# Set Theory

Part 1

# What is a Set?

Organizing Information

## What is a Set?

- A *set* is an unordered collection of "objects"
- The collection objects are also called "members" or "elements"
- One of the most fundamental structures in mathematics



## Set Notation

- We typically denote a set name using capital letter
- Members are separated with commas and encapsulated within curly brackets



## Standard Sets

| Letter | Name | Members |
|---|---|---|
| Z | Integers | …, -2, -1, 0, 1, 2, 3, … |
| N | Natural Numbers | 1, 2, 3, 4, … |
| Q | Rational Numbers | a / b where both a and b are integers and b is not 0 |



## Standard Sets

| Letter | Name | Members |
|---|---|---|
| R | Real Numbers | All non-imaginary numbers. e.g. 1, 2.5, 3.1415…. |
| U | Universal Set | All values of potential interest (U depends on context) |

## Set Notation: Membership

- Set notation uses a special symbol to denote if an object is a member of a set
- Below, the set V contains vegetables

```
potato ∈ V
```



## Set Notation: Membership

- This is read as "potato is an element of V"
- …or "potato is a member of V"

```
potato ∈ V
```



## Set Notation: Not a Member

- There is another special symbol that denotes an object is not a member of a set
- In the example below, the set F contains fluffy animals

```
lizard ∉ F
```



## Defining Sets

How to specify items

## Defining Sets

- Sets can be defined a number of different ways
- Each competing notation has advantages & disadvantages – depending on what you are defining



## Set Notation: Explicit

- We can *explicitly* define this by listing each element
- For example, we can define a set *S* for members of the Three Stooges

```
S = {moe, larry, curly, shemp}
```

## Set Notation: Pattern

- We can also specify a set by using a *pattern*.
- In the example below we are define a set of integers between 0 and 9.

```
A = {0, 1, 2, ... 9}
```



## Set Builder Notation

- A set can also be defined using *set builder notation*
- Consists of a variable name, a pipe symbol, and an true/false expression



## By Characteristic

- The most basic form consists of a variable and an true/false statement
- In this example, everything that satisfies "x is an even integer" will be the set

```
{x | x is a even integer}
```



## By Characteristic Examples

| Expression | Result |
|---|---|
| { x \| x is an integer } | { …, -1, 0, 1, 2, 3, … } |
| { x \| x is an even integer } | { …, -2, 0, 2, 4, 6, … } |
| { x \| x is odd natural number} | { 1, 3, 5, 7, 9, … } |



## Characteristic with Restriction

- Definitions can also be restricted by another set
- There are two different notations that *mean the same thing*

```
{x ∈ S | true/false expression on x}

{x | x ∈ S and true/false expression on x}
```



## Characteristic Example

- Remember, Z is the set of all integers
- It reads: *"All x where x is in Z and x is even"*

```
A = {x | x ∈ Z and x is even}
```

## By Characteristic Examples

| Expression | Result |
|---|---|
| $\{ x \in Z \mid 0 < x < 5 \}$ | {1, 2, 3, 4} |
| { x \| x ∈ N and x < 7 } | {1, 2, 3, 4, 5, 6} |



## Characteristic with Structure

- The left-hand-side (before the pipe) doesn't have to be a simple variable name
- It can also be any mathematical expression

```
{f(x) | true/false expression using x}

{y | y = f(x) and true/false using x}
```



## Let's Try One…

- The second part of the notation must always be a true/false expression
- So, how do we create a set that contains:

{2, 4, 6, 8, 10, …}



## Let's Try One…

```
First approach:
A = {x | x ∈ N and x is even}

Second approach:
A = {2x | x ∈ N}
```



## How Does It Evaluate?

- Basically, when you look at something like: { 2x | x ∈ N }, you should do the following
- Steps:
  1. Identify which variables make the right-hand-side true
  2. Plug them into the left-hand-side. These are the values in the set.



## More Examples

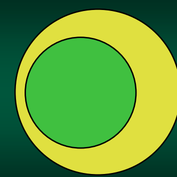
| Expression | Result |
|---|---|
| $\{ 2x + 1 \mid x \in Z \}$ | {…, -3, -1, 1, 3, 5, …} |
| $\{ x \in Z \mid \mathrm{sqrt}(x) \in Z \}$ | {0, 1, 4, 9, 16, 25, …} |

## Empty Set

- An *empty set* contains no elements
- Can be represented with two curly-brackets (nothing in between)
- There is also a special symbol for empty sets

```
A = { }
A = ∅
```



