 of A and column 2 of B.
3. In general, the entry in row i, column j of C is the dot product of row i of A and column j of B.

PARTICIPATION ACTIVITY | 6.6.6: Computing the product of two Boolean matrices.

The product of matrices A and B is given below with a few entries missing.

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \quad A \cdot B = \begin{pmatrix} 1 & ? & 1 \\ ? & 0 & 1 \\ 0 & 1 & ? \end{pmatrix}$$

1) What is $(AB)_{1,2}$?

○ 0

○ 1

2) What is $(AB)_{2,1}$?

○ 0

○ 1

3) What is $(AB)_{3,3}$?

○ 0

○ 1

Matrix multiplication is associative, meaning that if A, B, and C are all n × n matrices, then A(BC) = (AB)C. However, matrix multiplication is not commutative because there are matrices A and B for which AB ≠ BA. The k^{th} ***power of a matrix*** A is the product of k copies of A:

$$A^k = \underbrace{A \cdot A \cdots A}_{\text{k times}}$$

The powers of a matrix can be computed by repeated multiplication.

$$\begin{aligned} A^2 &= A \cdot A \\ A^3 &= A \cdot A \cdot A = (A \cdot A) \cdot A = A^2 \cdot A \\ A^4 &= A \cdot A \cdot A \cdot A = (A \cdot A \cdot A) \cdot A = A^3 \cdot A \end{aligned}$$

PARTICIPATION ACTIVITY 6.6.7: Computing the powers of a matrix.

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

1) What is A^2?

- $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
- $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$
- $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

2) What is A^3?

- $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
- $\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$

CHALLENGE ACTIVITY 6.6.1: Boolean matrix multiplication.

Start

Select the row of and the column of whose dot product is the entry in the product .

1 2 3 4 5

Check Next

If G is a directed graph, then the k^{th} power of G (G^k) represents walks of length k in G. There is an edge from vertex v to vertex w in G^k if and only if there is a walk of length exactly k from v to w in G. Matrix multiplication provides a systematic way of computing G^k.

1. Take the adjacency matrix A for graph G.
2. Compute A^k by repeated matrix multiplication.
3. The matrix A^k is the adjacency matrix for graph G^k. There is a walk of length k in G from vertex v to vertex w if and only if the entry in row v, column w in A^k is 1.

The fact that the method described above works is encapsulated in the theorem statement below. The proof of the theorem appears at the end of the section.

Theorem 6.6.1: Relationship between the powers of a graph and the powers of its adjacency matrix.

Let G be a directed graph with n vertices and let A be the adjacency matrix for G. Then for any $k \geq 1$, A^k is the adjacency matrix of G^k, where Boolean addition and multiplication are used to compute A^k.

PARTICIPATION ACTIVITY 6.6.8: Matrix multiplication and walks in directed graphs.

G is a graph with 5 vertices, numbered 1 through 5. A is the adjacency matrix for G. The matrices A and A^3 are given below but only some of the entries are visible:

$$A^3 = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{pmatrix} \qquad A = \begin{pmatrix} * & * & * & * & 1 \\ * & * & * & * & 1 \\ * & * & * & * & 0 \\ * & * & * & * & 0 \\ * & * & * & * & 1 \end{pmatrix}$$

1) What is the entry in row 1, column 5 of the adjacency matrix for G^4?

- 0
- 1

2) Is there a walk of length 4 in G from vertex 1 to vertex 5?

- Yes
- No

3) A walk of length 3 that starts at vertex 1 could end at which vertices?

- 1 and 3
- 2 and 3
- 1 and 2

4) Which vertices have an outgoing edge that points to vertex 5?

- 1, 2, and 3
- 1 and 2
- 1, 2, and 5

5) Which walk accurately describes a walk in G of length 4 from vertex 1 to vertex 5?

- ⟨1, *, *, 3, 5⟩

- ⟨1, *, *, 5, 5⟩
- ⟨1, *, *, 2, 5⟩

PARTICIPATION ACTIVITY | 6.6.9: Graph powers and matrix powers.

The matrix A given below is the adjacency matrix for a directed graph G with 4 vertices. The matrices $A^2 = A \cdot A$ and $A^3 = A \cdot A \cdot A$ have been computed using matrix multiplication. The rows and columns of the matrices are numbered 1 through 4 corresponding to the vertices in G, G^2, and G^3.

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \qquad A^2 = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \qquad A^3 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

1) How many edges are in the graph G^2?

Check Show answer

2) Is there a walk of length 3 in G from vertex 1 to vertex 3? Type yes or no.

Check Show answer

3) Starting from vertex 2, how many vertices can be reached by a walk of length 2 in G?

Check Show answer

The sum of two matrices A and B is well defined if A and B have the same number of rows and the same number of columns. If A and B are two m × n matrices, then the ***matrix sum*** of A and B, denoted A + B, is also an m × n matrix such that $(A + B)_{i,j} = A_{i,j} + B_{i,j}$ for all $1 \le i \le m$ and $1 \le j \le n$.

Figure 6.6.3: Matrix addition.

$$\underset{A}{\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}} + \underset{B}{\begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}} = \underset{A+B}{\begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}}$$

Matrix addition is computed by adding the corresponding entries in each array. For example, the entry in row 2, column 1 of A + B is $A_{2,1} + B_{2,1}$ as highlighted in blue. Boolean addition is used in adding entries of Boolean matrices.

Theorem 6.6.2: Addition and graph union.

Let G and H be two directed graphs with the same vertex set. Let A be the adjacency matrix for G and B the adjacency matrix for H. Then the adjacency matrix for G ∪ H is A + B, where Boolean addition is used on the entries of matrices A and B.

PARTICIPATION ACTIVITY 6.6.10: Boolean matrix addition and graph union.

Animation captions:

1. A is the adjacency matrix for graph G. B is the adjacency matrix for H.
2. The adjacency matrix for G ⋃ H is the Boolean sum of A and B.
3. (j,k) is an edge in G ⋃ H if (j,k) is an edge in G or an edge in H. Entry (j,k) is 1 in A+B if there is a 1 in entry (j,k) of A or B.

The union of two graphs defined on the same set of vertices is a single graph whose edges are the union of the edge sets of the two graphs. Graph union can be computed using matrix addition.

PARTICIPATION ACTIVITY 6.6.11: Addition and graph union.

The Boolean sum of matrices A and B is given below with a few entries missing.

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \quad A + B = \begin{pmatrix} 1 & ? & 0 \\ 0 & 1 & 1 \\ 1 & 1 & ? \end{pmatrix}$$

1) What is the value of the entry in row 1, column 2 of matrix A + B?

Check Show answer

2) What is the value of the entry in row 3, column 3 of matrix A + B?

Check Show answer

3) A is the adjacency matrix for graph G and B is the adjacency matrix for graph H. Is there an edge in G ∪ H from vertex 1 to vertex 3? Type yes or no.

Check Show answer

4) A is the adjacency matrix for graph G and B is the adjacency matrix for graph H. Is there an edge in G ∪ H from vertex 3 to vertex 1? Type yes or no.

Check Show answer

Finally, Boolean matrix multiplication and addition can be put together to compute the adjacency matrix A^+ for G^+, the transitive closure of G:

$$G^+ = G^1 \cup G^2 \cup \cdots \cup G^n$$
$$A^+ = A^1 + A^2 + \cdots + A^n$$

There is 1 in row v, column u of A^+ if and only if there is a walk of any length from v to u in G.

PARTICIPATION ACTIVITY 6.6.12: Computing the transitive closure of a graph by matrix operations.

The matrix A given below is the adjacency matrix for a directed graph G with 4 vertices. The matrix A^+ is computed by calculating A^2 , A^3 and A^4 using matrix multiplication and then computing the matrix sum $A + A^2 + A^3 + A^4$. The rows and columns of the matrices are numbered 1 through 4 corresponding to the vertices in G.

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad A^+ = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

1) Is there a walk in G that starts at vertex 4 and ends at vertex 2? Type yes or no.

Check Show answer

2) Is there a walk in G that starts at vertex 2 and ends at vertex 4? Type yes or no.

Check Show answer

3) How many vertices can be reached by a walk in G starting at vertex 4?

Check Show answer

Proof 6.6.1: Relationship between the powers of a graph and the powers of its adjacency matrix.

Theorem: Let G be a directed graph with n vertices and let A be the adjacency matrix for G. Then for any $k \geq 1$, A^k is the adjacency matrix of G^k, where Boolean addition and multiplication are used to compute A^k.

Proof.

The proof is by induction on k. For the base case, k = 1. By definition $G^1 = G$, and $A^1 = A$ is the adjacency matrix for G.

Now assume that A^{k-1} is the adjacency matrix for G^{k-1}, and prove that A^k is the adjacency matrix for G^k. Since A^{k-1} is the adjacency matrix for G^{k-1}, $(A^{k-1})_{i,j}$ is 1 if and only if there is a walk in graph G of length k - 1 from vertex i to vertex j. We will show that $(A^k)_{i,j}$ is 1 if and only if there is a walk of length k in G from vertex i to vertex j.

