Sets and Functions
(Rosen, Sections 2.1,2.2, 2.3)

TOPICS

• Discrete math
• Set Definition
• Set Operations
• Tuples

Discrete Math at CSU (Rosen book)

- CS 160 or CS122
  - Sets and Functions
  - Propositions and Predicates
  - Inference Rules
  - Proof Techniques
  - Program Verification
- CS 161
  - Counting
  - Induction proofs
  - Recursion
- CS 200
  - Algorithms
  - Relations
  - Graphs

Why Study Discrete Math?

- Digital computers are based on discrete units of data (bits).
- Therefore, both a computer's
  - structure (circuits) and
  - operations (execution of algorithms)

  can be described by discrete math
- A generally useful tool for rational thought! Prove your arguments.

What is 'discrete'?

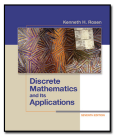
- Consisting of distinct or unconnected elements, not continuous (calculus)
- Helps us in Computer Science:
  - What is the probability of winning the lottery?
  - How many valid Internet address are there?
  - How can we identify spam e-mail messages?
  - How many ways are there to choose a valid password on our computer system?
  - How many steps are need to sort a list using a given method?
  - How can we prove our algorithm is more efficient than another?

Uses for Discrete Math in Computer Science

- Advanced algorithms & data structures
- Programming language compilers & interpreters.
- Computer networks
- Operating systems
- Computer architecture
- Database management systems
- Cryptography
- Error correction codes
- Graphics & animation algorithms, game engines, *etc.*…
- *i.e.*, the whole field!

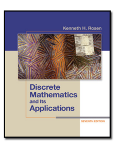
What is a set?

- *An unordered collection of unique objects*
  - {1, 2, 3} = {3, 2, 1} since sets are unordered.
  - {a, b, c} = {b, c, a} = {c, b, a} = {c, a, b} = {a, c, b}
  - {2}
  - {on, off}
  - { }
  - {1, 2, 3} = {1, 1, 2, 3} since elements in a set are unique



What is a set?

- Objects are called *elements* or *members* of the set
- Notation $\in$
  - $a \in B$ means "a is an element of set B."
  - Lower case letters for elements in the set
  - Upper case letters for sets
  - If $A = \{1, 2, 3, 4, 5\}$ and $x \in A$, what are the possible values of x?



What is a set?

- Infinite Sets *(without end, unending)*
  - N = {0, 1, 2, 3, …} is the Set of natural numbers
  - Z = {…, -2, -1, 0, 1, 2, …} is the Set of integers
  - Z+ = {1, 2, 3, …} is the Set of positive integers
- Finite Sets *(limited number of elements)*
  - V = {a, e, i, o, u} is the Set of vowels
  - O = {1, 3, 5, 7, 9} is the Set of odd #'s < 10
  - F = {a, 2, Fred, New Jersey}
  - Boolean data type used frequently in programming
    - B = {0,1}
    - B = {false, true}
  - Seasons = {spring, summer, fall, winter}
  - ClassLevel = {Freshman, Sophomore, Junior, Senior}

What is a set?

- Infinite vs. finite
  - If finite, then the number of elements is called the *cardinality*, denoted |S|
    - V = {a, e, i, o, u} |V| = 5
    - F = {1, 2, 3} |F| = 3
    - B = {0,1} |B| = 2
    - S = {spring, summer, fall, winter} |S| = 4
    - A = {a, a, a} |A| = 1

Example sets

- Alphabet
- All characters
- Booleans: true, false
- Numbers:
  - $\mathbf{N}=\{0,1,2,3\ldots\}$ Natural numbers
  - $\mathbf{Z}=\{\ldots,-2,-1,0,1,2,\ldots\}$ Integers
  - Q= $\{p/q \mid p\in Z, q\in Z, q\neq 0\}$ Rationals
  - **R**, Real Numbers
- Note that:
  - **Q** and **R** are not the same. **Q** is a *subset* of **R**.
  - **N** is a subset of **Z**.

Example: Set of Bit Strings

- A bit string is a sequence of zero or more bits.
- A bit string's length is the number of bits in the string.
- A set of all bit strings s of length 3 is
  - S= {000, 001, 010, 011, 100, 101, 110, 111}

What is a set?

