# MAS164/CSC163. Discrete mathematics for computer science

Sarah Rees

Recommended books:

**J.Gersting** Mathematical structures for computer science, 4th edition (Freeman, 1999).

**R.P.Grimaldi** Discrete and combinatorial mathematics, 4th edition (Addison-Wesley, 1999).

**J.K.Truss** Discrete mathematics for computer scientists, 2nd edition (Addison-Wesley, 1999).

Syllabus as on the web:

Propositional calculus: propositional connectives, formulae, truth tables, tautology; applications in logic design and reasoning about sequential programs. Set theory: defining sets by listing, by properties and by induction; unions, intersections, complements; Venn diagrams; power sets; duality; applications in databases. Building more useful structures: cartesian products; inductive definition of structures; reasoning about structures using structural and mathematical induction; applications in system software. Functions: domain, codomain, range; injections, surjections, bijections; inverse functions; total and partial functions; recursive functions; applications to program design, dealing with recursion and looping structures. Predicate logic: existential and universal quanitifiers; predicate calculus. Boolean algebra: axioms, examples; duality; simplification of Boolean expressions. Graphs: directed and undirected graphs; connectivity; trees, minimal spanning trees; parse trees.

I've inserted a tiny bit about relations, and am working from Fenton and Hill: Systems construction and analysis, McGraw-Hill, 1993, which is probably a bit hard for the students to refer to themselves. There are some changes in notation at the request of CS (e.g. $\iff$ rather that $\leftrightarrow$). I'm changing the order of presentation from that on the web page. I'm not doing <u>minimal</u> spanning trees, but only spanning trees (otherwise we'll have to talk about assigning weights to edges).

# 1 Sets and relations

Based on Fenton and Hill, sections 2.1 and 2.2.

**Definition 1.1** *A set is a collection of objects (known as the elements) for which membership is decidable.*

The definition is a bit mysterious, e.g. what do we mean by the phrase 'for which membership is decidable'?

Basically a set is a collection of things. It might be described explicitly, by listing its elements, but it might also be described in some more general way.

Examples:-

- Sets described by listing their elements e.g. $\{1, 3, 5, 7, 9\}$, also describable as 'the odd digits' and lots more (mixing types too). Use of $\{$ and $\}$, and of $\in$. Each element is counted once only, and the order the elements are listed in is irrelevant. Two sets are equal if they contain the same elements.

- Some standard sets, such as integers, rationals, natural numbers, reals, and the empty set $\emptyset$. In this course, we shall define 0 to be a natural number, in order to be consistent with the IFAD toolkit. In reality, some people think that 1 is the first natural number and some that 0 is; it's a bit confusing... We'll use the notation $\mathbb{Z}^+$ if we want to exclude 0, and call this set the positive integers.

- Sets described by rules, such as $\{x \in \mathbb{Z} : x > 10\}$, where : or $|$ as 'such that'.

- Sets described inductively/recursively (more on this later) e.g. '$S$ contains 1, and if $x \in S$ then $2x \in S$, but nothing else is in $S$' defines the set of positive integers which are powers of 2.

  '$S$ contains 0, and if $x \in S$ then $x + 1 \in S$, but nothing else is in $S$' defines the set of natural numbers.

  Similarly

  '$S$ contains 1, and if $x \in S$ then $x + 1 \in S$, but nothing else is in $S$' defines the set of positive integers.

  And we can even use

  '$S$ contains 10, and if $x \in S$ then $x + 1 \in S$, but nothing else is in $S$' to define the set $\{x \in \mathbb{Z} : x \geq 10\}$.

  Or trees, linked list structures.

Often we use sets in computing to define a <u>type</u> of variables.

How could membership not be decidable?

Consider $\mathcal{A}$, the collection of all sets $A$ such that $A \notin A$. Is $\mathcal{A}$ an element of $\mathcal{A}$? In fact it turns out that we can't decide. So actually, $\mathcal{A}$ isn't really a set. This strange situation is known as Russell's paradox.

**Definition 1.2** *The size (cardinality) of a set. Notation* $|A|$.

Some examples - note that some sets are infinite.

**Definition 1.3** *Define* $\subset$ *and* $\subseteq$, $\cap$ *and* $\cup$. *Also* $A \backslash B$, *and symmetric difference* $A+B$, *concept of a universal set* $\mathcal{U}$ *to define the working environment (the 'type') and of* $A^c$ *as* $\mathcal{U} \setminus A$.

Using these operations we can make new sets out of old ones.

Examples.

Compute a few which give the same answer, e.g. $A \cap (B \cup C)$ and its expansion.

**1.4** *A list of useful rules which tell us how sets combine using these operations. Suppose that* $\mathcal{U}$ *is a universal set here.*

1. Commutative laws.

$$A \cap B = B \cap A \quad A \cup B = B \cup A$$

2. Associative laws.

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

3. Distributive laws.

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

4.

$$A \cap \mathcal{U} = A \quad A \cup \emptyset = A$$

5.

$$A \cap A^c = \emptyset, \quad A \cup A^c = \mathcal{U}$$

6. De Morgan's laws.

$$(A \cap B)^c = A^c \cup B^c \quad (A \cup B)^c = A^c \cap B^c$$

7. Idempotent laws.

$$A \cap A = A \quad A \cup A = A$$

Note the rules generally come in pairs. Given one rule we get a second usually distinct rule by swapping $\cap$ with $\cup$ and $\emptyset$ with $\mathcal{U}$. The principal of duality makes this work.

Some more examples. Duality works for these too. A self-dual rule?

