Part 1 - Sets

- Standard sets ($\mathbb{Z}$, $\mathbb{Q}$, ...)
- Set Builder Notation
- Venn Diagrams
- Set Operators
- Set algebra
- Tuples

Standard Set

$\mathbb{Z}$ - Integers (...-2, -1, 0, 1, 2...)

N - Natural Numbers (1, 2, 3, 4, ...)

Q - Quotient/Rational ($\frac{a}{b}$ & a, b are $\mathbb{Z}$ and $b \neq 0$ zero)

R - Real Numbers (all Non-Imaginary 1, -2.5, 3.14159)

U - Universal Set (All values of interest) *Depends on context*

Set Builder Notation

Set Builder Notation consist of a variable name a pipe symbol ("|"), and a true/false expression

Example: $A = \{x \mid x \in \mathbb{Z} \text{ and } x \text{ is even}\}$

Read as: All x Where x is in $\mathbb{Z}$ and x is even

To emulate this in set builder: $\{2, 4, 6, 8, 10, \ldots\}$

You can 1) $A = \{x \mid x \in N \text{ and } x \text{ is even}\}$

OR

2) $A = \{2x \mid x \in N\}$

When evaluating $\{2x \mid x \in N\}$ do the following

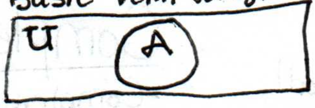
Step1| Identify which variables make the right-hand-side true.

Step2| Plug them into the left-hand-side. These are the values in the set.

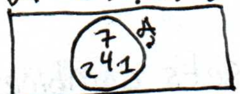
Venn Diagrams

- Each set is represented by a circle
  - Overlaps between each set can show logical relations with set numbers.

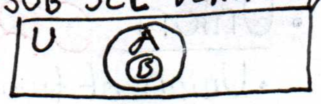
Basic Venn Diagram



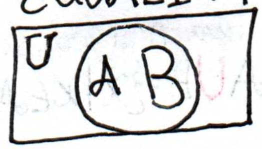
$A = \{2, 7, 1, 4\}$



SUB-SET Venn diagram



EQUALITY



Tuples - order is Important

- To denote sets we use curly brackets
  - For example prime numbers 1 to 10 is $\{2, 3, 5, 7\}$ ← 4-tuples cardinality of four
    - Order does not matter, so $\{2,3,5,7\} == \{2,5,7,3\}$
- In many cases order is important.
- Called n-tuples where "n" is the number of elements/members.
  - 2-tuples are also called ordered pairs
- To denote a tuple we use (parenthesis), ⟨angled brackets⟩, or [square brackets]

Back

- Set Operators
- Set Algebra

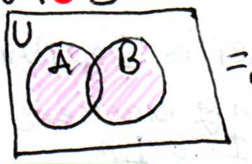
Set Operators

Union OR

- Union of two sets combines all members of each set into a new one
  - The result is two merged sets

Set Notation: $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$

$A \cup B$



$= A \cup B = \{x \mid x \in A \text{ or } x \in B\}$

Intersection AND

- Intersection contains only those elements found in both sets
  - The result is where the two sets overlap

Set Notation:

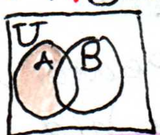
$A \cap B = \{x \mid x \in A \text{ and } x \in B\}$

Difference

- Difference excludes all items found in on set from another (AKA: Relative Complement)

Set Notation:

$A \setminus B = \{x \mid x \in A \text{ and } x \notin B\}$

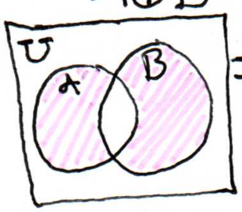


Symmetric Difference

- Symmetric Difference is all the items that are in either of two sets, but NOT both

Set Notation:

$A \oplus B = (A \cup B) \setminus (A \cap B)$



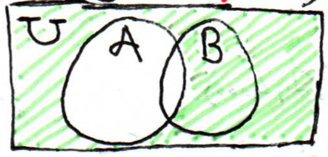
$= (A \setminus B) \cup (B \setminus A)$

Complement NOT/Negation

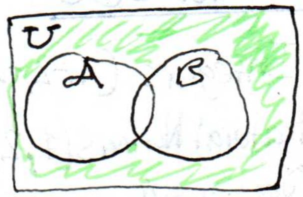
- Complement of a set A, is all elements in the universe, NOT in A

Set Notation:

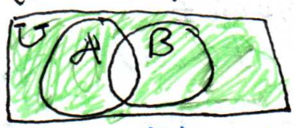
$A' = \{x \mid x \notin A\}$



$(A \cup B)'$



$(A \cap B)'$



Set Algebra

Communitive Law

- Both $\cap$ and $\cup$ are communitive
  - This means the left-hand & right hand can be switched

Example:

$A \cap B \equiv B \cap A$

$A \cup B \equiv B \cup A$

Idempotent Law

When a set is combined with itself

$A \cap A \equiv A$

$A \cup A \equiv A$

Involotion Law

- Double Negation

$(A')' \equiv A$

Complement Law

Set is used with its complement will result in either universe or empty

$A \cap A' \equiv \emptyset$

$A \cup A' \equiv U$

Associative Law

Example: $(a+b)+c = a+(b+c)$

$A \cap (B \cap C) \equiv (A \cap B) \cap C$

$A \cup (B \cup C) \equiv (A \cup B) \cup C$

Distributive Law

Example: $a \cdot (b+c) \equiv (a \cdot b) + (a \cdot c)$

$A \cap (B \cup C) \equiv (A \cap B) \cup (A \cap C)$

$A \cup (B \cap C) \equiv (A \cup B) \cap (A \cup C)$

Identity Law

$A \cap U \equiv A$

$A \cup \emptyset \equiv A$

Domination Law

$A \cup U \equiv U$

$A \cap \emptyset \equiv \emptyset$

Set Attributes - Part 2

Fundamental Products

Is an intersection of each set

Three major attributes

1) There are $m = 2^n$ fundamental products $n = \#$ of sets
2) Any two such fundamental products are disjoint
3) The universal set U is the union of all fundamental products

Cardinality

Cardinality of a set is the number of distinct elements

Notation Used: $|A| \equiv n(A)$

Example: $A = \{1,2,3,4,9\}$ & $B = \{1,2,3,3,3,4\}$ Duplicates doesnt count

$|A| = 5$ $|B| = 4$

Inclusion - Exclusion

- Sets can overlap - and can contain the same element
  - Counting items in sets be careful not to count an item twice

Set Exclusion: $|A \cup B| = |A| + |B| - |A \cap B|$

Power Series - Base 2 system

A Power Series of a set S is a set of all the subsets of S including the null set

Notation Used: S is $P(S)$

Example: $G = \{a, b\}$ $|G| = 2$

$P(G) = \{\emptyset, \{a\}, \{b\}, \{a,b\}\}$ $|P(G)| = 4$

$H = \{a, b, c\}$ $|H| = 3$

$P(H) = \{\emptyset, \{a\}, \{b\}, \{c\},$ $|P(H)| = 8$
$\{a,b\}, \{a,c\}, \{b,c\},$
$\{a,b,c\}\}$

☆ The Cardinality of a powerset is 2^n where n is the cardinality of the original set ☆

$$|P(S)| = 2^{|S|}$$

Partitions

A partition of a set A is a collection of non-empty disjoint sets whose union is A

- Each subset MUST be mutually exclusive; unless, they are identical (duplicates do not count)

Example

for the Set $\{1,2,3,4\}$...

Set	Partition
$\{\{1\},\{2\},\{3\},\{4\}\}$	Yes
$\{\{1,2\},\{1,2\},\{3,4\}\}$	Yes $\{1,2\}$ are duplicates
$\{\{1,2,3\},\{2,4\}\}$	NO "2" violates Non exclusive

Sets in Computer Science - Part 3

Binary Numbers

Base 2 number system

- Binary Numbers are tuples (order is important)

  0 1000 10100 $\neq$ 11100000

Example: The number 0100 1010 is

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
0	1	0	0	1	0	1	0

$64+8+2=74$

Bit Vectors

Is a way to store countable sets using bits

Example:
- $U = \{2,3,5,7,11,13,17,19\}$
- $A = \{3,5,11,19\}$

converted to Bit Vectors

U = 1111 1111
A = 0110 1001

Example w/ Union (using OR):

$U = \{a,b,c,d,e,f,g,h\}$
$A = \{b,c,d\} = 0111\ 0000$
$B = \{d,e,f\} = 0001\ 1100$

```
     0111 0000
∪ or 0001 1100
     ---------
     0111 1100 = {b,c,d,e,f}
```

Relations - Part 4

Cross Product - important in database

Sets can be multiplied which will result in a set of tuplets (a set of ordered pairs)

Unlike multiplication, the order of the operand is important

$A \times B = \{(x,y) \mid x \in A \text{ and } y \in B\}$

Example: $A = \{1,2\}$ - Sets of domains (x)

$B = \{x, y\}$ - Sets of range (y)

