Discrete Mathematics

Yike Guo

This course is based on previous lecture notes by Philippa Gardner and Iain Phillips. The slides are based on those made by Philippa Gardner.

Recommended books

K.H. Rosen. Discrete Mathematics and its Applications, McGraw Hill 1995.

J.L. Gersting. Mathematical Structures for Computer Science, Freeman 1993.

J.K. Truss. Discrete Mathematics for Computer Science, Addison-Wesley 1991.

R. Johnsonbaugh. Discrete Mathematics, Prentice Hall 2000.

C. Schumacher, Fundamental Notions of Abstract Mathematics, Addison-Wesley 2001.

Some related courses

1. Mathematical reasoning: logic
2. Mathematical reasoning: programming
3. Mathematical reasoning: discrete maths (continued)
4. Haskell
5. Databases

In particular, we will use some of the notation introduced in the logic course:

$$A \wedge B \quad A \vee B \quad \neg A \quad A \rightarrow B \quad A \leftrightarrow B \quad \forall x.A \quad \exists x.A$$

Motivation: sets

Sets are like types in Haskell.

Haskell type declaration:

```
data Bool = False | True
```

Set of Boolean values: $\{\mathsf{False}, \mathsf{True}\}$

List of Boolean values: [True, False, True, False]

Set equality $\{\mathsf{False}, \mathsf{True}\} = \{\mathsf{True}, \mathsf{False}, \mathsf{True}, \mathsf{False}\}$

Motivation: functions

Haskell function

```
myand :: Bool -> Bool -> Bool
myand False False = False
myand False True = False
myand True False = False
myand True True = True
```

Computable function described for example using Turing machines.

This course explores the notion of **mathematical function**.

Motivation: relations

Examples

1. The class of the first-year Computer Science students, year 2002.
2. George W Bush is the son of George Bush
3. $2 < 3$
4. One program is equivalent to another program.

Sets

Informal definition

A set is a collection of objects (or individuals) taken from a pool of objects. The objects in a set are also called the elements, or members, of the set. A set is said to contain its elements.

We write $x \in A$ when object x is a member of set A.

We write $x \notin A$, or $\neg(x \in A)$, when x is not a member of A.

Examples

1. vowels $\{a, e, i, o, u\}$
2. arbitrary (nonsense) set $\{1, 2, e, f, 5,$ Imperial$\}$
3. natural numbers $\mathcal{N} = \{0, 1, 2, 3, \ldots\}$
4. integers $\mathcal{Z} = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$
5. primes $\mathsf{P} = \{x \in \mathcal{N} : x$ is a prime number$\}$
6. empty set $\emptyset = \{\ \}$
7. nested sets, such as $\{\{a, e, i, o, u\},\ \{\emptyset\}\}$

Comparing sets: subsets

Let A, B be any two sets, Then A is a subset of B, written $A \subseteq B$, if and only if all the elements of A are also elements of B: that is,

$$A \subseteq B \Leftrightarrow \forall \text{ objects } x.(x \in A \rightarrow x \in B)$$

Object x comes from an underlying universe of discourse, sometimes written U.

Analogy

Similar to sublist from the Haskell course.

The following function `h` takes a list of integers and an integer `n`, and returns a sublist of elements less than `n`:

```
h:: [Int] -> Int -> [Int]
 h xs n = filter (<n) xs
```

Caution: lists are different from sets.

Examples

$$
\begin{aligned}
A &\subseteq A, \quad A \text{ set} \\
\{a, b\} &\subseteq \{a, b, c\} \\
\{c, c, b\} &\subseteq \{a, b, c, d\} \\
\mathcal{N} &\subseteq \mathcal{Z} \\
\emptyset &\subseteq \{1, 2, 5\}
\end{aligned}
$$

Proposition

Let A, B, C be arbitrary sets. If $A \subseteq B$ and $B \subseteq C$ then $A \subseteq C$.

Proof

Assume that A, B and C are arbitrary sets.

Assume that $A \subseteq B$ and $B \subseteq C$.

Assume $x \in A$.

By assumption, we know that $A \subseteq B$.

By the definition of the subset relation, $x \in B$.

We also know that $B \subseteq C$, and hence $x \in C$ as required.

Comparing sets: equality

Let A, B be any two sets. Then A **equals** B, written $A = B$, if and only if $A \subseteq B$ and $B \subseteq A$: that is,

$$A = B \Leftrightarrow A \subseteq B \wedge B \subseteq A$$

The sets $\{a, b, c\}$ and $\{b, a, a, c\}$ are equal sets.