Suppose that $(A^k)_{i,j} = 1$. Since $A^k = A^{k-1} \cdot A$, by definition of matrix multiplication,

$$(A^k)_{i,j} = (A^{k-1})_{i,1}A_{1,j} + (A^{k-1})_{i,2}A_{2,j} + \cdots + (A^{k-1})_{i,n}A_{n,j}$$

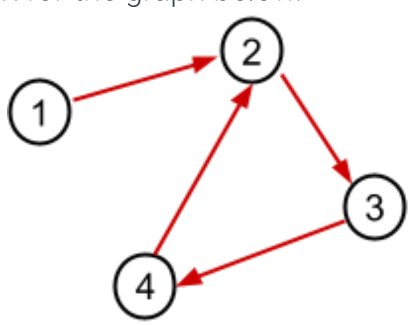
Since $(A^k)_{i,j} = 1$, there is at least one x in the range from 1 through n such that $(A^{k-1})_{i,x} \cdot A_{x,j} = 1$. For that value of x, $(A^{k-1})_{i,x} = 1$ and $A_{x,j} = 1$. Since $(A^{k-1})_{i,x} = 1$, by the inductive hypothesis, there is walk of length k-1 from vertex i to vertex x in G. Since $A_{x,j} = 1$, there is an edge in G from vertex x to vertex j. The length k-1 walk from i to x, followed by the edge from x to j, results in a walk of length k in G from vertex i to vertex j.

Now suppose that there is a walk of length k in G from vertex i to vertex j. Let x be the second-to-last vertex visited on the walk. There is a walk of length k-1 in G from vertex i to vertex x and an edge in G from vertex x to vertex j. By induction, $(A^{k-1})_{i,x} = 1$ and $A_{x,j} = 1$. Therefore, by definition of Boolean matrix multiplication, $(A^k)_{i,j} = 1$. ■

Additional exercises

EXERCISE | 6.6.1: Adjacency matrices for graph powers and the transitive closure via matrix multiplication.

(a) Give the adjacency matrix for the graph below.

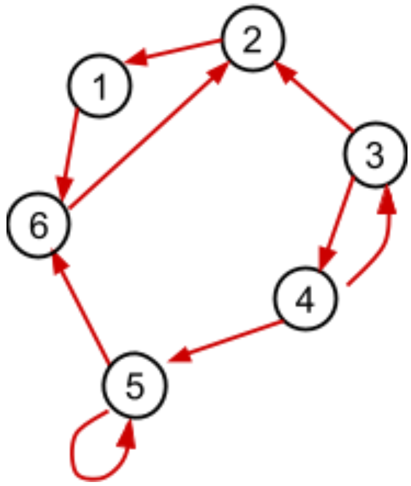


Then use matrix multiplication to find the adjacency matrices for G^2, G^3, G^4, and G^+.

Solution

EXERCISE | 6.6.2: Interpreting the adjacency matrix for a graph power.

A graph G is pictured below:



(a) Give the adjacency matrix for G.

(b) Use matrix multiplication to find the adjacency matrix for G^2.

(c) Which vertices can be reached from vertex 4 by a walk of length 2?

(d) Which vertices can reach vertex 2 by a walk of length 2?

EXERCISE 6.6.3: Inferring information about a graph from matrices of graph powers.

A directed graph G has 5 vertices, numbered 1 through 5. The 5×5 matrix A is the adjacency matrix for G. The matrices A^2 and A^3 are given below.

$$A^2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{pmatrix} \qquad A^3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Use the information given to answer the questions about the graph G.

(a) Which vertices can reach vertex 2 by a walk of length 3?

(b) What is the out-degree of vertex 4 in the transitive closure of G?

(c) Is there a walk of length 4 from vertex 4 to vertex 5 in G? (Hint: $A^4 = A^2 \cdot A^2$.)

(d) Is (2, 2) in the transitive closure of G?

(e) Is (5, 3) an edge in G^3?

Solution

(f) Is there a closed walk of length 3 in G?

Solution

EXERCISE 6.6.4: Inferring information about walks from matrices of graph powers.

(a) Suppose that matrix A is the adjacency matrix of a graph G with seven vertices, numbered 1 through 7. The matrices A and A^3 are given below but only some of the entries are visible:

$$A^3 = \begin{pmatrix} * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \end{pmatrix} \qquad A = \begin{pmatrix} * & * & * & * & * & * & 0 \\ * & * & * & * & * & * & 0 \\ * & * & * & * & * & * & 0 \\ * & * & * & * & * & * & 1 \\ * & * & * & * & * & * & 1 \\ * & * & * & * & * & * & 1 \\ * & * & * & * & * & * & 0 \end{pmatrix}$$

Is there a walk of length 4 from vertex 5 to vertex 7 in G? If so, is it possible to infer the second-to-last vertex in that walk?

EXERCISE 6.6.5: Products of adjacency matrices of graph powers.

(a) Suppose that G is a graph with 10 vertices. Let S be the adjacency matrix for G^3 and T be the adjacency matrix for G^4. Describe how the matrix S·T relates to the graph G.

Solution

6.7 Equivalence relations

A relation R is an ***equivalence relation*** if R is reflexive, symmetric, and transitive. If a relation R is an equivalence relation, the notation a~b is used to express aRb. a~b is read "a is equivalent to b".

Consider an example where the domain is the set of all people. Define relation B such that xBy if person x and person y have the same birthday. The relation B is reflexive since every person has the same birthday as himself/herself. The relation is symmetric because if x has the same birthday as y, then y has the same birthday as x. The relation is also transitive because if x and y share a birthday and y and z share a birthday, then x and z must also share a birthday.

PARTICIPATION ACTIVITY | 6.7.1: Equivalence relation example.

Animation captions:

1. R is reflexive (all self-loops are present).
2. R is symmetric (if (x,y) is an edge, then (y,x) is also an edge).
3. R is transitive (edges (e,a) and (a,b) imply the presence of edge (e,b)). Therefore, R is an equivalence relation.
4. If an edge (e,d) is added, then R is no longer transitive: (b,e) and (e,d) are edges, but there is no edge from b to d.

PARTICIPATION ACTIVITY | 6.7.2: Identifying equivalence relations.

Select the choice that applies to each relation.

1) The domain of R is a group of employees at a company. xRy if x earns at least as much money as y. There are at least two employees at the company who do not earn the same amount of money.

- R is an equivalence relation.
- Not an equivalence relation because R is not symmetric.
- Not an equivalence relation because R is not transitive.

2) The domain of relation R is the set of integers. xRy if $x^2 = y^2$

- R is an equivalence relation.
- R is not an equivalence relation because R is not reflexive.
- R is not an equivalence relation because R is not transitive.

3) The domain of relation R is the set of integers greater than 1. xRy if a positive integer other than 1 evenly divides both x and y.

- R is an equivalence relation.

- R is not an equivalence relation because R is not reflexive.
- R is not an equivalence relation because R is not transitive.

If A is the domain of an equivalence relation and a ∈ A, then [a] is defined to be the set of all x ∈ A such that a~x. The set [a] is called an ***equivalence class***. Consider an example in which the domain is the set of natural numbers and x~y if x and y have the same remainder when divided by 3.

> [0] is the set of all natural numbers whose remainder is 0 when divided by 3.
> [1] is the set of all natural numbers whose remainder is 1 when divided by 3.
> [2] is the set of all natural numbers whose remainder is 2 when divided by 3.

Equivalence relations have a special mathematical structure, described in the following theorem which says that two equivalence classes are either identical or completely disjoint.

Theorem 6.7.1: Structure of equivalence relations.

Consider an equivalence relation on a set A. Let x,y ∈ A:

- If x~y then [x] = [y]
- If it is not the case that x~y, then [x] ∩ [y] = ∅

Proof 6.7.1: Structure of equivalence relations.

Proof.
Suppose that x~y. Pick an element z ∈ [y]. It follows that y~z. Since x~y and y~z, it follows from transitivity that x~z, which in turn means that z ∈ [x]. This establishes that any element in [y] must also be in [x]. Since x~y and the relation is symmetric, we know that y~x. The same argument can then be used to show that any element in [x] must also be in [y]. Thus, [x] = [y].

The proof of the second part of the theorem is a proof by contradiction. Suppose that it is not the case that x~y, but there is some z ∈ [x] ∩ [y]. Since z ∈ [x] ∩ [y], it must be the case that x~z and y~z. By symmetry, y~z implies that z~y. Since x~z and z~y, it follows from transitivity that x~y, which contradicts the assumption. ■

The theorem can be used to show that an equivalence relation defines a partition of the domain. A ***partition*** of a set A is a set of non-empty subsets of A that are pairwise disjoint and whose union is A.

Theorem 6.7.2: Equivalence relations define a partition.

Consider an equivalence relation over a set A. The set of all distinct equivalence classes defines a partition of A. The term "distinct" means that if there are two equal equivalence classes [a] = [b] the set [a] is only included once.

Proof 6.7.2: Equivalence relations define a partition.

Proof.
By the definition of an equivalence class, if $a \in A$, then $[a] \subseteq A$. Since $a \sim a$ for every $a \in A$, $a \in [a]$. Therefore every element in A is included in at least one set in the partition which implies that the union of all the equivalence classes is the set A. The previous theorem established that if $[a] \neq [b]$, then $[a] \cap [b] = \varnothing$. Therefore, the set of distinct equivalence classes are pairwise disjoint. A set of sets is ***pairwise disjoint*** if the intersection of any pair of the sets is empty. ■

PARTICIPATION ACTIVITY 6.7.3: Equivalence relations and partitions.

Animation captions:

1. Equivalence relation R defines a partition on the underlying set A. All possible edges are present between the elements in each set of the partition.

Consider an equivalence relation on the vertices of a communication network. Suppose that two vertices have a direct communication link between them if they are within a certain distance of each other. Define the relation so that x~y if x can send a message to y, either directly or indirectly via a series of hops through other vertices in the network. For a given placement of the vertices, it may or may not be the case that each pair of vertices in the network can communicate with each other. The relation is reflexive because each vertex can communicate with itself. The relation is symmetric because any sequence of hops from x to y can be reversed so that y can send a message to x. The relation is also transitive because if x can send a message to y and y can send a message to z, then x can send a message to z through y. The theorem above says that the vertices of the network can be partitioned into sets of vertices that can all communicate with each other. Any two vertices in the same set can communicate with each other. Any two vertices from different sets can not communicate with each other.