- Defining a set:
  - Option 1: List the members
  - Option 2; Use a set builder that defines set of x that hold a certain characteristic
  - Notation: $\{x \in S \mid \text{characteristic of } x\}$
  - Examples:
    - $A = \{ x \in Z^{+} \mid x \text{ is prime} \}$ – set of all prime positive integers
    - $O = \{ x \in N \mid x \text{ is odd and } x < 10000 \}$ – set of odd natural numbers less than 10000

Equality

- Two sets are *equal* if and only if (iff) they have the same elements.
- We write A=B when for all elements x, x is a member of the set A iff x is also a member of B.
  - Notation: $\forall x\{x \in A \leftrightarrow x \in B\}$
  - For all values of x, x is an element of A if and only if x is an element of B

Set Operations

- Operations that take as input sets and have as output sets
- Operation1: *Union*
  - The union of the sets A and B is the set that contains those elements that are either in A or in B, or in both.
  - Notation: $A \cup B$
  - Example: union of {1,2,3} and {1,3,5} is?

Operation 2: Intersection

- The intersection of sets A and B is the set containing those elements in both A and B.
- Notation: $A \cap B$
- Example: {1,2,3} intersection {1,3,5} is?
- The sets are disjoint if their intersection produces the empty set.

Operation3: Difference

- The difference of A and B is the set containing those elements that are in A but not in B.
- Notation: $A - B$
- Aka the complement of B with respect to A
- Example: {1,2,3} difference {1,3,5} is?
- Can you define Difference using union, complement and intersection?

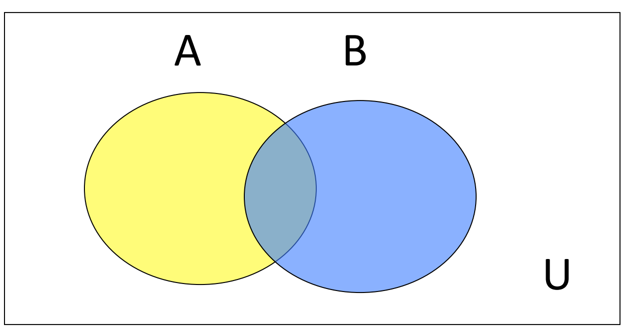
Operation3: Complement

- The complement of set A is the complement of A with respect to U, the universal set.
- Notation: $\overline{A}$
- Example: If N is the universal set, what is the complement of {1,3,5}?

  Answer: {0, 2, 4, 6, 7, 8, …}

Venn Diagram

- Graphical representation of set relations:



Identities

Identity	$A \cup \varnothing = A, A \cap U = A$
Commutative	$A \cup B = B \cup A, A \cap B = B \cap A$
Associative	$A \cup (B \cup C) = (A \cup B) \cup C, A \cap (B \cap C) = (A \cap B) \cap C$
Distributative	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C), A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
Complement	$A \cup \overline{A} = U, A \cap \overline{A} = \varnothing$

Subset

- The set A is said to be a subset of B iff for all elements x of A, x is also an element of B.
  But not necessarily the reverse…
- Notation: $A \subseteq B$ $\forall x\{x \in A \rightarrow x \in B\}$
  - Unidirectional implication
- $\{1,2,3\} \subseteq \{1,2,3\}$
- $\{1,2,3\} \subseteq \{1,2,3,4,5\}$
- What is the cardinality between sets if $A \subseteq B$?

Answer: |A| <= |B|

Subset

- Subset is when a set is contained in another set. Notation: $\subseteq$
- Proper subset is when A is a subset of B, but B is not a subset of A. Notation: $\subset$
  - $\forall x\, ((x \in A) \rightarrow (x \in B)) \wedge \exists x\, ((x \in B) \wedge (x \notin A))$
  - All values x in set A also exist in set B
  - … but there is at least 1 value x in B that is not in A
  - A = {1,2,3}, B = {1,2,3,4,5}

  $A \subset B$, means that $|A| < |B|$.

Empty Set

- Empty set has no elements and therefore is the subset of all sets. { } Alternate Notation: $\varnothing$
- Is $\varnothing \subseteq \{1,2,3\}$? - Yes!
- The cardinality of $\varnothing$ is zero: $|\varnothing| = 0$.
- Consider the set containing the empty set: $\{\varnothing\}$.
- Yes, this is indeed a set: $\varnothing \in \{\varnothing\}$ and $\varnothing \subseteq \{\varnothing\}$.