$A \times B = \{(1,x),(1,y),(2,x),(2,y)\}$

Binary Relations

A binary relation is a stated fact between on two objects - that fact is called a predicate

Examples

- 'x' is bigger than 'y'
- 'x' lives less than 50 mi from 'y'
- 'x' ≤ 'y'
- 'x' has a 'y'
- 'x' & 'y' are siblings

Relations

- Binary Relation from A to B is a subset of the cross-product A×B
- A relation from A to B is a set of ordered pairs (a,b) where $a \in A$ and $b \in B$
- We can use the shorthand notation of: aRb to denote that $(a,b) \in R$

Example

- A is a set of all cities in the world
- B is a set of all States in the world
- The relation aRb specifies that a is the Capital of b

Types of Relations

Relation on

- Some relations of a set A are upon itself
- Each object in the related to the same "type" of object

	1	2	3	4
1	x	x	x	x
2		x		
3			x	
4				x

Types of Relations cont.

Reflexive Relations

Means that there is aRa for every a

- In short everything has to be related to itself

every element, in the domain, must be related to itself

Relation on set {1,2,3,4}

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

	1	2	3	4
1	x			x
2		x	x	
3			x	
4				x

Non-Reflexive

Relation on set {1,2,3,4}

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

	1	2	3	4
1	x	x		
2		x	x	
3				x
4				x

Symmetric Relations

Means that for every aRb there is a bRa

- Summary (a,b) exist in the relation So must (b,a)

Relation on set {1,2,3,4}

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

	1	2	3	4
1	x			x
2		x	x	
3			x	
4				x

Nonsymetric

	1	2	3	4
1	x	x		
2	x			x
3			x	
4		x		

Symetric

Relation on set {1,2,3,4}

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

Back for Rest

Types of Relation

Transitive Relations

Means that for every aRb and bRC that also aRC

- So if (a,b) and (b,C) exist in the relation, so must (a,C)
- Look for an a,b,C where there is a aRb & bRC but NO aRC

Relation on Set {1,2,3,4}

R={(2,1),(3,1),(3,2),(4,1),(4,2),(4,3)} Start w/ (4,3)

R={(2,1),(3,1),(3,2),(4,1),(4,2),(4,3)} Start w/ (3,2)

R={(2,1),(3,1),(3,2),(4,1),(4,2),(4,3)} Start again w/ (4,3)

	1	2	3	4
1				
2	X			
3	X	X		
4	X	X	X	

Composition

The output of one function is used as an input in another

Example

R={(1,2),(3,1),(5,3)}

S={(2,3),(2,6),(3,9)}

∴ R∘S={(1,3),(1,6),(5,9)}

Equivalence Relations

If a relation has all three properties

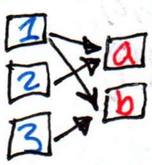
- Reflexive
- symmetric
- Transitive

Then, and only then, it is an equivalance

Functions - A special kind of relation

A function from set X to set Y is a mapping from each element in set X to elements in set Y

Example



{(1,a),(1,b),(2,a),(3,b)}

Rules:

- Must be defined for every element in domain
- Each value in domain maps to one element

Functions Continued

Let f be a relation from A→B

f is a function if and only if: each a∈A appears exactly once in an ordered pair (a,b)∈f for some b

Manipulating Relations

Relations are representable as sets, we can use set notation to define them

Example: A={(1,1),(2,2),(3,3)}
B={(1,1),(2,4),(3,9)}

A∪B={(1,1),(2,2),(2,4),(3,3),(3,9)}

A∩B={(1,1)}

A\B={(2,2),(3,3)}

B\A={(2,4),(3,9)}

Closures

Closures of relation R is the smallest set (when unioned) gives R the desired property

- So, the closure of R is R∪C, Where C is the smallest set giving R∪C the desired property

Example: Reflexive Closure

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

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

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

Symmetric closure

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

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

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

Transitive Closure

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

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

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

∴ R∪C={(1,2),(2,3),(3,4),(1,3),(2,4),(1,4)}

Relations in Computer Science - Part 5

Database Terminology

Information is stored in databases
These systems are based on tuples & sets

- Fields contain the smallest unit of data
  - Number, text
  - Each can be seen as a tuple (it can be a set, but rarely so)
  - Each field has a unique field name
    - Name
    - Age
    - School
- Record is a set of data fields
  - Represents a logical group of data
  - Includes numbers, text, images, ect...
  - Examples
    - Course: Department, Number, Section
    - Student: Name, age, Class
    - Computer: Brand, Speed, cost

Abstract Data Types

Defines
- A set of possible values and operations (functions) that can be performed on those values
  - The basis for all classes and data structures in programming languages