The lists $[a, b, c]$ and $[b, a, a, c]$ are **not** equal lists.

Constructing Sets

List elements inside curly brackets:

$$V = \{a, e, i, o, u\} \quad \mathcal{N} = \{0, 1, 2, \ldots\} \quad \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$$

Define a set by stating the property that its elements must satisfy:

$$P = \{x \in \mathcal{N} : x \text{ is a prime number}\}$$
$$\mathcal{R} = \{x : x \text{ is a real number}\}$$

Basic Set Constructors

Let A and B be any sets:

Union $A \cup B = \{x : x \in A \vee x \in B\}$

Intersection $A \cap B = \{x : x \in A \wedge x \in B\}$

Difference $A - B = \{x : x \in A \wedge x \notin B\}$

Symmetric difference $A \triangle B = (A - B) \cup (B - A)$

Example

Let $A = \{1, 3, 5, 7, 9\}$ and $B = \{3, 5, 6, 10, 11\}$. Then

$$\begin{aligned} A \cup B &= \{1, 3, 5, 6, 7, 9, 10, 11\} \\ A \cap B &= \{3, 5\} \\ A - B &= \{1, 7, 9\} \\ A \triangle B &= \{1, 7, 9, 6, 10, 11\} \end{aligned}$$

It is often helpful to illustrate these combinations of sets using **Venn diagrams**.

Properties of operators

Commutativity

$$A \cup B = B \cup A$$

$$A \cap B = B \cap A$$

Associativity

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

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

Distributivity

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

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

Idempotence

$$A \cup A = A$$

$$A \cap A = A$$

Empty set

$$A \cup \emptyset = A$$

$$A \cap \emptyset = \emptyset$$

Absorption

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

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

Proposition Let A, B and C be arbitrary sets. Then $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$.

Proof Let A, B and C be arbitrary sets. We prove

1. $A \cup (B \cap C) \subseteq (A \cup B) \cap (A \cup C)$
2. $(A \cup B) \cap (A \cup C) \subseteq A \cup (B \cap C)$

To prove **part 1**, assume $x \in A \cup (B \cap C)$ for arbitrary x.

By definition, $x \in A$ or $x \in B \cap C$.

By definition, either $x \in A$, or x is in both B and C.

By distributivity, $x \in A$ or $x \in B$, and $x \in A$ or $x \in C$.

This means that $x \in A \cup B$ and $x \in A \cup C$, and hence $x \in (A \cup B) \cap (A \cup C)$. **Exercise** Prove **part 2**.

Proposition Let A, B and C be arbitrary sets. Then $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$.

Alternative Proof

Let A, B and C be arbitrary sets.

$$\begin{aligned} A \cup (B \cap C) &= \{x : x \in A \vee x \in (B \cap C)\} \\ &= \{x : x \in A \vee (x \in B \wedge x \in C)\} \\ &= \{x : (x \in A \vee x \in B) \wedge (x \in A \vee x \in C)\} \\ &= \{x : (x \in A \cup B) \wedge (x \in A \cup C)\} \\ &= \{x : x \in (A \cup B) \cap (A \cup C)\} \end{aligned}$$

We will go through some more examples in the tutorial.

Counter-examples

The statement $A \cup (B \cap C) = (A \cap B) \cup (A \cap C)$ is **false**.

A simple **counter-example** is $A = \{a\}$, $B = \{b\}$ and $C = \{c\}$, where a, b and c are different. The statement is true when $A - (B \cup C) = \emptyset$ and $(B \cap C) - A = \emptyset$.

The statement $A \cup (B \cap C) = (A \cap B) \cup C$ is **false**.

A **counter-example** is $A = \{a\}$, $B = \emptyset$ and $C = \{c\}$ for a different from c. The statement is true when $A - (B \cup C) = \emptyset$ and $C - (A \cup B) = \emptyset$.

Cardinality

Definition

Let A be a **finite** set. The **cardinality** of A, written $|A|$, is the number of elements contained in A.

Notice the similarity with the length function over lists.

Examples

$$\begin{aligned} |\{a, e, i, o, u\}| &= 5 \\ |\emptyset| &= 0 \\ |\mathcal{N}| &= \text{undefined for now} \end{aligned}$$

Proposition

Let A and B be finite sets. Then $|A \cup B| = |A| + |B| - |A \cap B|$

Informal proof

The number $|A| + |B|$ counts the elements of $A \cap B$ twice, so we abstract $A \cap B$ to obtain the result.

