Discrete Mathematics

Lecture 1

Outline, motivations and speaking mathematically

Textbook: Discrete Mathematics with Applications, 4th Ed. - Susanna S. Epp

What is discrete mathematics?

> *"Discrete mathematics is the part of mathematics devoted to the study of discrete (as opposed to continuous) objects."*
>
> \- Rosen, Discrete Mathematics and its Applications, Ch. 1

Think of it this way:

Discrete objects are countable and separated - like steps on a staircase.

Continuous objects flow smoothly - like a ramp.

A digital clock shows 3:00, then 3:01 - discrete jumps.

An analog clock's hands sweep smoothly - continuous motion.

Core areas: logic, sets, combinatorics, graph theory, number theory, relations, functions

Discrete vs. continuous

Discrete	Continuous
Countable, separated values	**Smooth, unbroken values**
Integers: 1, 2, 3, ...	Real numbers: 3.14159...
Characters: a, b, c, ...	Temperature over time
True / False	Speed of a car
Pixels on a screen	Water level in a tank
Number of students in a class	Height of a growing plant
"How many?" questions	*"How much?" questions*

A brief history

1736 **Euler and the Konigsberg bridges**
Can you cross all 7 bridges exactly once? Euler proved you cannot - and invented graph theory.

1847 **Boole's algebra of logic**
George Boole showed that logical reasoning can be reduced to algebraic calculation - the foundation of digital circuits.

1874 **Cantor and set theory**
Georg Cantor formalised the concept of a set and proved that some infinities are larger than others.

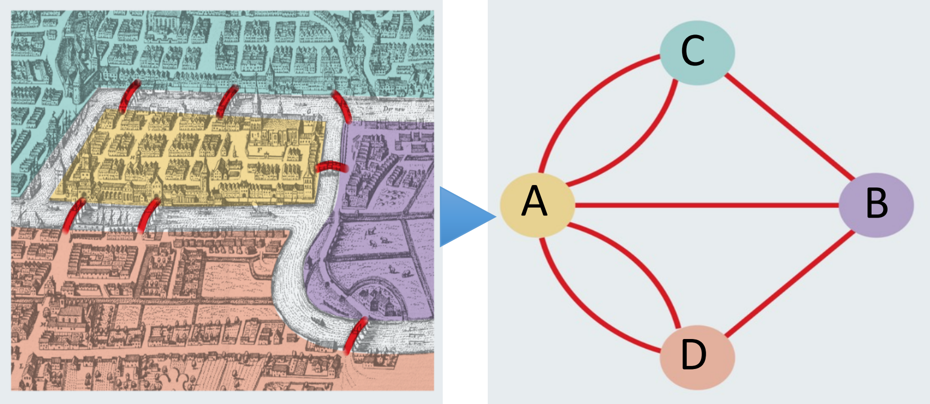
1936 **Turing and computability**
Alan Turing defined what it means for a problem to be 'computable' - discrete math meets computer science.

Each of these breakthroughs started with a simple question about countable, finite objects.

Try it: the Konigsberg bridges

Can you trace a path that crosses each of the 7 bridges exactly once?

The graph model



Euler's insight

Count the edges (bridges) at each node:

```
A has 3 bridges  (odd)
B has 3 bridges  (odd)
C has 5 bridges  (odd)
D has 3 bridges  (odd)
```

All four nodes have an odd number of edges.

An Eulerian path needs at most 2 odd-degree nodes. With 4, it is impossible.

Discrete math turns a physical puzzle into a provable result.

But why do we need to study it, anyway?

Programming and algorithms

Loops, conditionals, recursion - all built on logic and counting. Sorting 1,000 names is a discrete problem.

Cryptography and security

Your bank's website uses RSA encryption, which relies on prime numbers and modular arithmetic.

Databases

SQL queries are logical expressions. Tables are relations. A JOIN is a set operation.

Networks

Finding the fastest route on Google Maps is a graph theory problem (Dijkstra's algorithm).