PARTICIPATION ACTIVITY | 6.7.4: Equivalence relations and partitions.

Let A = { a, b, c, d, e }. Consider the equivalence relation R on A:
{ (c, d), (a, e), (a, a), (b, b), (d, c), (c, c), (e, a), (d, d), (e, e) }.

1) What is the partition defined by the equivalence relation R?

- {c, b}, {d, a}, {e}
- {c, d}, {a, e}, {b}.
- {c, b, e}, {d, a}

Additional exercises

EXERCISE | 6.7.1: Recognizing equivalence relations.

Determine whether each relation is an equivalence relation. Justify your answer. If the relation is an equivalence relation, then describe the partition defined by the equivalence classes.

(a) The domain is a group of people. Person x is related to person y under relation P if x and y have a common parent (i.e., x and y have the same biological mother or the same biological father or both). You can assume that there is at least one pair in the group, x and y, such that xPy.

Solution

(b) The domain is a group of people. Person x is related to person y under relation M if x and y have the same biological mother. You can assume that there is at least one pair in the group, x and y, such that xMy.

(c) The domain is a group of people. Person x is related to person y under relation S if x is currently the spouse of person y. You can assume that there is no polygamy and that there is at least one person who is married to another person in the group.

Solution

(d) The domain is the set of all integers. xEy if x + y is even. An integer z is even if z = 2k for some integer k.

(e) The domain is the set of all integers. xOy if x + y is odd. An integer z is odd if z = 2k + 1 for some integer k.

Solution

EXERCISE 6.7.2: Equivalence classes for remainders of integer division.

(a) The domain of the equivalence relation D is the set S:

$$S = \{7, 2, 13, 44, 56, 34, 99, 31, 4, 17\}$$

For any x, y ∈ S, xDy if x has the same remainder as y when divided by 4. Show the partition of S defined by the equivalence classes of D.

EXERCISE 6.7.3: Equivalence relations and transitive closures.

(a) Prove that the transitive closure of a symmetric relation is also symmetric.

Solution

(b) Use the result from the previous problem to argue that if P is reflexive and symmetric, then P^+ is an equivalence relation.

(c) Is it possible to have a relation R that is symmetric but R^+ is not an equivalence relation?

EXERCISE 6.7.4: Equivalence relations on strings.

D = $\{0, 1\}^6$. The following relations have the domain D. Determine if the following relations are equivalence relations or not. Justify your answers.

(a) Define relation R: xRy if y can be obtained from x by swapping any two bits.

Solution

(b) Define relation R: xRy if y can be obtained from x by reordering the bits in any way.

Solution

EXERCISE 6.7.5: Equivalence relations on numbers.

The domain of the following relations is the set of all integers. Determine if the following relations are equivalence relations. Justify your answers.

(a) xRy if $x - y = 3m$ for some integer m.

(b) xRy if $x + y = 3m$ for some integer m.

6.8 Strict orders and directed acyclic graphs

A partial order acts similar to the ≤ operator on the elements of its domain. A strict order acts similar to the < operator on the elements of its domain. A relation R is a ***strict order*** if R is transitive and anti-reflexive. The notation a < b is used to express aRb and is read "a is less than b". The difference between the < and the < symbols is the slight curve in the < symbol. The domain along with the strict order defined on it is called a strictly ordered set and is denoted by (A, <).

The arrow diagram for a strict order is basically an arrow diagram for a partial order without the self loops. The definitions for a partial order carry over in a natural way to strict orders. Two elements, x and y, are said to be ***comparable*** if x < y or y < x. Otherwise, the elements are said to be ***incomparable***. A strict order is a ***total order*** if every pair of elements is comparable. An element x is a ***minimal*** element if there is no y such that y < x. An element x is a ***maximal*** element if there is no y such that x < y.

PARTICIPATION ACTIVITY | 6.8.1: Strict orders.

Animation content:

undefined

Animation captions:

1. The relation P is anti-symmetric, transitive, and reflexive, so P is a partial order. The reflexive property means that the arrow diagram has a self-loop at each element.
2. When the self-loops are removed, P becomes a strict order.
3. P is anti-reflexive (no self-loops are present).
4. P is transitive (edges (a,c) and (c,e) imply the presence of edge (a,e)). Therefore, P is a strict order.
5. a and d are minimal elements because there are no arrows into a or d.
6. e and f are maximal elements because there are no arrows out of e or f.

Example 6.8.1: Strict order examples.

- The real numbers along with the < relation is a strict order. The relation is transitive: if a < b and b < c, then a < c. Furthermore, the relation is anti-reflexive because there is no real a such that a < a.
- If A is a finite set, then (P(A), ⊂) is a strict order. The domain is P(A), the set of all subsets of A. Two subsets of A, X and Y, are related if X ⊂ Y. Recall that the notation X ⊂

Y means that X is a strict subset of Y, that is, X is a subset of Y and $X \neq Y$. The relation $\subset$ is transitive: if $X \subset Y$ and $Y \subset Z$, then $X \subset Z$. The relation $\subset$ is also anti-reflexive, because a set X can not be a strict subset of itself.

The definition of strict order does not include the condition that the relation be anti-symmetric, even though every strict order is in fact anti-symmetric. However, any relation that is anti-reflexive and transitive, must also be anti-symmetric:

Fact 6.8.1: Strict orders are anti-symmetric.

Consider a relation R that is transitive and anti-reflexive. Then R is also anti-symmetric.

Proof 6.8.1: Strict orders are anti-symmetric.

Proof.
Suppose by contradiction that domain contains two elements, $x \neq y$, such that xRy and yRx. Suppose that R is also anti-reflexive and transitive. Since R is transitive, then the fact that xRy and yRx implies that xRx. The fact that xRx contradicts the assumption that R is anti-reflexive. ▪

PARTICIPATION ACTIVITY 6.8.2: Identifying strict orders and partial orders.

Each question below describes a relation R whose domain is the set of positive integers. Select the choice that best describes the relation.

1) xRy if y divided by x is a prime number.

- R is a strict order.
- R is a partial order.
- R is neither a strict order nor a partial order.

2) xRy if y divided by x is an odd integer.

- R is a strict order.
- R is a partial order.

- R is neither a strict order nor a partial order.

3) xRy if y divided by x is an even integer.

- R is a strict order.
- R is a partial order.
- R is neither a strict order or a partial order.

Directed acyclic graphs

Strict orders are closely related to an important class of graphs called directed acyclic graphs.

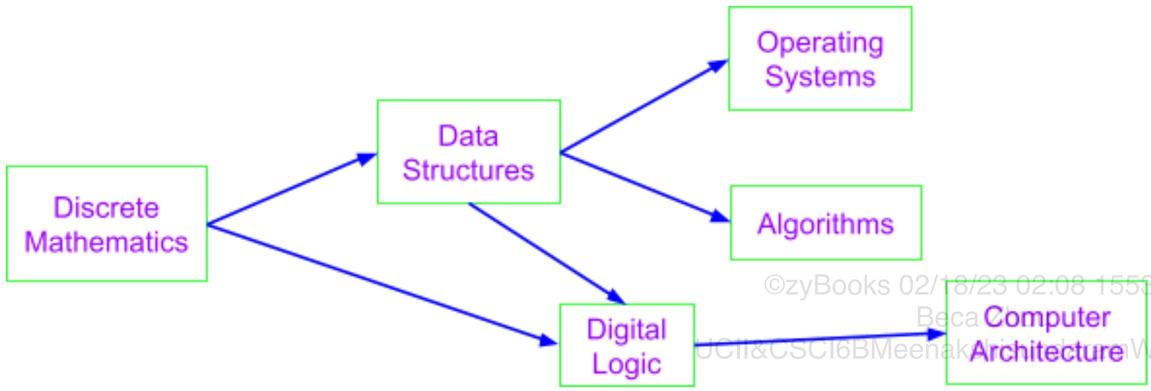
Definition 6.8.1: Definition of a DAG.

A ***directed acyclic graph*** (or ***DAG*** for short) is a directed graph that has no cycles.

Note that a cycle always has length at least one, so therefore has at least one edge.

DAGs are particularly useful for representing precedence relationships. For example, consider the set of courses required for a degree in Computer Science. (Most CS programs have some choices in courses, but for simplicity we will assume that there is a fixed set of courses required for the degree). The catalog lists the descriptions of the courses as well as a set of prerequisites: courses that must be taken before enrolling in the course in question. The prerequisite structure can be represented by a DAG. The set of vertices corresponds to the set of required courses. There is a directed edge from course a to course b if the catalog lists course a as a prerequisite for course b. Of course, it is important that the prerequisite graph not have any cycles. Otherwise, it would be impossible to complete the requirements for the major. The diagram below shows some of the courses that might be required for a computer science degree and their prerequisites:

Figure 6.8.1: A DAG corresponding to course prerequisites.



The relation corresponding to the prerequisite graph is not necessarily transitive, so there are also implicit prerequisites that arise as the results of paths in the graph. For example, in the diagram above, consider Discrete Math, Data Structures, and Algorithms. If Discrete Math is a prerequisite for Data Structures and Data Structures is a prerequisite for Algorithms, then Discrete Math must be taken before a student can take Algorithms. The entire set of constraints (explicit and implicit) are expressed by G^+, where G is the DAG corresponding to prerequisites mentioned specifically in the catalog. It turns out, that if G is a directed acyclic graph, then G^+ is a strict order. In fact the converse is also true:

Theorem 6.8.1: Directed acyclic graphs and strict orders.

Let G be a directed graph. G has no cycles if and only if G^+ is a strict order.

Proof 6.8.2: Directed acyclic graphs and strict orders.

