Discrete Mathematical Models

Lecture 8

Kane Townsend

Semester 2, 2024

Back to classification

Classifications and relations

Recall our discussion about classifying the elements of a set S. One way to think about a classification is through the lens of relations. Two elements of S are related (in some way) if you classify them as having the same type. Not every relation can be thought of as a relation that is related to classification. We can, however, identify properties of relations that provide 'excellent' classifications.

Properties of relations

Let S be a set and let $\sim$ be a relation on S. We say that

- $\sim$ is **reflexive** when $\forall s \in S \; s \sim s$;
- $\sim$ is **symmetric** when

$$\forall s, t \in S \; s \sim t \to t \sim s;$$

- $\sim$ is **transitive** when

$$\forall s, t, u \in A \; ((s \sim t) \wedge (t \sim u)) \to (s \sim u);$$

- $\sim$ is an **equivalence relation on** S when $\sim$ is reflexive, symmetric and transitive.

The equivalence relation you are most familiar with is the relation $=$ on different sets of numbers such as $\mathbb{N}, \mathbb{R}, \mathbb{Q}, \mathbb{Z}$.

Motivation

Equivalence relations appear throughout all mathematics, because this allows us to define ‘sameness’.

- Equality of numbers.
- Congruence in modular arithmetic.
- Isomorphism of groups.
- Equality of sets.
- Homeomorphisms of topological spaces.
- Isomorphism of graphs.
- Quasi-isometry of metric spaces.

Equivalence relations give us a way to say two things are the same (depending on our purposes) up to some level (eg. labelling or naming).

Some thoughts

On a previous slide I claimed that '$=$' is an equivalence relation on $\mathbb{Z}_{\geq 0}$.

The idea is to define a relations that has elements like $1 + 3 = 2 + 2$, $2 = 1 + 1$ etc.

We will not go into the details of this, we will instead describe how to construct the rational numbers from the the integers.

Equivalence classes

Let S be a set and let $\sim$ be an equivalence relation on S. For each $s \in S$, the $\sim$-**equivalence class** of s, denoted $[s]_\sim$, is the set

$$\{t \in S \mid s \sim t\}$$

Theorem: Let S be a set and let $\sim$ be an equivalence relation on A. The set $\mathcal{A}$ of $\sim$-equivalence classes is a partition of S

Example

Let $F = \{(n, d) \mid n, d \in \mathbb{Z} \text{ and } d \neq 0\}$.

For all $(n_1, d_1), (n_2, d_2) \in F$, we write

$$((n_1, d_1) \sim (n_2, d_2)) \Leftrightarrow (n_1 d_2 = n_2 d_1).$$

Theorem: $\sim$ is an equivalence relation on F.

Let T be the set of $\sim$ equivalence classes.

Q: The set T may seem familiar. Do you know something like it?

Example

Let $F = \{(n, d) \mid n, d \in \mathbb{Z} \text{ and } d \neq 0\}$.

For all $(n_1, d_1), (n_2, d_2) \in F$, we write

$$((n_1, d_1) \sim (n_2, d_2)) \Leftrightarrow (n_1 d_2 = n_2 d_1).$$

Theorem: $\sim$ is an equivalence relation on F.

Let T be the set of $\sim$ equivalence classes.

Q: The set T may seem familiar. Do you know something like it? We have classified the ordered pairs of integers (with the second element non-zero) by the rational number they represent.

Coda

The preceding example demonstrates one of the key uses of equivalence relations: Equivalence relations are a tool for recognising when two labels/names are just different labels/names for the same thing.

For example: It is convenient to allow the same rational number to be represented by different fractions. Our construction does this. We write $\frac{n}{d}$ as shorthand for $[(n, d)]_\sim$. Then we have

$$\frac{2}{3} = \frac{4}{6} = \frac{-8}{-12} = \dots$$

We have 'constructed' or 'defined' the rational numbers from the integers:

$$\mathbb{Q} = \{[n, d]_\sim | n, d \in \mathbb{Z} \text{ and } d \neq 0\}$$

Some more discussion

ZFC defines the nonnnegative integers recursively using the **Successor function** $S(n) = n \cup \{n\}$ and setting $0 = \{\} = \emptyset$.

That is $1 = \{\emptyset\}, 2 = \{0, 1\} = \{\emptyset, \{\emptyset\}\}, 3 = \{0, 1, 2\} = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$.

We then need to understand addition and equality on $\mathbb{N}$ to explain for example $1 + 3 = 2 + 2$.

