06-21254 *Mathematical Techniques for Computer Science*
Autumn Semester 2011

The University of Birmingham
School of Computer Science
November 2, 2011

Handout 9
Sets

54. The intuitive idea of a set. A **set** is a collection of things. The "things" can be concrete (such as "people") or abstract (such as "colours"). No reason for collecting the things into a set needs to be given but often there is a shared attribute. Examples:

- the set of undergraduate students at the University of Birmingham in the academic year 2007/08 (there are about 15,000);
- the seasons (there are four);
- the set of numbers that can be stored exactly as a `float` in Java (there are slightly fewer than 2^{32} many).

The "things" in the set are called the **elements** or the **members**. The following are important aspects of the idea of a set:

1. The elements of a set are identifiable and can be distinguished from each other.
   (So the "waves on the Atlantic" do not form a set.)
2. There is a clear criterion that defines when a "thing" is a member of the set and when it is not.
   (So we can not form the set of "tall people in Britain.")
3. A member of a set is counted only once.
   (So in the "set of staff members who serve on Staff-Student Committee or Teaching Committee" contains Mark Lee only once, although he satisfies both criteria.)
4. As far as the set is concerned, its elements are not ordered in any way.
   (So although it may appear natural to list the elements of the "set of Harry Potter books" in the order
   - Harry Potter and the Philosopher's Stone
   - Harry Potter and the Chamber of Secrets
   - Harry Potter and the Prisoner of Azkaban
   - Harry Potter and the Goblet of Fire
   - Harry Potter and the Order of the Phoenix
   - Harry Potter and the Half-Blood Prince
   - Harry Potter and the Deathly Hallows

   we are not required to do so; listing them in any other order would define the same set.)
5. The set is defined by which members it has and nothing else. If two collections have the same elements, then they form the same set, independently of how the collections were defined.
   (So the "set of triangles which have three equal sides" is the same as the "set of triangles which have three equal angles.")

55. Notation. To indicate that x is an element of the set A, one writes

$$x \in A$$

In mathematics the symbol "$\in$" is used for this purpose and this purpose alone (it originates from the Greek letter *epsilon*, written "ε"). Principle 5 from the previous item can now be written in the form

67

Sets with few elements can often be written out explicitly, for example, the set of seasons:

$$\{\text{Spring}, \text{Summer}, \text{Autumn}, \text{Winter}\}$$

Infinite sets can be indicated by suggesting a formation principle:

$$\{1, 4, 9, 16, 25, \ldots\}$$

56. Sets in Computer Science. Computer scientists use the language of sets for formal and precise descriptions, just like mathematicians, scientists, and engineers. They even developed it into a full-blown *specification language*, called Z.

Set theory has also influenced the design of systems and programming languages. A particularly striking example are "relational databases," suggested in 1970 by the British computer scientist Edgar F. "Ted" Codd. The principles of set theory, listed in Item 54, all have immediate relevance here, in particular 1, 3, and 4:

- 1: this is related to the issue of "keys" in a database table;
- 3: storing information about the same object more than once is wasteful and in many cases causes real errors;
- 4: giving the database management system the freedom to arrange the entries of a table in any order allows many optimizations in storage, processing, and retrieval.

Sets are also related to the programming language idea of a "type." Indeed, finite sets can be defined directly in C as "enumerated types," for example

```
enum Seasons {Spring, Summer, Autumn, Winter};
```

The corresponding Haskell syntax is

```
data Seasons = Spring | Summer | Autumn | Winter
```

Java has supported enumerated types since version 1.5; the syntax is the same as in C except for the inevitable `public` in front of the declaration. Underneath, however, `enum`'s are actually special `class` definitions and the Java compiler automatically creates several methods for it.

You should also know that Java supports the collection type `Set` (as an `interface`) and various concrete implementations, for example, the very useful `HashSet`.

57. Infinite sets. The prime example is the **set of natural numbers**

$$\mathbb{N} = \{0, 1, 2, 3, 4, \ldots\}$$

This is a valid set, though we can never list all its elements. Checking the first two principles listed in Item 54, we find that nevertheless there is no doubt what the members of $\mathbb{N}$ are, and how to distinguish any two of them.