Proof.
If G has a cycle, then there is a path $\langle v_0,\ldots,v_k\rangle$ in G where $v_0 = v_k$ and $k \geq 1$. The existence of the path of length k implies that there is an edge (v_0, v_k) in G^k. All the edges in G^k are included in G^+. Since $v_0 = v_k$, the edge is a self-loop, implying that G^+ cannot be anti-reflexive. Therefore, G^+ is not a strict order.

G^+ is by definition transitive, so in order for it not to be a strict order, it must not be anti-reflexive which means that it has a self-loop. Suppose that G^+ has a self loop at vertex v. The edge (v, v) must be in G^k for some $k \geq 1$. The existence of the edge (v, v) in G^k means that there is a path of length k in G that begins and ends at vertex v. Thus, there is a cycle of length k in G and G is not acyclic. ■

If a DAG G is transitive, then $G^+ = G$. Thus, the theorem above implies that a directed graph G is a strict order if and only if G is acyclic and transitive.

If G is a directed acyclic graph and G^+ is the transitive closure of G, then the minimal elements in G^+ are exactly the vertices with in-degree 0 in G. Similarly, the maximal elements in G^+ are the vertices with out-degree 0 in G.

PARTICIPATION ACTIVITY | 6.8.3: Directed acyclic graphs.

Animation content:

undefined

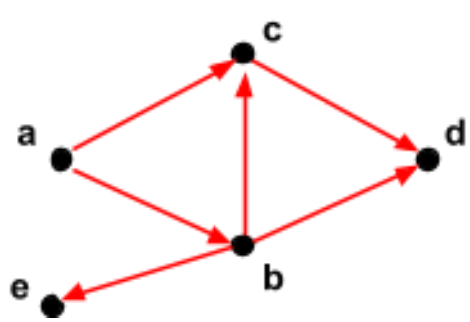
Animation captions:

1. G is a DAG because G has no cycles. A DAG is not necessarily transitive: (a,b) and (b,e) are edges, but there is no edge from a to e.
2. a and h are minimal elements because their in-degree is 0. g, f, and j are maximal elements because their out-degree is 0.
3. G^+ is obtained by adding edges to G to make G transitive. G^+ is then a strict order.

PARTICIPATION ACTIVITY | 6.8.4: Identifying DAGs and strict orders.

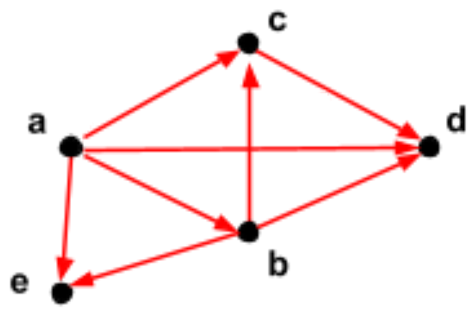
Each question below shows a graph. Select the choice that best describes the graph.

1)



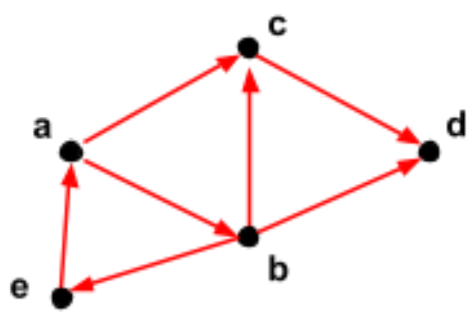
- G has a cycle.
- G is acyclic but not a strict order.
- G is a strict order.

2)



- G has a cycle.
- G is acyclic but is not a strict order.
- G is a strict order.

3)



- G has a cycle.
- G is acyclic but is not a strict order.
- G is a strict order.

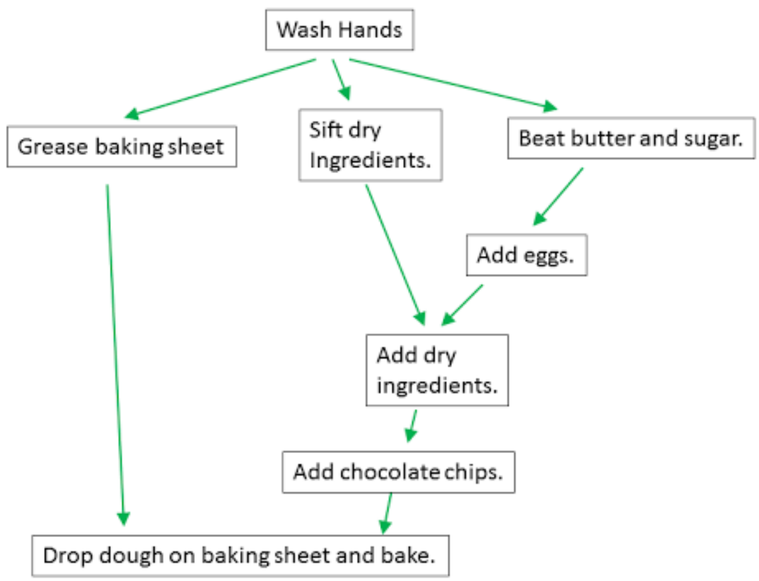
Example 6.8.2: Precedence constraints for project planning.

Complex jobs are often broken into smaller tasks for purposes of planning and scheduling. For example, large pieces of software are often produced by teams of many people whose work needs to be carefully coordinated. The software development process can be broken up into smaller tasks which a project manager assigns to individuals or teams. The project manager's job is to coordinate the completion of the tasks and make projections for the timeline of the entire project. Sometimes the results of one task are needed before another task can begin. The project manager would use a directed acyclic graph to express the dependencies between individual tasks. For tasks x and y, (x, y) is a directed edge in the graph if task x must be completed before task y begins. Analyzing the resulting DAG is important for scheduling and planning the execution of the project.

To illustrate the process, here is an example of simpler project: baking chocolate chip cookies. The individual tasks are as follows:

- Wash hands.
- Grease cooking sheet.
- Sift together dry ingredients.
- Beat together butter and sugar.
- Add eggs to butter and sugar.
- Add dry ingredients to butter, sugar and eggs.
- Add chocolate chips.
- Drop spoonfuls of batter onto cookie sheet and bake.

Below is a directed acyclic graph showing the dependencies between tasks. Note that incomparable tasks could be done simultaneously by different people.

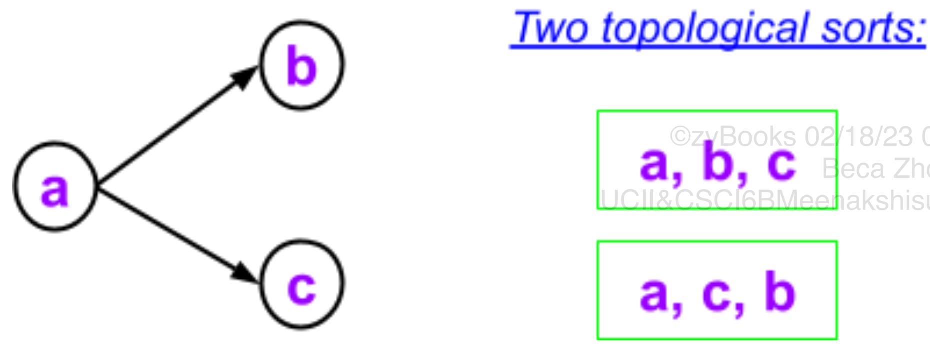


Topological sorts of DAGs

Consider a situation like the example above in which a directed acyclic graph represents precedence constraints for a set of tasks. If the set of tasks will be completed sequentially (one at a time), then we would like to find an ordering of the tasks that does not violate any of the precedence constraints. A ***topological*** sort for a DAG is an ordering of the vertices that is consistent with the edges in the graph. That is, if there is an edge (u, v), then u appears earlier than v in the topological sort. For example, if a student completing a computer science major can only take one course per semester, then the order in which she takes the courses must be a topological sort of the vertices (courses) in the prerequisite graph.

A topological sort for a DAG G is also a topological sort for G^+. A DAG always has at least one topological sort and may have many more than one. The diagram below shows a small DAG and two possible topological sorts:

Figure 6.8.2: A DAG and two different topological sorts.



One way to construct a topological sort for a DAG G is to pick a vertex x with in-degree 0 and remove x from G. When a vertex is removed from the graph, all the edges going into or out of that vertex are also removed. Then pick another vertex with in-degree 0 from among the remaining vertices. Keep selecting vertices until there are no vertices left. The animation below illustrates this method of constructing a topological sort, using the tasks from the baking cookies example above:

PARTICIPATION ACTIVITY 6.8.5: Topological sort example.

Animation content:

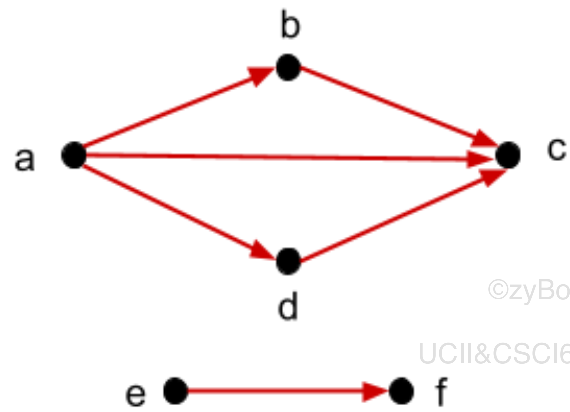
undefined

Animation captions:

1. To find a topological sort of the DAG, select a vertex with in-degree 0. The only vertex with in-degree 0 is "Wash".
2. At each step, until the graph is empty, select a vertex with in-degree 0, remove the vertex from the graph and add the vertex to the topological sort.

PARTICIPATION ACTIVITY 6.8.6: Topological sorts.