You can then introduce an equivalence relation on $\mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0}$ to construct the integers $\mathbb{Z}$.

The previous slides explain how to construct $\mathbb{Q}$ from $\mathbb{Z}$.

Equivalence relations can then be used to construct $\mathbb{R}$ from $\mathbb{Q}$. This requires an understanding analysis (Cauchy sequences, completions, limits, metrics). Not really in the scope of discrete mathematics.

Partially and Totally Ordered Sets

Let S be a set and let $\leq$ be a relation on S.

$\sim$ is **antisymmetric** when $\forall s, t \in S\ ((s \sim t) \wedge (t \sim s)) \rightarrow s = t$

If $\sim$ is reflexive, antisymmetric, and transitive, then we say that $\sim$ is a **partial order on** S.

Exercise: Prove that $\subseteq$, defines a partial order on $S = \mathcal{P}(U)$, where U is a known set.

Partially and Totally Ordered Sets

Let S be a set and let $\leq$ be a relation on S.

$\sim$ is **antisymmetric** when $\forall s, t \in S\ ((s \sim t) \wedge (t \sim s)) \rightarrow s = t$

If $\sim$ is reflexive, antisymmetric, and transitive, then we say that $\sim$ is a **partial order on** S.

Exercise: Prove that $\subseteq$, defines a partial order on $S = \mathcal{P}(U)$, where U is a known set.

$\sim$ is **comparable** when $\forall s, t \in S\ (s \sim t) \vee (t \sim s)$.

If $\sim$ is reflexive, antisymmetric, transitive, and comparable then we say that $\sim$ is a **total order on** S.

Exercise: Prove that $\leq$, defines a total order on $\mathbb{N}$.

Section B: Digital Information

Section B1: Representing numbers

Positional notation

Let b, called the **base**, be an integer such that $b \geq 2$, and let $d_0, d_1, \ldots, d_{b-1}$ be symbols, called **digits**, that represent the first b non-negative integers in order.

Any non-negative integer number can be written in exactly one way using these digits and **positional notation** as follows: For any

$$x_{i_s}, x_{i_{s-1}}, \ldots x_{i_0} \in \{d_0, d_1, \ldots, d_{b-1}\},$$

the expression

$$(x_{i_s} x_{i_{s-1}} \ldots x_{i_0})_b$$

means

$$x_{i_s} \times b^s + x_{i_{s-1}} \times b^{s-1} + \ldots x_{i_0} \times b^0$$

Decimal

$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ is the set of **decimal digits**. These digits are used for representing numbers in base 10.

$$(63403)_{10}$$
$$= (6 \times 10^4 + 3 \times 10^3 + 4 \times 10^2 + 0 \times 10^1 + 3 \times 10^0)_{10}$$

Binary

$\{0,1\}$ is the set of **binary digits**, usually called **bits** for short. These digits are used for representing numbers in base 2.

$$
\begin{aligned}
&(10111)_2 \\
&= (1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0)_{10} \\
&= (16 + 0 + 4 + 2 + 1)_{10} \\
&= (23)_{10}
\end{aligned}
$$

Octal

$\{0, 1, 2, 3, 4, 5, 6, 7\}$ is the set of **octal digits**. These digits are used for representing numbers in base 8.

$$
\begin{aligned}
&(63403)_8 \\
&= (6 \times 8^4 + 3 \times 8^3 + 4 \times 8^2 + 0 \times 8^1 + 3 \times 8^0)_{10} \\
&= (24576 + 1536 + 256 + 0 + 3)_{10} \\
&= (26371)_{10}
\end{aligned}
$$

Hexadecimal

$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$ is the set of **hexadecimal digits**. These digits are used for representing numbers in base 16.

$$
\begin{aligned}
&(3AD0F)_{16} \\
&= (3 \times 16^4 + 10 \times 16^3 + 13 \times 16^2 + 0 \times 16^1 + 15 \times 16^0)_{10} \\
&= (196608 + 40960 + 3328 + 0 + 15)_{10} \\
&= (240911)_{10}
\end{aligned}
$$

Notational conventions

We often omit the parentheses, or omit both the parentheses and the subscript indicating the base when the omission is unlikely to cause misunderstanding.

Example: We often write 11010100_2 instead of $(11010100)_2$. In the middle of a page of computations in binary, we might write simply write 11010100.

In the frames above you also saw that we sometimes group sums and products inside a single set of parentheses to indicate the base in which all the numbers are represented.