A consequence of this proposition is that, if A and B are disjoint sets, then $|A \cup B| = |A| + |B|$.

Powerset

Definition Let A be any set. Then the **powerset** of A, written $\mathcal{P}(A)$, is $\{X : X \subseteq A\}$.

Examples

$$\begin{aligned} \mathcal{P}(\{a,b\}) &= \{\emptyset, \{a\}, \{b\}, \{a,b\}\} \\ \mathcal{P}(\emptyset) &= \{\emptyset\} \\ \mathcal{P}(\mathcal{N}) &= \{\emptyset, \{1\}, \{2\}, \ldots, \{1,2\}, \{1,3\}, \ldots, \\ &\quad \{2,3\}, \ldots, \{1,2,3\}, \ldots\} \end{aligned}$$

Proposition Let A be a finite set with $|A| = n$. Then $|\mathcal{P}(A)| = 2^n$. Proof given in lectures and notes. Not required

Cartesian (or binary) product

An **ordered pair** (a, b) is a pair of objects a and b where the order of a and b matters.

For any objects a,b, c, d, we have $(a, b) = (c, d)$ if and only if $a = c$ and $b = d$.

Definition

Let A and B be arbitrary sets. The **Cartesian (or binary) product** of A and B, written $A \times B$, is

$$\{(a, b) : a \in A \wedge b \in B\}.$$

We sometimes write A^2 instead of $A \times A$.

Examples

1. The coordinate system of real numbers $\mathcal{R}^2$.
2. Computer marriage bureau: let M be the set of men registered and W the set of women, then the set of all possible matches is $M \times W$.
3. Products are analogous to the product types of Haskell.

   (Int, Char) is Haskell's notation for the product $\mathsf{Int} \times \mathsf{Char}$.

Proposition

Let A and B be finite sets. Then $|A \times B| = |A| \times |B|$.

Proof Suppose that A and B are arbitrary sets with $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$. Draw a table with m rows and n columns of the members of $A \times B$:

$$(a_1, b_1) \quad (a_1, b_2) \quad \ldots$$

$$(a_2, b_1) \quad (a_2, b_2) \quad \ldots$$

$$\ldots$$

Such a table has $m \times n$ entries.

You do not need to remember this proof.

n-ary product

For any $n \geq 1$, an **n-tuple** is a sequence $(a_1, \ldots, a_n)$ of n objects where the order of the a_i matter.

Definition Let $A_1, \ldots, A_n$ be arbitrary sets. The **n-ary product** of the A_i, written $A_1 \times \ldots \times A_n$ or $\bigcup_{i=1}^{n} A_i$, is $\{(a_1, \ldots, a_n) : a_i \in A_i \text{ for } 1 \leq i \leq n\}$.

The n-ary product of As is written A^n, with A^2 corresponding to the Cartesian product.

Examples

1. The three dimensional space of real numbers $\mathcal{R}^3$.
2. The set timetable = day $\times$ time $\times$ room $\times$ courseno.

   A typical element is (Wednesday, 11.00, 308, 140).

   In Haskell notation, this timetable example can be given by:

```
type Day = String
type Time = (Int, Int)
...
type Timetable  = (Day, Time, Room, CourseNo)
(Wednesday, (11,00), 308, 140) :: Timetable
```

3. **Record types** are similar to n-ary products:

```
Person = RECORD
    who : Name;
    height : Real;
    age : [0...120];
    eyeColour : Colour;
    dateOfBirth : Date
END
```

Just like products, records can be nested:

```
Date = RECORD
    day : [1...31];
    month : [1...12];
    year : [1900...1990]    END
```

This record is like a Haskell type augmented with *projector* functions:

```
type Name = String
type Colour = String
type Date = (Int, Int)
type Person = (Name, Float, Int, Colour, Date)

who :: Person -> Name
height :: Person -> Float
age :: Person -> Int
eyeColour :: Person -> Colour
dateOfBirth :: Person -> Date

height (_, h, _, _, _) = h   ...
```

Proposition Let A_i be finite sets for each $1 \leq i \leq n$. Then $|A_1 \times \ldots \times A_n| = |A_1| \times \ldots \times |A_n|$.

This fact can be simply proved by the induction principle introduced next term.

Future

We can form the product of three sets in three different ways:

$$A \times B \times C \qquad (A \times B) \times C \qquad A \times (B \times C)$$

There is a natural correspondence between these three sets.

We will make this intuition precise later in the course.