The figure below shows DAG G:



1) If the vertex a is selected to be the first element in a topological sort of G, then which three elements are candidates to be the second vertex selected?

- {b, d, f}
- {b, d, e}
- {a, e, b}

2) Which one of the sequences is a topological sort for G?

- (a, b, e, c, f, d)
- (a, e, f, b, d, c)
- (a, b, f, d, c, e)

Additional exercises

EXERCISE 6.8.1: Identifying partial, strict, and total orders.

For each relation, indicate whether the relation is a partial order, a strict order, or neither. If the relation is a partial or strict order, indicate whether the relation is also a total order. Justify your answers.

(a) The domain is the set of all words in the English language (as defined by, say, Webster's dictionary). Word x is related to word y if x appears before y in alphabetical order. Assume that each word appears exactly once in the dictionary.

(b) The domain is the set of all words in the English language (as defined by, say, Webster's dictionary). Word x is related to word y if x appears as a substring of y. x is a substring of y if all the letters in x appear in consecutive order somewhere in y. For example, "logical" is substring of "topological" because the letters l-o-g-i-c-a-l appear consecutively in order in the word "topological". However, "local" is not a substring of "topological" because the letters l-o are separated from c-a-l by the letters g and i.

(c) The domain is the set of all cell phone towers in a network. Two towers can communicate if they are within a distance of three miles from each other. Tower x is related to tower y if x can send information to y through a path of communication links. You can assume that there are at least two towers that are within three miles of each other.

(d) The domain is the set of all positive integers. x is related to y if y = 3·n·x, for some positive integer n.

(e) The domain of relation P is the set of all positive integers. For x, y ∈ $\mathbf{Z}^+$, xPy if there is a positive integer n such that $x^n = y$.

(f) The domain for the relation is **Z**×**Z**. (a, b) is related to (c, d) if a ≤ c and b ≤ d.

(g) The domain is the set of girls at a basketball camp. Player x is related to y if x is taller or weighs more than player y (inclusive or). You can assume that no two players have the same height and that no two players have the same weight. The answer may depend on the actual weights or heights of the players, in which your answer may be "not necessarily", but you need to give an example to justify your answer.

(h) The domain is the set of all runners in a race. x is related to y if x beat y in the race. No two players tied.

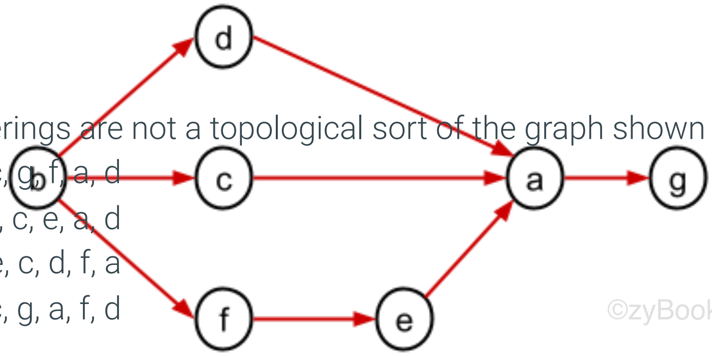
(i) The domain is the set of all runners in a race. x is related to y if x beat y in the race. At least two runners in the race tied.

(j) S = {a, b, c, d}. The domain is P(S), the power set of S. For X, Y that are subsets of S, X is related to Y if |X| ≤ |Y|.

(k) S = {a, b, c, d}. The domain is P(S), the power set of S. For X, Y that are subsets of S, X is related to Y if |X| < |Y|.

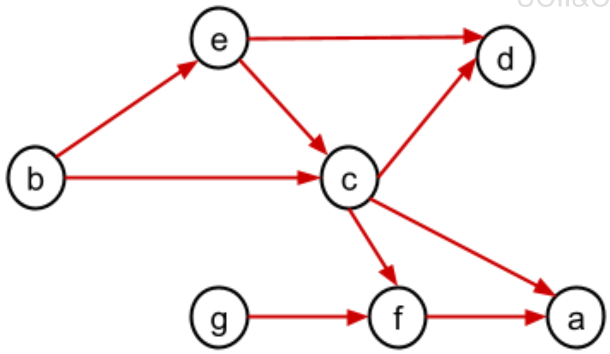
EXERCISE 6.8.2: Topological sort of a DAG.

(a) Give two different topological sorts of the directed acyclic graph shown below.



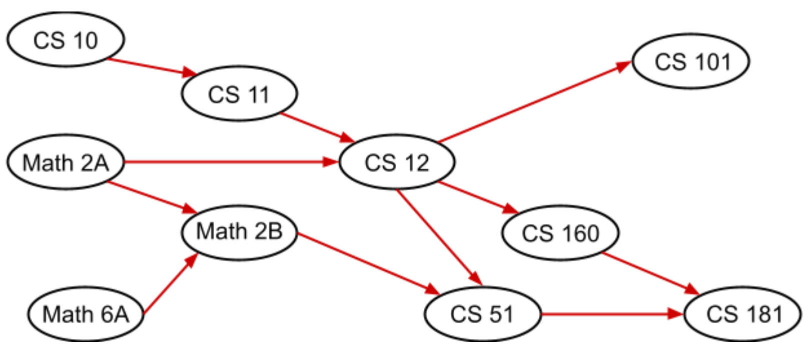
(b) Which orderings are not a topological sort of the graph shown below?

b, e, c, g, f, a, d
b, g, f, c, e, a, d
b, g, e, c, d, f, a
b, e, c, g, a, f, d



EXERCISE 6.8.3: Academic plan with course prerequisites.

Below is a set of required courses for a degree. The directed acyclic graph shows the prerequisite structure for the courses.



Your job is to devise an academic plan for a student. An academic plan is a set of courses for the student to take in each quarter. In each case, you need to devise an academic plan which respects the prerequisites and takes the fewest number of quarters. You can assume that every course is offered in every quarter.

(a) Devise an academic plan for the student if he can only take one course per quarter.

(b) Devise an academic plan for the student if he can take up to two courses per quarter.

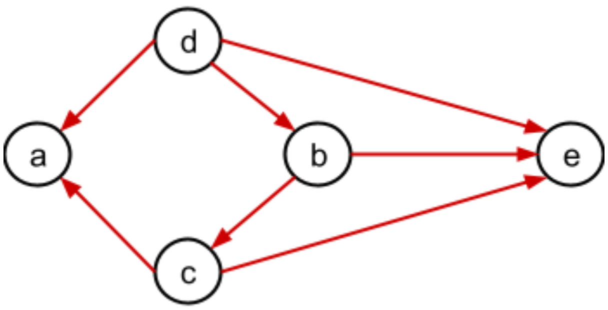
(c) Devise an academic plan for the student if he can take up to three courses per quarter.

(d) What's the fewest number of quarters the degree will take if the student can take an unlimited number of courses per quarter?

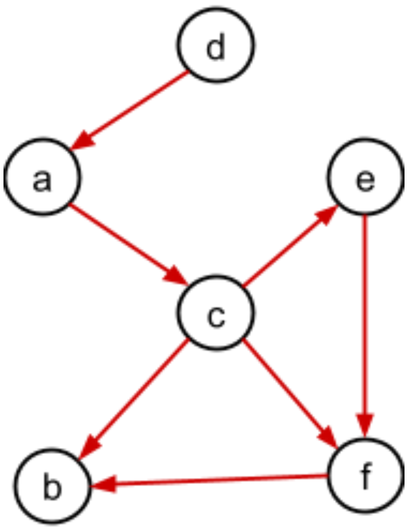
EXERCISE 6.8.4: Identifying DAGs and finding transitive closures.

Which of the following graphs are acyclic? For each graph that is acyclic, give the transitive closure for the graph.

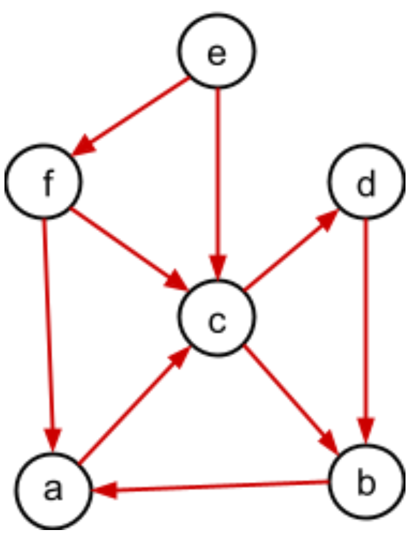
(a)



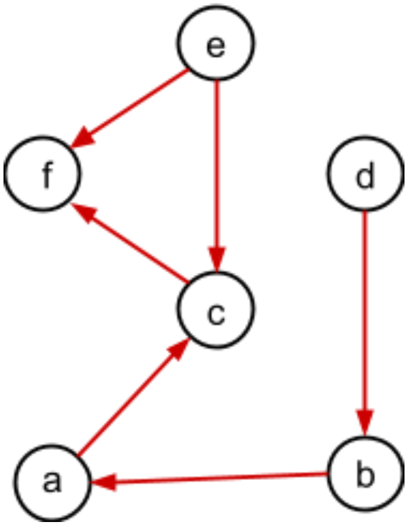
(b)



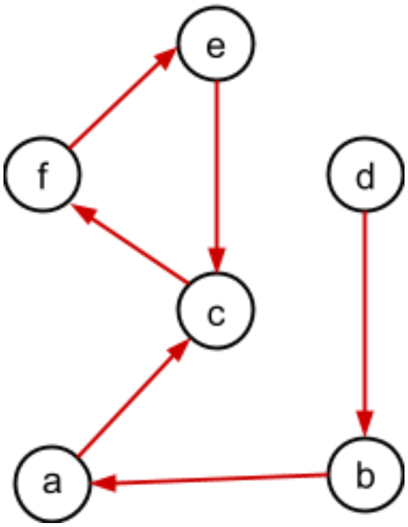
(c)



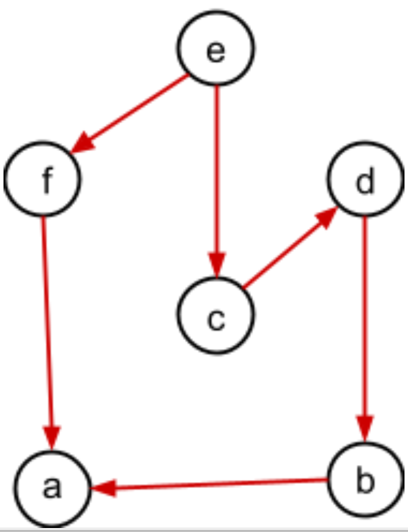
(d)



(e)



(f)



6.9 Partial orders

A relation R on a set A is a ***partial order*** if it is reflexive, transitive, and anti-symmetric. The notation a ≤ b is used to express aRb, reflecting the fact that a partial order acts like a ≤ operator on the elements of A. The expression a ≤ b is read "a is at most b". The difference between the ≤ and the ≤ symbols is the slight curve in the ≤ symbol. The domain along with a partial order defined on it is denoted (A, ≤) and is called a ***partially ordered set*** or ***poset***.