Converting from decimal to binary

Suppose that $n \in \mathbb{N}$ is represented in decimal. To write n in binary: write n as a sum of powers of 2. Start by finding the largest power of 2 that does not exceed n.

$$
\begin{aligned}
(71)_{10} &= (64 + 7)_{10} \\
&= (64 + 4 + 3)_{10} \\
&= (64 + 4 + 2 + 1)_{10} \\
&= (2^6 + 2^2 + 2^1 + 2^0)_{10} \\
&= (1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 \\
&\qquad + 1 \times 2^1 + 1 \times 2^0)_{10} \\
&= (1000111)_2
\end{aligned}
$$

An example

$$
\begin{aligned}
(347)_{10} &= (256 + 91)_{10} \\
&= (256 + 64)_{10} \\
&= (256 + 64 + 27)_{10} \\
&= (256 + 64 + 16 + 11)_{10} \\
&= (256 + 64 + 16 + 8 + 3)_{10} \\
&= (256 + 64 + 16 + 8 + 2 + 1)_{10} \\
&= (2^8 + 2^6 + 2^4 + 2^3 + 2^1 + 2^0)_{10} \\
&= (1 \times 2^8 + 0 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 \\
&\qquad + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0)_{10} \\
&= (101011011)_2
\end{aligned}
$$

Converting between binary and octal

Suppose that $n \in \mathbb{N}$ is represented in binary. To write n in octal: collect the bits into groups of 3, then replace each group of 3 bits by the appropriate octal digit.

$$\begin{aligned}(1011110)_2 &= (\underbrace{1}\underbrace{011}\underbrace{110})_2 \\ &= (\underbrace{001}\underbrace{011}\underbrace{110})_2 \\ &= (136)_8\end{aligned}$$

Reverse the process the go from octal to binary.

Converting between binary and hexadecimal

Suppose that $n \in \mathbb{N}$ is represented in binary. To write n in hexadecimal: collect the bits into groups of 4, then replace each group of 4 bits by the appropriate hexadecimal digit.

$$
\begin{aligned}
(1011110)_2 &= (\underbrace{\ 101}\underbrace{1110})_2 \\
&= (\underbrace{0101}\underbrace{1110})_2 \\
&= (5E)_{16}
\end{aligned}
$$

Reverse the process the go from hexadecimal to binary.

Why binary?

Many real work phenomena exist in one of two states: current flowing or no current flowing; a north pole or south pole of a magnet; at each location on a laser disc, we either burned a tiny hole or we did not burn a tiny hole. By interpreting one of the states as 0 and the other as 1, we can represent information. We simply need to agree upon how to encode information in the form of 0's and 1's. It is therefore natural to represent numbers in binary, rather than encode decimal digits as strings of 0's and 1's and then try to make numerical meaning from them.

Why octal and hexadecimal?

A page of binary is difficult for humans to read.

The same information is written in a more compact form on a page, and is more easily read by humans, in octal or hexadecimal.

Conversions between binary, octal and hexadecimal are convenient.

Some vocabulary

A digit in base 2 is called a **bit**. This is simply a contraction of **bi**nary digi**t**. It was first used in information theory by Tukey in 1948.

A block of 8 bits is called a **byte**. It was first used by IBM around the 60's when they started using an 8-bit character code known an EBCDIC.

A block of 4 bits is called a **nibble**.

A sequence of several adjacent bytes is called a **word**. The number of bytes varies, depending on the purpose of the word. For example, a 2-byte word can store non-negative integers in the range from 0 to $2^1 6 - 1 = 65535$.

Addition in decimal

In base 10:

$$123 + 678 : \quad \begin{array}{rrrrl} & 1 & 2 & 3 & \\ + & 6 & 7 & 8 & \\ & 1 & 1 & & \text{carry digits} \\ \hline & 8 & 0 & 1 & \\ \hline \end{array}$$

Addition in binary

In base 2:

$$111_2 + 10_2 : \quad \begin{array}{cccccl} & 1 & 1 & 1 & & \\ + & & 1 & 0 & & \\ 1 & 1 & & & & \text{carry bits} \\ \hline 1 & 0 & 0 & 1 & & \\ \hline \end{array}$$

Addition in hexadecimal

In base 16:

$$456_{16} + ABC_{16}: \quad \begin{array}{ccccc} & 4 & 5 & 6 & \\ + & A & B & C & \\ & 1 & 1 & & \text{carry bits} \\ \hline & F & 1 & 2 & \\ \hline \end{array}$$