Where you will see it

Search engines
PageRank uses graph theory to rank web pages

AI and machine learning
Decision trees, logic, and probability

Scheduling
Exam timetables, airline crew assignments

Social networks
Friend recommendations via graph algorithms

Digital signatures
Verifying a document has not been altered

Compiler design
Parsing code with formal grammars

Course roadmap

Ch 1 — **Speaking mathematically**
Variables, sets, relations, functions

Ch 2 — **Propositional logic**
Compound statements, truth tables

Ch 3 — **Quantified statements**
Predicates, quantifiers

Ch 4 — **Number theory and proof**
Direct proof, contradiction, divisibility

Ch 5 — **Sequences and induction**
Mathematical induction, recursion

Ch 6 — **Set theory**
Set operations, identities, proofs

Ch 7 — **Functions**
One-to-one, onto, composition

Ch 8 — **Relations**
Equivalence relations, partial orders

Ch 9 — **Counting**
Permutations, combinations, pigeonhole

Ch 10 — **Graphs and trees**
If time permits

Chapter 1

Speaking mathematically

Sections 1.1 - 1.3 | Epp, Ch. 1

1.1 Variables

A **variable** is a symbol that stands for a value from a specified set, called its **domain**.

Examples

```
Is there a number  x  such that  x + 3 = 8 ?
```
x is a variable. It represents an unknown. Here, x = 5.

```
For every positive integer  n,   2n  is even.
```
n is a variable ranging over all positive integers (its domain).

```
If  x > 2  then  x + 1 > 3.
```
x is a variable. The statement is true no matter what value x takes.

```
Let  p = the price of an item.  If  p > 100  then  tax = p * 0.15.
```
Variables in programming work the same way - a name standing for a value.

1.1 Types of statements

Universal statement

"For all elements in a set, a property holds."

All dogs are mammals.

Every even number is divisible by 2.

To disprove: find one counterexample.

Existential statement

"There exists at least one element with a property."

There is a prime number that is even.
(Yes - the number 2.)

Some student scored 100 on the exam.

To prove: find one example.

Universal conditional statement

"For all x, if P(x) then Q(x)." Combines both forms.

For all students, if you pass all quizzes then you pass the course.

1.2 The language of sets

A **set** is a well-defined collection of distinct objects, called **elements** or **members**.

- Epp, Section 1.2

Roster notation

List elements inside braces.

```
A = { 1, 2, 3, 4 }
V = { a, e, i, o, u }
Days = { Mon, Tue, Wed, ... }
```

Set-builder notation

Describe elements by a rule.

```
B = { x in Z | x > 0 }
```

read: "x in the integers such that x > 0"

```
E = { 2n | n in Z }
```

read: "2n where n is any integer" (all even numbers)

"Well-defined" means there is no ambiguity about membership: an object is either in the set or not.

1.2 Membership and standard sets

Membership notation

```
x  ∈  S        "x is an element of S"
```

Example: $3 \in \{ 1, 2, 3 \}$ (true)

```
x  ∉  S        "x is not an element of S"
```

Example: $5 \notin \{ 1, 2, 3 \}$ (true)

Symbol	Name	Contents	Example elements
$\mathbb{N}$	Natural numbers	0, 1, 2, 3, ...	0, 7, 42
$\mathbb{Z}$	Integers (Zahlen)	..., -2, -1, 0, 1, 2, ...	-3, 0, 5
$\mathbb{Q}$	Rationals	Fractions p/q ($q \neq 0$)	1/2, -3/4, 7
$\mathbb{R}$	Real numbers	All points on the number line	π, $\sqrt{2}$, -0.5

1.2 Subsets

A ⊆ B means every element of A is also an element of B. Read: "A is a subset of B."