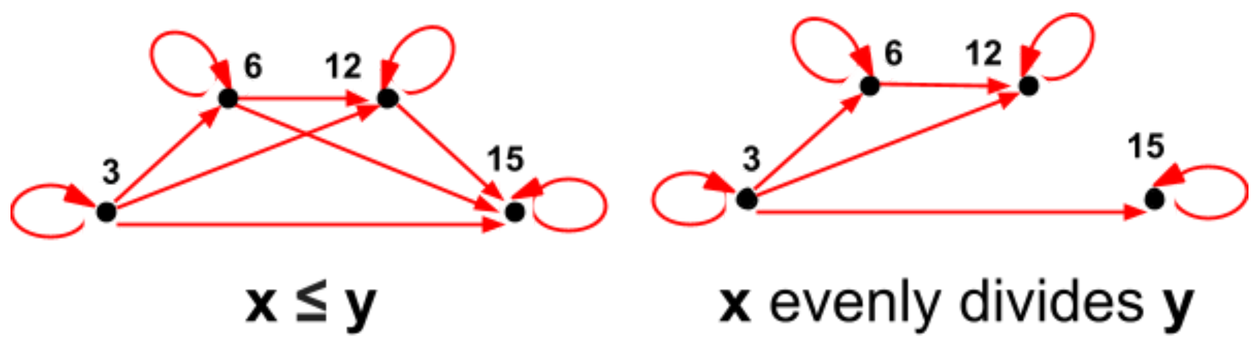
For example the ≤ operator acting on the set of integers is a partial order, denoted by (**Z**, ≤). The relation is reflexive (x ≤ x) and anti-symmetric (if x ≤ y and y ≤ x then x = y). The relation is also transitive (x ≤ y and y ≤ z imply that x ≤ z).

Here is another example of a partial order: the domain is the set of natural numbers and x ≤ y if x evenly divides y. The pair (**N**, ≤) satisfies the conditions for a partial order:

- x evenly divides itself (reflexive)
- If x evenly divides y and y evenly divides x, then x = y (anti-symmetric)
- If x evenly divides y and y evenly divides z, then x evenly divides z (transitive)

The diagram below gives the arrow diagram for the two partial order examples restricted to the set {3, 6, 12, 15}:

Figure 6.9.1: Arrow diagrams for two partial orders.



The two partial orders depicted above have noticeably different structure. In the example with the ≤ operator, every pair of elements are related to each other in some way. That is, either x ≤ y or y ≤ x, or in the case that x = y, then both x ≤ y and y ≤ x hold. However, in the partial order on the right neither 12 ≤ 15 nor 15 ≤ 12 hold. Two elements of a partially ordered set, x and y, are said to be ***comparable*** if x ≤ y or y ≤ x. Otherwise they are said to be ***incomparable***. A partial order is a ***total order*** if every two elements in the domain are comparable. The partial order (**Z**, ≤) is an example of a total order.

An element x is a ***minimal*** element if there is no y ≠ x such that y ≤ x. An element x is a ***maximal*** element if there is no y ≠ x such that x ≤ y.

The following animation gives an example of a partial order set and reviews the definitions introduced so far:

PARTICIPATION ACTIVITY 6.9.1: Partially ordered set example.

Animation content:

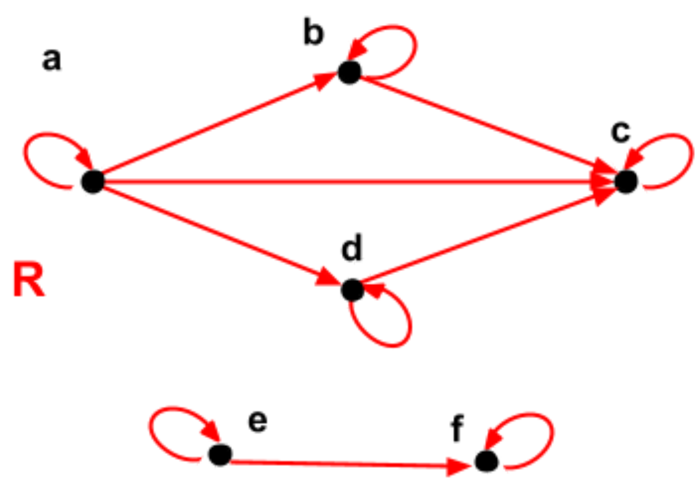
undefined

Animation captions:

1. P is reflexive (all self-loops are present).
2. P is anti-symmetric (there are no two elements that point to each other).
3. P is transitive (edges (a,c) and (c,e) imply the presence of edge (a,e)). Therefore, P is a partial order.
4. a and d are minimal elements because there are no arrows into a or d except the ones from themselves.
5. e and f are maximal elements because the only edges leaving e and f point to themselves.

PARTICIPATION ACTIVITY 6.9.2: Properties of partial orders.

The figure below shows an arrow diagram of a partial order R:



1) Are c and f comparable?

- Yes
- No

2) Is R a total order?

- Yes
- No

3) What are the minimal elements in the example?

- a and e

- b and d
- c and f

4) What are the maximal elements in the example?
   - a and e
   - b and d
   - c and f

5) If the pair (d, f) is added, is R still a partial order?
   - Yes
   - No

6) From the original arrow diagram, if the pair (a, f) is added, is R still a partial order?
   - Yes
   - No

PARTICIPATION ACTIVITY 6.9.3: Recognizing partial orders.

Are the following relations partial orders?

1) The domain is a group of brothers and sisters.
   $x \preceq y$ if y is at least as old as x. You can assume that all the brothers and sisters have the same mother, so no two of them were born at exactly the same time.
   - Yes, it is a partial order.
   - No, it is not a partial order because it is not reflexive.
   - No, it is not a partial order because it is not anti-symmetric.

2) The domain is the set of people working at a company. $x \preceq y$ if y has a higher salary than x.

- Yes, it is a partial order.
- No, it is not a partial order because it is not transitive.
- No, it is not a partial order because it is not reflexive.

3) The domain is a set of students at a school. x ≤ y if x has the same birthday as y. There are at least two students in the school with the same birthday.

- Yes, it is a partial order.
- No, it is not a partial order because it is not anti-symmetric.
- No, it is not a partial order because it is not transitive.

Consider a finite set A. There is a natural partial order on the P(A), the power set of A: for any two subsets X and Y of A, X ≤ Y if X ⊆ Y. The following animation illustrates why the subset relation is a partial order:

PARTICIPATION ACTIVITY | 6.9.4: Partial order defined on a power set.

Animation content:

undefined

Animation captions:

1. In an arrow diagram for the partial order on P({1,2,3}), start with the arrows between subsets A and B such that A ⊆ B and B has one more element than A.
2. The relation is reflexive because A ⊆ A, so self loops are added.
3. The relation is transitive because A ⊆ B and B ⊆ C implies that A ⊆ C, so edges are added to reflect that the relation is transitive.

A ***Hasse diagram***, named after the 20th century German mathematician Helmut Hasse, is a useful way to depict a partial order on a finite set. Each element is represented by a labeled point. The idea is to show precedence relationships by placing elements that are greater than others towards the top of the diagram. Some precedence relationships are depicted with lines between elements, but only enough to make the relationship between elements clear. Otherwise, the goal is to not clutter the diagram with unnecessary edges. The rules for placement and for connecting segments are given below.

For any x and y such that x ≠ y:

- If x ≤ y, then make x appear lower in the diagram than y.
- If x ≤ y and there is no z such that x ≤ z and z ≤ y, then draw a segment connecting x and y.

Since any partial order is transitive, it is understood that if there is a path from x to y along segments that all travel upwards, then x ≤ y. Similarly, the self-loops at each element are not drawn, but it is understood that a partial order is reflexive. The below animation illustrates Hasse diagrams for two partial orders.

PARTICIPATION ACTIVITY | 6.9.5: Hasse diagram for a partially ordered set.

Animation content:

undefined

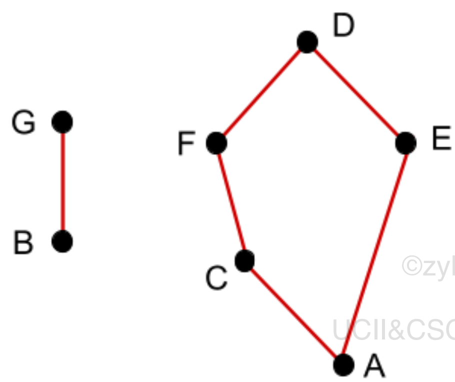
Animation captions:

1. The Hasse diagram for P({1,2,3}) has an undirected edge between subsets A and B such that A ⊆ B and B has one more element than A.
2. The Hasse diagram for a total order is just a single line of edges.