Set Theory - Definitions and notation

- Quiz time:
  - A = { x∈N | x ≤ 2000 }   What is |A| = 2001 ?
  - B = { x∈N | x ≥ 2000 }   What is |B| = Infinite!
  - Is {x} ⊆ {x}? Yes
  - Is {x} ∈ {x,{x}}? Yes
  - Is {x} ⊆ {x,{x}}? Yes
  - Is {x} ∈ {x}? No

Powerset

- The <u>powerset</u> of a set is the set containing *all* the subsets of that set.
- Notation: ***P***(A) is the powerset of set A.
- Fact: | ***P***(A) | = $2^{|A|}$.
- If A = { x, y }, then ***P***(A) = {∅, {x}, {y}, {x,y} }
- If S = {a, b, c}, what is ***P***(S)?

Powerset example

- **Number of elements in powerset = 2^n where n = # elements in set**

- ***S*** is the set **{a, b, c}**, what are all the subsets of ***S***?
  - { } – the empty set
  - {a}, {b}, {c} – one element sets
  - {a, b}, {a, c}, {b, c} – two element sets
  - {a, b, c} – the original set

and hence the power set of *S* has 2^3 = 8 elements:

{{}, {a}, {b}, {c}, {a, b}, {b, c}, {c, a}, {a, b, c}}

Why sets?

- Programming - Recall a *class*… it is the set of all its possible objects.
- We can restrict the *type* of an object, which is the set of values it can hold.
  - Example: Data Types

int	set of integers (finite)
char	set of characters (finite)

  - Is $\mathbb{N}$ the same as the set of integers in a computer?

Order Matters

- What if order matters?
  - Sets disregard ordering of elements
  - If order is important, we use *tuples*
  - If order matters, then are duplicates important too?

Tuples

- Order matters
- Duplicates matter
- Represented with parens ( )
- Examples
  - (1, 2, 3) ≠ (3, 2, 1) ≠ (1, 1, 1, 2, 3, 3)

    $(a_1, a_2, \ldots, a_n)$

Tuples

- The *ordered n-tuple* $(a_1, a_2, \ldots, a_n)$ is the ordered collection that has a_1 as its first element a_2 as its second element … and a_n as its nth element.
- An *ordered pair* is a *2-tuple*.
- Two ordered pairs (a,b) and (c,d) are equal iff a=c and b=d *(e.g. NOT if a=d and b=c)*.
- A 3-tuple is a *triple*; a 5-tuple is a *quintuple*.

Tuples

- In programming?
  - Let's say you're working with three integer values, first is the office room # of the employee, another is the # years they've worked for the company, and the last is their ID number.
    - Given the following set {320, 13, 4392}, how many years has the employee worked for the company?
    - What if the set was {320, 13, 4392}?
      Doesn't {320, 13, 4392} = {320, 4392, 13} ?
    - Given the 3-tuple (320, 13, 4392) can we identify the number of years the employee worked?

Why?

- Because ordered n-tuples are found as lists of arguments to functions/methods in computer programming.
- Create a mouse in a position (2, 3) in a maze: **`new Mouse(2,3)`**
- Can we reverse the order of the parameters?
- From Java, **`Math.min(1,2)`**

Cartesian Product of Two Sets

- Let A and B be sets. The Cartesian Product of A and B is the set of all ordered pairs (a,b), where $b \in B$ and $a \in A$
- Cartesian Product is denoted A x B.
- Example: A={1,2} and B={a,b,c}. What is A x B and B x A?

Cartesian Product

- A = {a, b}
- B = {1, 2, 3}
- A X B = {(a, 1), (a, 2), (a, 3), (b, 1), (b, 2), (b, 3)}
- B X A = {(1, a), (1, b), (2, a), (2, b), (3, a), (3, b)}

Functions in CS

- Function = mappings or transformations
- Examples

  f(x) = x

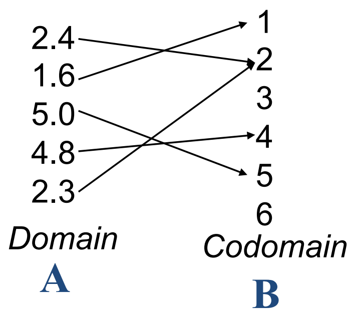
  f(x) = x + 1

  f(x) = 2x

  f(x) = x^2

Function Definitions

- A function **f** from sets **A** to **B** assigns exactly one element of **B** to each element of **A**.
- Example: the **floor** function



What's the difference between codomain and range?