A more formal way to verify rules like these:-

**Definition 1.5** *Venn diagrams*

More ways to make new sets out of old sets:-

**Definition 1.6** *Power set of A, $\mathcal{P}(A)$, or (sometimes) $2^A$. Where A has n elements, its power set has $2^n$.*

**Definition 1.7** *Direct product*

Examples.

The sets $A_1, \ldots A_n$ are the components of the cartesian product $A_1 \times \ldots A_n$.

Order of composition is important.

In CS a cartesian product is often called a record, or a structure. We usually access the components of a record using dot notation.

Size of $A_1 \times A_2 \times \ldots A_n$.

Using IFAD to work examples in set theory

The IFAD toolbox is used throughout the CS degree, so it is worthwhile for us to famililarise ourselves with it. It allows us to do elementary operations on sets. You'll be able to check many of your homework exercises on it before you submit them!

The IFAD notation is sometimes a little different from the standard mathematical notation, simply because only ascii characters are available to it. Sometimes it is a little more restrictive too.

The empty set is represented as $\{\}$, but not as $\emptyset$. $\cup$ is represented by union and $\cap$ by inter.
is used for $\setminus$.

Some examples:

```
print {1,2,3} union { 2,4}
```

will return the union of $\{1,2,3\}$ and $\{2,4\}$.

So will the sequence

```
A: set of nat;
A={1,2,3};
B: set of nat;
B= {3,2,4}
```

in a file, (the ;'s are necessary to end statements when they are input from a file, whereas a return is all that is needed in interactive mode.) followed by the statement

```
print A union B
```

IFAD only allows us to work with finite sets. And it's a little restrictive on sets which are defined by rules. If we want to describe the set

$$\{x \in \{1,2,3,4,5,6,7,8,9,10\} : x^2 < 10\}$$

we have to re-express it as

$$\{x : x \in \{1,2,3,4,5,6,7,8,9,10\} \quad \text{and} \quad x^2 < 10\}$$

In IFAD we write

```
{ x | x in set {1,2,3,4,5,6,7,8,9,10 } & x**2 < 10 }
```

(I think, check the exponent notation) More details can be found in 'Modelling systems', by John Fitzgerald and Peter Gorm Larsen, which is available in the library.

**Definition 1.8** *A relation $R$ is defined to be a subset of a cartesian product. Where the relation is a subset of $A_1 \times \ldots \times A_n$, $R$ is said to be an n-ary relation relating $A_1, A_2 \ldots A_n$. If $A_1 = A_2 = \ldots A_n = A$, we'll say that $R$ is a relation 'on $A$'. Where $R$ is an $n$-ary relation and $(x_1, \ldots x_n) \in R$, we say that $R$ holds at $(x_1, \ldots x_n)$. Sometimes we write $R(x_1, \ldots x_n)$ rather than $(x_1, \ldots x_n) \in R$.*

*When $n = 1$, $R$ is called a unary relation. When $n = 2$, $R$ is called a binary relation. When $n = 3$, $R$ is called a ternary relation.*

In this course most of the relations we shall meet will be either binary or unary. A few will be ternary.

We also use the terms property and predicate to mean exactly the same thing as 'relation'. In particular we usually use the term 'property' for a unary relation, and we'll meet the term 'predicate' later when we do some logic.

The notation we've introduced, $R(x_1, \ldots, x_n)$, rather than $(x_1, \ldots x_n) \in R$, and the use of the term 'property' above gives us a clue as to how we really think of relations. Officially a relation may be a subset of a cartesian product, but *really* we think of an $n$-ary relation as a rule which may or may not hold between some (ordered) sets of $n$ variables.

Some examples of relations:-

**Some unary relations** :

On $\mathbb{Z}$, '$x$ is even', '$x$ is positive'. We can write the statement '$x$ is even' in set theory language, as $x \in EvenInt$ or in the form $EvenInt(x)$. Similarly we can write the statement $x$ is postive as $x \in PosInt$ or as $PosInt(x)$.

**Some binary relations** :

On $\mathbb{Z}$, we have '$x + y$ is even', $x$ is greater than $y$. We can write the statement '$x + y$ is even' in set theory language as $(x, y) \in EvenSum$ or $EvenSum(x, y)$.

Where $\mathcal{U}$ is any set, the relation '$x$ is an element of $A$', which we might write either in the form $(x, A) \in ElementInSet$ or $ElementInSet(x, A)$, relates $\mathcal{U}$ and $\mathcal{P}(\mathcal{U})$.

**A ternary relation** :

On $\mathbb{Z}$, $SumEquals(x, y, z)$ to mean $x + y = z$.

### 1.9 *Binary relations*

Binary relations are particularly useful, and since a binary relation relates two variables, it is often pleasant to use a different kind of notation to denote them (infix notation). Gievn a binary relation $R$, rather than write $(a,b) \in R$, or $R(a,b)$ (prefix notation), we often write $aRb$. Actually, the most common binary relations don't have alphabet characters as their most natural labels.

$= . <, \leq, =$ on $\mathbb{Z}, \mathbb{Q}, \mathbb{R}$. We write $x < y$ in preference to $LessThan(x,y)$,

$\subseteq$ on the set of subsets of some universal set $\mathcal{U}$, i.e. on $\mathcal{P}(\mathcal{U})$.

$\in$ relating $\mathcal{U}$ and $\mathcal{P}(\mathcal{U})$.

### 1.10 *Diagram of a relation*

We also use diagrams to illustrate binary relations, points with directed arcs joining them. These are special cases of <u>directed graphs</u>, which we'll meet later.