If a partial order is drawn according to the above rules of a Hasse diagram, then x ≤ y if and only if there is a path from x to y along segments that all travel upwards. For example, in the Hasse diagram shown in the figure below, there is a path from A to D that only moves along segments in the upward direction (A to C to F to D). Therefore, A ≤ D. By contrast, any path from C to E must go up and then down (C to F to D to E)) or down and then up (C to A to E)). Therefore, C ≰ E. Also, since there is no path along segments from C to G, then C ≰ G

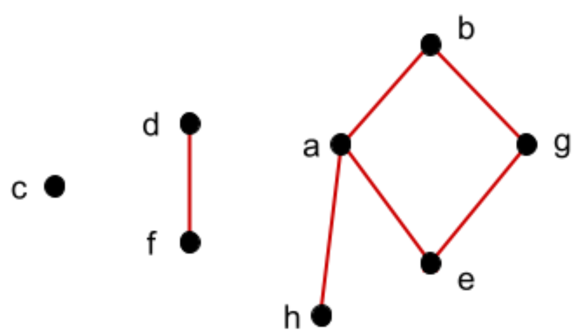
In general, if two elements are incomparable (neither x ≤ y nor y ≤ x), then either they are not connected at all by a path of line segments or the only paths between x and y require a change in direction from up to down or from down to up.

Figure 6.9.2: A Hasse diagram for a partial order on the set {A, B, C, D, E, F, G}.



PARTICIPATION ACTIVITY | 6.9.6: Interpreting Hasse diagrams.

The Hasse diagram below depicts a partial order on the set {a, b, c, d, e, f, g, h}



1) Are f and d comparable?

- Yes
- No

2) Are c and a comparable?

- Yes
- No

3) Are h and e comparable?

- Yes
- No

4) Are g and a comparable?

- Yes
- No

5) Are b and e comparable?

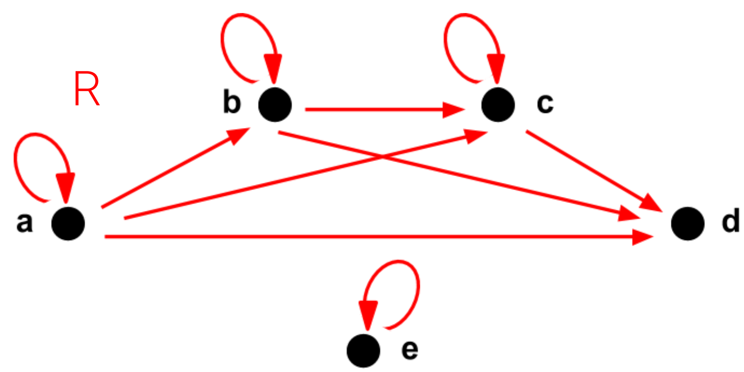
- Yes
- No

CHALLENGE ACTIVITY | 6.9.1: Partial orders.

460784.3106054.qx3zqy7

Start

The figure below shows an arrow diagram of a relation R.



Is the relation reflexive?

Is the relation transitive?

Is the relation anti-symmetric?

Is the relation a partial order?

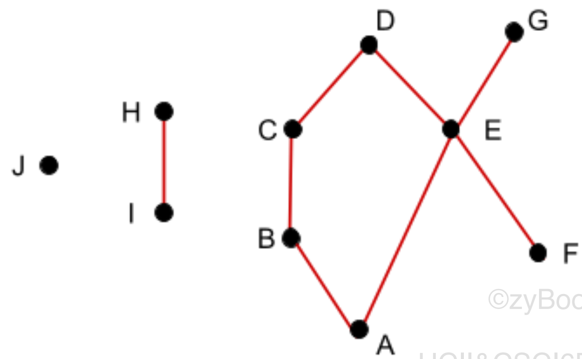
1 2 3

Check Next

Additional exercises

EXERCISE | 6.9.1: Interpreting Hasse diagrams.

The drawing below shows a Hasse diagram for a partial order on the set {A, B, C, D, E, F, G, H, I, J}



(a) What are the minimal elements of the partial order?

(b) What are the maximal elements of the partial order?

(c) Which of the following pairs are comparable?
(A, D), (J, F), (B, E), (G, F), (D, B), (C, F), (H, I), (C, E)

EXERCISE 6.9.2: Drawing Hasse diagrams.

Each relation given below is a partial order. Draw the Hasse diagram for the partial order.

(a) The domain is the power set of A (P(A)), where A = {x, y}. For any X, Y ⊆ A, X ≼ Y if X ⊆ Y.

(b) The domain is {3, 5, 6, 7, 10, 14, 20, 30, 60}. x ≼ y if x evenly divides y.

(c) The domain is {1, 5, 8, 11, 13}. x ≼ y if x ≤ y.

(d) The domain is {a, b, c, d, e, f}. The relation is the set:
{ (b, e), (b, d), (c, a), (c, f), (a, f), (a, a), (b, b), (c, c), (d, d), (e, e), (f, f) }

(e) The domain is {A, B, C, D, E, F, G, H}. The relation is the set:
{ (C, B), (D, C), (C, F), (G, H), (F, B), (A, B), (D, F), (D, B), (D, A),
(A, A), (B, B), (C, C), (D, D), (E, E), (F, F), (G, G), (H, H) }

EXERCISE 6.9.3: Partial orders on complex relations.

Suppose R_1 is a relation on domain X_1 and R_2 is a relation on domain X_2

Define a new relation R whose domain is $X_1 \times X_2$:

$(x_1, x_2)R(x'_1, x'_2)$ if $x_1 R_1 x'_1$ and $x_2 R_2 x'_2$.

(a) Show that if R_1 and R_2 are both partial orders, then R is also a partial order.

(b) Now change the relation R so that $(x_1, x_2)R(x'_1, x'_2)$ if $x_1 R_1 x'_1$ OR $x_2 R_2 x'_2$.
Give an example in which R_1 and R_2 are partial orders but R is not a partial order.

6.10 N-ary relations and relational databases

The definition of a binary relation (a relation between two sets) can be generalized to relations on more than two sets:

> A relation on sets $A_1, A_2, \ldots, A_n$ is a subset of $A_1 \times A_2 \times \ldots \times A_n$.
> Each element of the relation is an n-tuple in which the i^{th} entry in the n-tuple is from A_i.
> A relation on n sets is called an ***n-ary relation***.

For example, a relation could consist of all 4-tuples $(w, x, y, z) \in \mathbf{R}^4$ such that wx = yz. Then (3, 12, 4, 9) would be in the relation because $3 \cdot 12 = 4 \cdot 9$. However (3, 8, 5, 6) would not be in the relation because $3 \cdot 8 \neq 5 \cdot 6$.

Relations can also be defined on sets that are a combination of numerical and non-numerical data. Consider for example a set of 7-tuples, each of which gives information about a flight on a particular airline. The 7-tuples would have the form:

> (flight number, departure airport, arrival airport, date, departure time, arrival time, type of aircraft)

A ***database*** is a large collection of data records that is searched and manipulated by a computer. The ***relational database model*** stores data records as relations. The type of data stored in each entry of the n-tuple is called an ***attribute***. In the airline example, "flight number" is the first attribute.

A large airline would store many thousands of relations for past, current, and future flights. A ***query*** to a database is a request for a particular set of data. For example, employees of the airline would need to access the database to answer queries like:

- How many flights were there in July 2013 on a Boeing 767-300?
- What flights will be leaving LAX on July 4, 2013?
- What is the earliest flight from LAX to DTW on June 19, 2013?

A relation (or relational database) can be represented in table format with a column for each attribute and a row for each n-tuple in the relation. The name of the attribute is the header for that column. The usual parentheses and commas are omitted since the value for each attribute is clear from the layout of the table:

Table 6.10.1: A relational database shown in table format.

Flight Number	Departure Airport	Arrival Airport	Date	Departure Time	Arrival Time	Aircraft
1806	LAX	DTW	06/19/2013	9:15AM	4:44PM	Boeing 767-300
18	LAX	DTW	06/19/2013	1:35PM	8:54PM	Boeing 757
1719	DTW	LAX	06/22/2013	12:05PM	1:51PM	Boeing 737-800
2262	LAX	JFK	07/03/2013	1:20PM	9:55PM	Boeing 767-300
2262	LAX	JFK	07/04/2013	1:20PM	9:55PM	Boeing 767-300
2171	JFK	LAX	07/07/2013	7:00AM	9:57AM	Boeing 767-300
226	LAX	SFO	07/07/2013	11:30AM	12:52PM	CRJ 900

The questions above can be answered with respect to the small set of data represented in the table. For example, there were three flights on a Boeing 767-300 in July 2013. Flight 2262 is the only flight leaving LAX on July 4, 2013 and the earliest flight from LAX to DTW on June 19, 2013 leaves at 9:15AM.

A ***key*** is an attribute or set of attributes that uniquely identifies each n-tuple in the database. In the airline example, flight number alone would not be a key since there are two flights numbered 2262 that leave on different days. In the table above, flight number and date would work as a key since no two entries have the same flight number and date. Typically, though, airlines use the combination of flight number, date, and departure city to uniquely identify a flight. A company that maintains a database of its employees would likely include attributes such as social security number, name, start date, and salary. The social security number alone would suffice as a key since no two people have the same social security number.

PARTICIPATION ACTIVITY 6.10.1: Relational databases: Basic definitions.