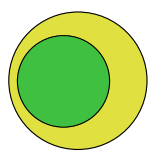


# Subsets

Set Your Sets to Fun

## Subsets

- Commonly, sets are compared to one another using set relationship operators
- Basically, set are defined on elements which they may have in common





## Subsets

- Set A is considered a subset of set B if <u>all</u> the members of A are also members of B
- The subset operator is similar looking to the member operator

$$\{ 1, 4 \} \subseteq \{ 1, 3, 4, 5 \}$$



## Subsets

- A set A is not a subset of B if A contains an element not found in B

$$\{ 3, 5 \} \nsubseteq \{ 3, 7 \}$$



## Null is Always a Subset

- A null set contains no elements
- Hence, the null set is <u>always</u> a subset

$$\emptyset \subseteq \{ 2, 3 \}$$
$$\{\} \subseteq \{ 2, 3 \}$$

## Proper Subsets

- Set A is a *proper subset* of B if A <u>is</u> a subset of B, but <u>not</u> equal to B
- Note: the notation lacks the underline – it is consistent with other operators like < and ≤

```
{ 3, 5 } ⊂ { 3, 5, 7 }
{ 1, 2 } ⊄ { 1, 2 }
```



## Equality

- Sets A and B are considered equal if-and-only-if…

for every $x \in A$ it is also true $x \in B$

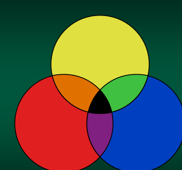
for every $x \in B$ it is also true $x \in A$



## Equality

- So, are { 1, 2, 3 } and { 2, 1, 3 } equal?
- How about { 1, 1, 2, 3, 3 } and { 3, 2, 1 }
- Answer is *yes!*
  - order does not matter in a set
  - multiple occurrences does not change if an element is a member



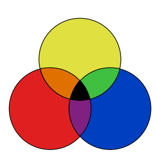


## Venn Diagrams

Graphically Representing Sets

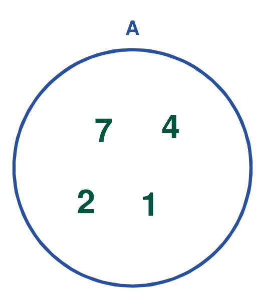
## Venn Diagrams

- Sets can also be abstractly representing graphically using Venn Diagrams
- Each set is represented by circle
- Overlaps between each set can show logical relations with set members

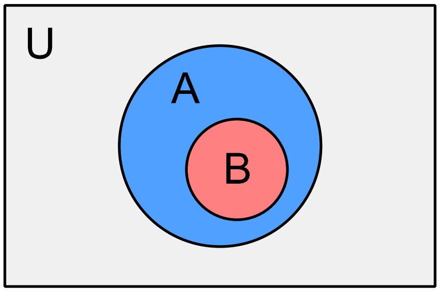




## Example: A = {2, 7, 1, 4}

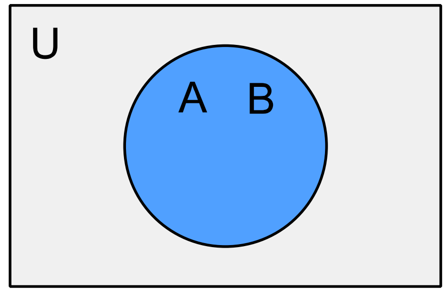


## Subset Venn Diagram



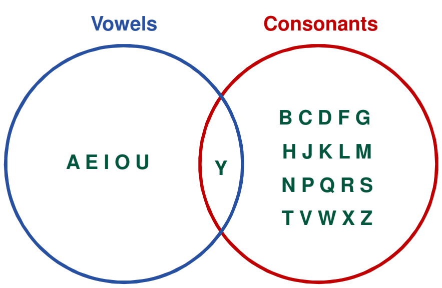



## Equality

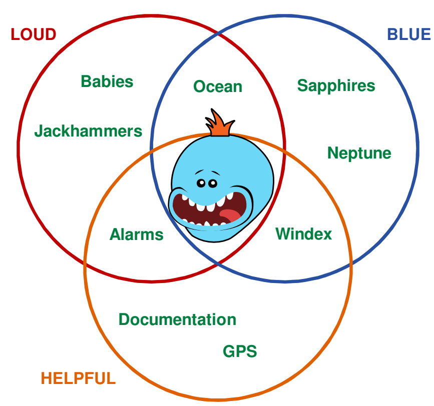


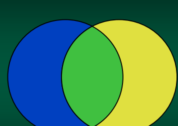



## Example: Letters in English





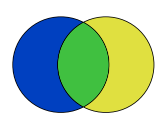



## Set Operators

Defining Sets Using Sets

## Operations on Sets

- New sets can be made from old sets using set operators.
- Just like new numbers can be created from old numbers: 1 + 2 = 3
- So, for the rest of this section, let U be the universe, and let A and B be sets