Range: {1,2,4,5}

Range contains the codomain values that A maps to

Function Definitions

- In Programming
  - Function header definition example

```
int floor( float real)
{

}
```

  - Domain = R
  - Codomain = Z

Other Functions

- The **identity** function, $\mathbf{f_{ID}}$, on $\mathbf{A}$ is the function where: $f_{ID}(x) = x$ for all **x** in $\mathbf{A}$.
  - A = {a,b,c} and f(a) = a, f(b) = b, f(c) = c
- **Successor function,** $f_{succ}(x) = x+1$, on $\mathbf{Z}$
  - f(1) = 2
  - f(-17) = -16
  - f(a) Does NOT map to b

Only works on set **Z**

- **Predecessor function,** $f_{pred}(x) = x-1$, on $\mathbf{Z}$
  - f(1) = 0
  - f(-17) = -18

Other Functions

- $f_{NEG}(x) = -x$, also on $\mathbf{R}$ (or $\mathbf{Z}$), maps a value into the negative of itself.
- $f_{SQ}(x) = x^2$, maps a value, x, into its square, x^2.
- The **ceiling** function: ***ceil***(2.4) = 3.

Functions in CS

- What are ceiling and floor useful for?
  - Data stored on disk are represented as a string of bytes. Each byte = 8 bits. How many bytes are required to encode 100 bits of data?

Need smallest integer that is at least as large as 100/8

100/8 = 12.5
But we don't work with ½ a byte.
So we need 13 bytes

What is NOT a function?

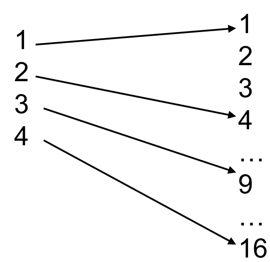
- Consider $f_{SQRT}(x)$ from $\mathbf{Z}$ to $\mathbf{R}$.
- This does **not** meet the given definition of a function, because $f_{SQRT}(16) = \pm 4$.
- In other words, $f_{SQRT}(x)$ assigns exactly *one* element of $\mathbf{Z}$ to **two** elements of $\mathbf{R}$.

No Way!
Say it ain't so!!

Note that the convention is that $\sqrt{x}$ is always the positive value.
$f_{SQRT}(x) = \pm\sqrt{x}$

1 to 1 Functions

- A function **f** is said to be *one-to-one* or *injective* if and only if **f(a) = f(b)** implies that **a = b** for all **a** and **b** in the domain of **f**.
- Example: the **square** function from $\mathbf{Z}^+$ to $\mathbf{Z}^+$



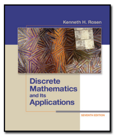
1 to 1 Functions, cont.

- Is **square** from $\mathbf{Z}$ to $\mathbf{Z}$ an example?
  - **NO!**
  - Because $f_{SQ}(-2) = 4 = f_{SQ}(+2)$!
- Is **floor** an example?

  INCONCEIVABLE!!
- Is **identity** an example?

  Unique at last!!

How *dare* they have the same codomain!



Increasing Functions

- A function **f** whose domain and co-domain are subsets of the set of real numbers is called *increasing* if **f(x) <= f(y)** and *strictly increasing* if **f(x) < f(y)**, whenever
  - **x < y** and
  - **x** and **y** are in the domain of **f**.
- Is **floor** an example?

1.5 < 1.7 and floor(1.5) = 1 = floor(1.7)
1.2 < 2.2 and floor(1.2) = 1 < 2 = floor(2.2)

So YES floor is an increasing function

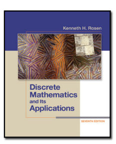
BUT it is **NOT** STRICTLY increasing

- Is **square** an example?

When mapping Z to Z or R to R:
square(-2) = 4 > 1 = square (1) yet -2 < 1

NO square is NOT an increasing function UNLESS….

Domain is restricted to positive #'s



How is Increasing Useful?

- Most programs run longer with larger or more complex inputs.
- Consider looking up a telephone number in the paper directory…

Cartesian Products and Functions

- A function with multiple arguments maps a Cartesian product of inputs to a codomain.
- Example:
  - **`Math.min`** maps **Z x Z** to **Z**

    ```
    int minVal = Math.min( 23, 99 );
    ```
  - **`Math.abs`** maps **Q** to **Q+**

    ```
    int absVal = Math.abs( -23 );
    ```

Find the minimum value between two integers

Find the absolute value of a number

Quiz Check

- Is the following an increasing function?

  Z → Z   f(x) = x + 5

  Z → Z   f(x) = 3x -1