Use the table above with the database of airline flights to answer the following questions:

1) In the database defined in the table, would departure airport and arrival

airport suffice as a key?

- Yes
- No

2) In the database defined in the table, would departure time and date suffice as a key?

- Yes
- No

3) What is the last attribute in the database?

- 226
- Aircraft
- CRJ 9000

This section describes two common operations that can be performed on relational databases that are useful in answering queries.

The selection operation

The ***selection operation*** chooses n-tuples from a relational database that satisfy particular conditions on their attributes. The animation below gives an example of the selection operation applied to a small database:

PARTICIPATION ACTIVITY 6.10.2: Selection operation example.

Animation captions:

1. A Select[Complete = No] is executed by crossing out rows that have a "Yes" in the Complete column and selecting rows with a "No".
2. The crossed out rows are deleted.
3. A Select[Complete = No $\wedge$ Date < 6/21/2013] is executed by finding the rows with a "No" in the complete column and whose Date is before 6/21/2013.
4. Rows with a "Yes" in the complete column or with a date that is on or after 6/21/2013 are deleted.

Applying the following operation:

Select[Departure Airport = LAX]

to the table with airline flights would return all the flights ever to have left or scheduled to leave LAX:

Table 6.10.2: Select[Departure Airport = LAX] applied to the airline database.

Flight Number	Departure Airport	Arrival Airport	Date	Departure Time	Arrival Time	Aircraft
1806	LAX	DTW	06/19/2013	9:15AM	4:44PM	Boeing 767-300
18	LAX	DTW	06/19/2013	1:35PM	8:54PM	Boeing 757
2262	LAX	JFK	07/03/2013	1:20PM	9:55PM	Boeing 767-300
2262	LAX	JFK	07/04/2013	1:20PM	9:55PM	Boeing 767-300
226	LAX	SFO	07/07/2013	11:30AM	12:52PM	CRJ 900

The operation

Select[Departure Airport = LAX ∧ date ≤ 07/03/2013]

would return all the flights that left LAX on or before July 3, 2013:

Table 6.10.3: Select[Departure Airport = LAX ∧ date ≤ 07/03/2013] applied to the airline database.

Flight Number	Departure Airport	Arrival Airport	Date	Departure Time	Arrival Time	Aircraft
1806	LAX	DTW	06/19/2013	9:15AM	4:44PM	Boeing 767-300

18	LAX	DTW	06/19/2013	1:35PM	8:54PM	Boeing 757
2262	LAX	JFK	07/03/2013	1:20PM	9:55PM	Boeing 767-300

The projection operation

The ***projection*** operation takes a subset of the attributes and deletes all the other attributes in each of the n-tuples. An instance of the projection operation takes the form: Project[list of attributes]. If the number of attributes in the list is m (with m ≤ n), then the result is a set of m-tuples. Duplicates are then removed from the resulting set. The animation below shows the projection operation applied to a small database:

PARTICIPATION ACTIVITY 6.10.3: Projection operation example.

Animation captions:

1. A Project[Date,City] command is executed by deleting all columns except for "Order Date" and "Client City".
2. The question "Which cities have incomplete orders?", can be answered by first executing a Select[Complete=No] command to find the incomplete orders.
3. The rows that are not selected are deleted.
4. Followed by: Project[City], to remove all the columns except the "Client City" column.
5. Finally, duplicate results are merged.

In the airline example, Project[Departure Airport, Arrival Airport] would yield:

Table 6.10.4: Project[Departure Airport, Arrival Airport] applied to airline database.

Departure Airport	Arrival Airport
LAX	DTW
DTW	LAX
LAX	JFK

JFK	LAX
LAX	SFO

The selection and projection operations can be combined to answer queries. Consider the query:

From which airports does the Boeing 767-300 depart?

To answer this question, one first applies the operation Select[aircraft = Boeing 767-300] to yield only the flights that use a Boeing 767-300. The next step is to project on to departure airport to get the departure airports for the flights that use a Boeing 767-300.

PARTICIPATION ACTIVITY 6.10.4: Relational databases: Selection and projection.

Consider the following database of students in a school:

Name	Student ID	Year	Credits
Brad Smith	385475	Freshman	16
Samuel Wu	360442	Senior	146
Sonya Nguyen	833288	Sophomore	36
Naresh Gupta	334295	Sophomore	42

1) The operation Select[Credits ≥ 40] returns the 4-tuples corresponding to which two students?

- Sonya and Naresh
- Samuel and Naresh
- Samuel and Brad

2) The operation Project[Year, Credits] returns a set of which kind of object?

- Triples
- 4-tuples
- Pairs

3) A query first performs a Select[Credits ≤ 50] and then does a Project[Year] on

the results of the first operation. What is returned by the query?

- (Freshman), (Sophomore)
- (Sophomore)
- (Freshman, 16), (Sophomore, 36), (Sophomore, 42)

CHALLENGE ACTIVITY 6.10.1: Relational databases.

460784.3106054.qx3zqy7

Start

A key for this database is {Departure Time}. Enter a different key for the database.

Airline	Flight Number	Gate	Destination	Departure Time
FlyRight	322	22	Detroit	08:10
FlyRight	222	13	Denver	08:27
FlyRight	122	33	Denver	08:50
JetGreen	122	13	Anchorage	09:30
JetGreen	221	22	Detroit	09:31
JetGreen	323	13	Honolulu	09:52
JetGreen	322	34	Honolulu	09:57

Key: { Ex: Flight Number }

1 2 3 4 5 6

Check Next

CHALLENGE ACTIVITY 6.10.2: Relational databases select and project.

460784.3106054.qx3zqy7

Start

What operation should be performed if you wanted to know what airlines fly to Der

Airline	Flight Number	Gate	Destination	Departure Time
JetGreen	322	34	Denver	08:19
JetGreen	323	13	Detroit	08:33
JetGreen	322	22	Detroit	08:53
JetGreen	122	22	Honolulu	08:57
FlyRight	199	33	Anchorage	08:59
FlyRight	199	34	Denver	09:23
JetGreen	323	33	Anchorage	09:48

Select[Ex: Gate="22"] Project[Ex: Destination]

1 2 3 4

Check Next

Additional exercises

EXERCISE | 6.10.1: n-ary relations: Numerical examples.

Express each relation as a set of n-tuples in roster format.

(a) R = { (a, b, c, d): a, b, c, d are positive integers and a + b + c + d = 6 }

(b) S = { (a, b, c): a, b, c are positive integers and a < b < c < 5 }

EXERCISE | 6.10.2: n-ary relations, cont.

Consider the following two relations below:

- R = { (a, b, c): a, b, c are positive integers and a < b < c < 6 and a + b + c < 9 }
- S = { (a, b, c): a, b, c are positive integers such that $a \neq b$, $b \neq c$, $a \neq c$, and a + b + c < 8 }

(a) Express R as a set of triples in roster format.

(b) Express S as a set of triples in roster format.

(c) Since R and S are both subsets of Z^3, $R \cap S$ is well defined. Express $R \cap S$ in roster format.

(d) Since R and S are both subsets of Z^3, $R \cup S$ is well defined. Express $R \cup S$ in roster format.

EXERCISE 6.10.3: Select and Project operations on a relational database: Airline flights.

The table below shows a small database whose records correspond to flights departing from an airport.

Airline	Flight Number	Gate	Destination	Departure Time
FlyRight	122	34	Detroit	08:10
JetGreen	221	22	Denver	08:17
JetGreen	122	33	Anchorage	08:22
JetGreen	323	34	Honolulu	08:30
FlyRight	199	13	Detroit	08:47
JetGreen	222	22	Denver	09:10
FlyRight	322	34	Detroit	09:44

(a) Are the attributes "Airline" and "Destination" a key for this database?

(b) Give two examples of sets of attributes that are keys for this database.

(c) Show the results of Select[Airline = "JetGreen" and Destination="Denver"].

(d) Show the results of Project[Airline, Gate].

(e) Show the results of Select[Destination = "Denver"] followed by Project[Gate].

(f) What operations should be performed if you want to know which airlines use gate 22?

(g) What operations should be performed if you want to know whether there are any flights to Detroit departing before 09:10? A flight is identified by the airline and flight number.

EXERCISE 6.10.4: Select and Project operations on a relational database: University courses.

The table below shows a small database whose records correspond to courses offered in a university.

Course Number	Course Title	Instructor	Quarter
CS 111	Digital Image Processing	Gupta	Spring
CS 112	Computer Graphics	Gupta	Spring
CS 117	Project in Computer Vision	Hawkins	Fall
CS 116	Intro to Computer Vision	Hawkins	Winter
CS 122A	Intro to Database Management	Spencer	Spring
CS 122A	Intro to Database Management	Wang	Winter
CS 122C	Principles of Database Management	Spencer	Winter

(a) Give a key for this database.

(b) Show the results of the operation Select[Course Number = "CS 122A" ∨ Course Number = "CS 122C"]

(c) Show the results of the operation Project[Instructor]

(d) Show the results of the operation Select[Quarter = "Spring"] followed by Project[Course Number]. Express in English what question this combination of operations is asking.

(e) Which operations should be performed if you want to know in which quarters CS 122A is being offered?

(f) Which operations should be performed if you want to know which courses are being taught by Prof. Gupta? A course is identified by its course title.

EXERCISE 6.10.5: Select operations on relational databases.

Consider a relational database R. C_1 and C_2 are two conditions that could be used in a Select operation on R.

(a) Prove that Select[C_1] followed by Select[C_2] yields the same results as Select[C_2] followed by Select[C_1]

(b) How could the two operations performed in sequence (Select[C_1] followed by Select[C_2]) be replaced by a single Select operation and achieve the same results?