## Union

- A union of two sets combines all members of each set into a new one
- So, the result is two merged sets

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



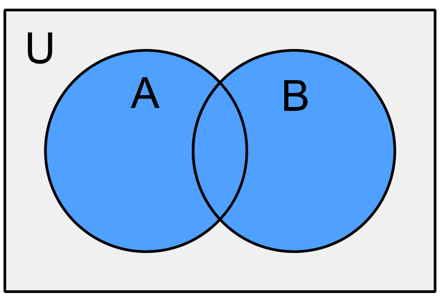
## Union

- The symbol ∪ looks like U
  - which is also used for the "universe set"
  - be careful not the confuse the two

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



## A ∪ B





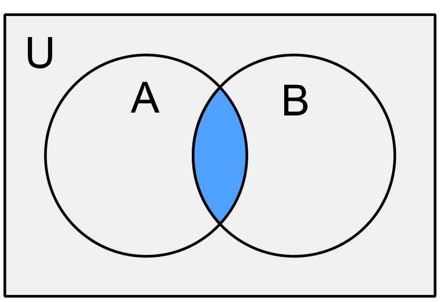

## Intersection

- The intersection of two sets contains only those elements that are found in both sets
- So, the result is where the two sets overlap

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



## A ∩ B



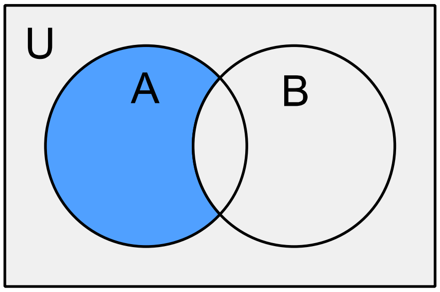



## Difference

- *Difference* (aka exclusion) removes all items found in set from another
- Typically, it is written as A – B even though it is not the same as subtraction

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

## A – B






## Difference – So Many Notations

- The notation for difference varies greatly
- Below are two different variations on the same notation

```
A – B
A \ B
```



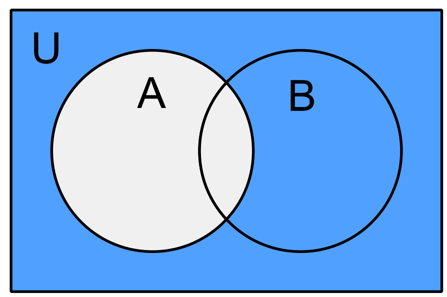
## Complement

- The *complement* of a set A, is all elements in the Universe, <u>not</u> in A
- Remember: what elements are in the Universe depends on the sets

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



## A'






## Different Notations Used

- Single postfix apostrophe
- An "over bar" (which is underlining on top)
- Superscript "c" for complement

$A' \equiv \overline{A} \equiv A^c$



## Complement Example

- If set A is a subset of a set B, then the complement of A is all elements not in A but still in B
- Look at the following:

```
A ⊂ B
A = {1, 2, 3}
B = {1, 2, … 10}
```

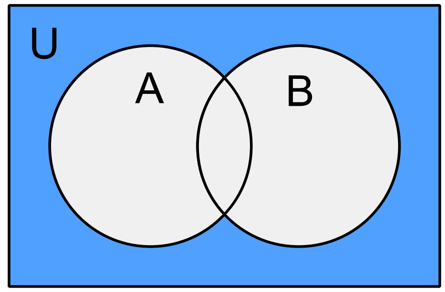
## Complement Example

- Set A is a subset of a set B
- Therefore its "universe" is defined as the set of B

```
Therefore…

A' = {4, 5, 6, 7, 8, 9, 10}
```

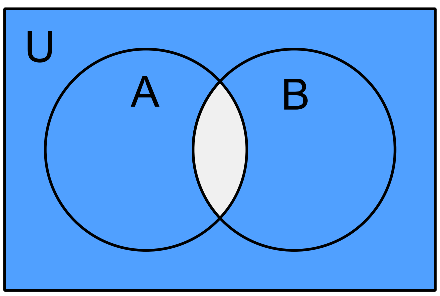


## (A ∪ B) '





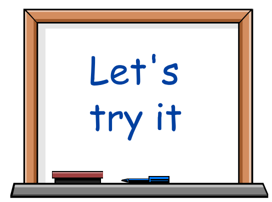

## (A ∩ B) '






## Let's Draw Some…

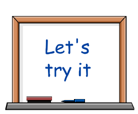
- Let's draw some Venn Diagrams using a several sets
- Using set operators we can highlight any area we want






## Let's Draw Some

```
U = {a, b, c, d, e, f}

A = {a, b, c}
B = {b, c, d, e}
```






## Let's Draw Some

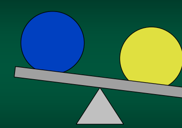
```
Two sets:

In A but not in B: {a}
In B but not in A: {d, e}
In both A and B: {b, c}
In neither A nor B: {f}
```

## Reasoning with Venn Diagrams

Say 100 people receive a questionnaire with two questions (1) Do you program Java?, (2) Do you program C# ?