- Epp, Section 1.2

Examples

`{ 1, 3 } ⊆ { 1, 2, 3, 4 }` *Every element of the left set (1 and 3) is found in the right set.*	`true`
`{ 1, 5 } ⊆ { 1, 2, 3, 4 }` *5 is not in the right set, so the subset relation fails.*	`false`
`{ a, b } ⊆ { a, b, c, d }`	`true`
`Z ⊆ R` *Every integer is a real number.*	`true`

1.2 Ordered pairs and Cartesian products

Ordered pair

(a, b) - order matters.

```
(1, 2)  ≠  (2, 1)

(a, b) = (c, d)  iff  a = c and b = d
```

Why "ordered"?

GPS coordinates: (lat, long)

```
(51.5, -0.1) = London
(-0.1, 51.5) = somewhere in Nigeria
```

Swapping changes the meaning entirely.

Cartesian product

```
A x B  =  { (a, b)  |  a in A  and  b in B }
```

If Sizes = { S, M, L } and Colours = { red, blue }, then:

```
Sizes x Colours = { (S,red), (S,blue), (M,red), (M,blue), (L,red), (L,blue) }
```

This gives every possible T-shirt combination - 3 x 2 = 6 pairs.

The Cartesian plane R x R = R^2 is the most familiar example.

1.3 The language of relations

A **relation** from set A to set B is a subset of A x B.

If (a, b) is in the relation, we write a R b.

- Epp, Section 1.3

Examples from everyday life

"is less than" on integers

2 < 5 means (2, 5) is in the relation. 3 < 1 is false, so (3, 1) is not.

"is enrolled in" between students and courses

```
Students = { Ali, Ruba }    Courses = { Math, CS }
```

If Ali takes Math and CS, and Ruba takes CS:

```
R = { (Ali, Math), (Ali, CS), (Ruba, CS) }
```

"is the capital of" between cities and countries

```
R = { (London, UK), (Paris, France), (Tokyo, Japan), ... }
```

1.3 The language of functions

A **function** from A to B is a relation where every element of A is related to **exactly one** element of B.

`Notation:   f : A -> B        f(a) = b`

- Epp, Section 1.3

What makes it a function?

Function (valid)

`Student -> student ID`
Each student has exactly one ID.

`Country -> capital city`
Each country has exactly one capital.

Not a function (invalid)

`Student -> phone number`
A student might have two phones.

`Person -> child`
A person can have multiple children.

A is the **domain** (inputs). B is the **co-domain** (possible outputs). f(a) is the **image** of a.

1.3 Function examples

f(x) = 2x + 1   domain: Z   co-domain: Z

f(0) = 1,   f(1) = 3,   f(2) = 5,   f(-1) = -1

Each integer input gives exactly one integer output.

len(s) = length of string s   domain: all strings

len("") = 0,   len("hi") = 2,   len("hello") = 5

A familiar function from programming.

floor(x) = largest integer <= x   domain: R   co-domain: Z

floor(3.7) = 3,   floor(-1.2) = -2,   floor(5) = 5

Used in programming whenever you truncate a decimal.

mod(n, d) = remainder when n is divided by d

mod(17, 5) = 2,   mod(12, 4) = 0,   mod(7, 2) = 1

Central to cryptography and number theory (Chapter 4).

Key takeaways

- **Discrete math studies countable, separated objects**
  Integers, sets, graphs, logical values - not smooth curves.
- **Variables have domains**
  A variable is only meaningful within its specified set of possible values.
- **Statements can be universal or existential**
  "For all" claims need general proofs. "There exists" claims need one witness.
- **Sets are collections; notation matters**
  { }, ∈, ∉, ⊆, and set-builder notation are the vocabulary of mathematics.
- **Relations are sets of ordered pairs**
  They connect elements across sets. Functions are a special kind of relation.
- **Functions map each input to exactly one output**
  f : A -> B, where every a in A has precisely one f(a) in B.

Next session

Chapter 2

The logic of compound statements

Sections 2.1 - 2.3

Logical connectives (and, or, not, if-then)

Truth tables and logical equivalence

Valid and invalid arguments