If 65 said 'yes' to Java, 40 said 'yes' to both. How many program just program C#?



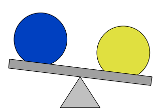


## Cardinality

Counting Sets

## Cardinality of a Set

- The *cardinality* of a set is the number of *distinct* elements
- This information is used in counting – the classification of the set's contents





## Different Notations Used

- There are two different notations used
- The most common is the | pipe delimiters
- Alternatively, the "n" function is used

$$|A| \equiv n(A)$$



## Examples

```
A = {1, 3, 5, 7}

|A| = 4
```



## Examples

```
B = {1, 2, 3, 3, 3, 4}

|B| = 4
```

## Counting

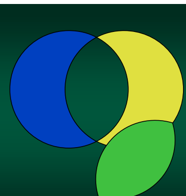
- If the set contains a finite number of elements, it is said to be *countable* – i.e. the cardinality is knowable
- If the set is infinitely large, but the elements can be uniquely identified, then it is *countably infinite*
- Otherwise it is said to be *uncountable*



## Countable Examples

| Set | Result |
|---|---|
| $\{ x \mid x \in N \text{ and } x \leq 100\}$ | Countable |
| $\{ 2x \mid x \in N \}$ | Countably Infinite |
| $\{ x \mid x \in R \text{ and } 0 < x < 1\}$ | Uncountable |



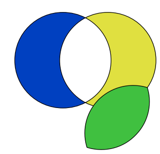


## Inclusion-Exclusion

Counting by Subtracting!

## Inclusion-Exclusion

- Sets can overlap – and can contain the same elements
- So, when counting items in sets, you must be careful not to count an item twice
- *Inclusion-exclusion* principle, can get the correct count





## Disjoint Set Cardinality

- If sets A and B are disjoint then they have no elements in common
- Cardinality of the union is the sum of the cardinality of both A and B

$$|A \cup B| = |A| + |B|$$



## Set Exclusion

- If sets A and B overlap they have elements in common
- The cardinality of the union is the sum of A and B excluding the intersection

$$|A \cup B| = |A| + |B| - |A \cap B|$$

## Set Exclusion

- Why?
- |A| + |B| counts the intersection twice!
- So, we need to remove the duplicate count

$$|A \cup B| = |A| + |B| - |A \cap B|$$



## Set Exclusion

- Also note: this is the same equation for disjoint sets
- If disjoint, the intersection is ∅

$$|A \cup B| = |A| + |B| - |A \cap B|$$



## Let's look at this again…

Say 100 people receive a questionnaire with two questions (1) Do you program Java?, (2) Do you program C# ?

If 65 said 'yes' to Java and 40 say 'yes' to both. How many program just program C#?



## Using the Formula…

- The union of Java & C# contains 100
- Java set contains 65
- The intersection contains 40

$$|J \cup C| = |J| + |C| - |J \cap C|$$



## Using the Formula…

- The union of Java & C# contains 100
- Java set contains 65
- The intersection contains 40

$$100 = 65 + |C| - 40$$



## Using the Formula…

- The union of Java & C# contains 100
- Java set contains 65
- The intersection contains 40

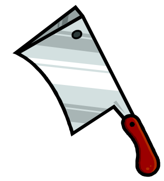
$$|C| = 100 - 65 + 40$$
$$= 75$$

# Partitions

Cutting a Set Into Pieces

## Partitions

- A *partition* of a set A is a collection of non-empty disjoint sets whose union is A
- So, it is like the set A was "chopped", cleanly, into subsets





## Requirements

- Each subset must be mutually exclusive
- … unless they are identical *(because duplicates don't count in sets)*



## Partition Example

- The following is a valid partition of the set {1, 2, 3, … 9}

```
{ {1}, {2,3,5,7}, {4,6}, {8,9} }
```



## Partition Example

- The following is a partition of N.

```
N = { {1}, {2,3}, {4,5,6}, ... }
```



## Countable Examples

For the set {1, 2, 3, 4}…

| Set | Partition? |
|---|---|
| `{ {1}, {2}, {3}, {4} }` | **Yes** |
| `{ {1,2}, {1,2}, {3,4} }` | **Yes.** {1,2} is duplicate |
| `{ {1,2,3}, {2,4} }` | **No.** |

# Power Series

All the combinations

## Power Series

- A *power set* of a set S is a set of all the subsets of S
- This also, obviously, contains the null set
- The notation for the power set S is *P*(S)



## Power Set Example

```
G = {a, b}

P(G) = { ø, {a}, {b}, {a,b} }
```



## Power Set Example 2

```
H = {a, b, c}

P(H) = { ø, {a}, {b}, {c},
         {a,b}, {a,c}, {b,c}
         {a,b,c} }
```



## Power Set Example 3

```
I = {a, b, c, d}

P(I) = { ø,
         {a}, {b}, {c}, {d}
         {a,b}, {a,c}, {a,d},
         {b,c}, {b,d}, {c,d},
         {a,b,c}, {a,b,d}, {b,c,d}, {a,c,d}
         {a,b,c,d} }
```



## Cardinality of Power Sets

- What is the cardinality of power sets?
- Is it possible to determine | P(S) | if we know | S |
- This will be important later…

## Let's Look at the Examples

G = {a, b}

P(G) = { ∅, {a}, {b}, {a,b} }

|G| = 2
|P(G)| = 4



## Power Set Example 2

H = {a, b, c}

P(H) = { ∅, {a}, {b}, {c},
{a,b}, {a,c}, {b,c}
{a,b,c} }

|H| = 3
|P(H)| = 8



## Power Set Example 3

I = {a, b, c, d}

P(I) = {∅, {a}, {b}, {c}, {d}
{a,b}, {a,c}, {a,d},
{b,c}, {b,d}, {c,d},
{a,b,c}, {a,b,d}, {b,c,d}, {a,c,d}
{a,b,c,d}}

|I| = 4
|P(I)| = 16

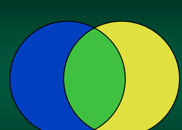


## Cardinality of Power Set

- The cardinality of a power set is $2^n$ where n is the cardinality of the original set
- This is used in statistics… covered later

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





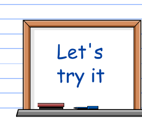
## Set Algebra

Just a preview…

## Let's Look at These…

Let's see if the following are same:

A ∪ (B ∩ C)




(A ∪ B) ∩ (A ∪ C)

## Set Algebra

- We will cover more of this later, but set algebra shares the same principles as basic math
- You can visually treat the union as a "*" and the intersection as a "+"
- You can then factor out sets



## Another Look

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

$\rightarrow (A * B) + (A * C)$

$= A * (B + C)$

$A \cup (B \cap C)$

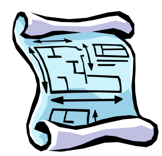


## Tuples

Order is Important

## Tuples & Sets

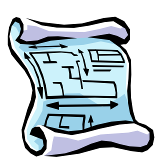
- To denote sets, we delimit the list of members with curly brackets
- For example, the prime numbers between 1 and 10 is {2, 3, 5, 7}
- Order does not matter, so {2, 3, 5, 7} = {7, 5, 3, 2}





## Tuples

- However, in many cases the *order is important*
- These are called *n-tuples* where "n" is the number of elements
- 2-tuples are also called *ordered pairs*





## Tuple Notation

- To denote a tuple – we delimit the elements by either parenthesis, angle brackets or square brackets
- Curly-brackets are never used to avoid obvious confusion

```
( 1, 2, 3 )
< 1, 2, 3 >
[ 1, 2, 3 ]
```

## Tuple Examples

- Order is important, so any element out of position will cause inequality

```
( 1, 2, 3 ) ≠ ( 3, 2, 1 )
```

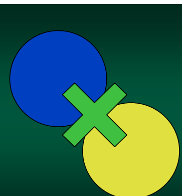


## Tuple Examples

- Logic generally applies to algorithms since, in procedural programming, order is important
- The following is a tuple of events in California History

```
( Sutter's Fort Built, Bear Flag Revolt,
  Gold Rush, California Joins Union )
```



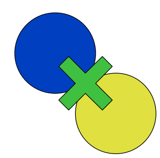


## Cross Products

Databases use this…

## Products

- Sets can be multiplied, which will result in a set of tuples
- Well, a set of ordered pairs, to be more specific
- A × B =
  { (x, y) | x ∈ A and y ∈ B }





## Example Product

```
Given:
  A = {1, 2}
  B = {x, y}

A × B = { (1, x), (2, x),
          (1, y), (2, y) }
```