We also have the **integers**

$$\mathbb{Z} = \{\ldots -3, -2, -1, 0, 1, 2, 3, \ldots\}$$

and the **rational numbers**

$$\mathbb{Q} = \{\frac{p}{q} \;\mid\; p \in \mathbb{Z}, q \in \mathbb{N}, q \neq 0, \gcd(p, q) = 1\}$$

where we meet a new way of describing a set. Read this as "all expressions $\frac{p}{q}$ where p is an integer, q is a natural number different from 0, and the greatest common divisor of p and q is 1."

A set that is yet more difficult to write down is that of the **real numbers**, $\mathbb{R}$. Every element of $\mathbb{R}$ is itself an infinite object, namely, an infinite sequence of digits (containing exactly one decimal point). Both mathematicians and computer scientists have their difficulties with $\mathbb{R}$.

58. New sets from old.

Pairing If A and B are sets (possibly the same one) we can form the set $A \times B$ whose elements are pairs (x, y) where x is an element of A and y is an element of B. If $A = B$ then we abbreviate $A \times A$ to A^2.

Tupling More generally, we can form the set $A_1 \times A_2 \times \ldots \times A_n$ whose elements are "n-tuples." If $A_1 = A_2 = \ldots = A_n$ then we write this also as A^n.

Finite sequences If A is a set then A^* is the set whose elements are finite sequences of elements of A.

In computer science, the set A is often finite and called the "alphabet;" the elements of A^* are then the "words over A." A common letter to use for alphabets is Σ.

One peculiar element of A^* is the "empty word," or the "empty sequence," often denoted by ε (the real Greek letter epsilon).

A^* corresponds to the concept of a "string" or more generally, a "list" in programming languages.

Subsets If we already have a set A then we can select some of its elements and form a new set B, called a **subset** of A. The notation for this is

$$B = \{x \in A \ \mid\ x \text{ satisfies condition } c\}$$

Read this as "B is the set of all elements of A for which the condition c is true." Some examples:

68

When defining a subset in this fashion, it is important that in the first part we state which set we select elements from; this allows us to use negative conditions as well, for example:

$$\{x \in \mathbb{N} \ \mid\ x \text{ is not a perfect square}\} = \{2,3,5,6,7,8,10,\dots\}$$

To indicate that one set is a subset of another, we use a special symbol, reminiscent of "$\leq$"

$$B \subseteq A$$

When things get more complicated then the condition c is best expressed with the help of formal logic. The last example above could be written as

$$\{\frac{p}{q} \ \mid\ \exists n \in \mathbb{N}.\, q = 2^n\}$$

Whatever the set A is that we started from, we can always form the following subsets

$$\{x \in A \ \mid\ \text{true}\}$$
$$\{x \in A \ \mid\ \text{false}\}$$

The first is the same as A, the second is the **empty set**, which has no elements. The empty set is denoted by $\emptyset$ or $\{\}$.

59. Operations on subsets. Suppose B, B_1 and B_2 are subsets of a common set A. We have the following operations:

Union By the "union of B_1 and B_2" we mean taking the elements of B_1 and B_2 together (but not repeating those elements that appear in both). As a formula:

$$B_1 \cup B_2 = \{x \in A \ \mid\ x \in B_1 \text{ or } x \in B_2\}$$

As a **Venn diagram**:

69

Intersection By the "intersection of B_1 and B_2" we mean the set of elements that are common to B_1 and B_2. As a formula:

$$B_1 \cap B_2 = \{x \in A \mid x \in B_1 \text{ and } x \in B_2\}$$

As a Venn diagram:

70

Complement By the "complement of B" we mean those elements of A which do *not* belong to B. As a formula:

$$\overline{B} = \{x \in A \mid x \notin B\}$$

As a Venn diagram:

71

Note that these three set operations correspond precisely to the connectives of propositional logic:

	logic	sets	Java	SQL
or	$\vee$	$\cup$	`||`	`OR`
and	$\wedge$	$\cap$	`&&`	`AND`
not	$\neg$	$\overline{}$	`!`	`NOT`

The two logical constants "true" and "false" give rise to A (as a subset of itself) and $\emptyset$, the empty set, as we have seen before.

The following operation is derived from intersection and complement, but appears quite often in practice and therefore has its own symbol:

Set difference $B_1 \setminus B_2 = B_1 \cap \overline{B_2} = \{x \in A \mid x \in B_1 \text{ but } x \notin B_2\}$

72

60. Power sets. In set theory it is postulated that the subsets of a set A can again be collected together into a set. The result is called the **power set** of A and written $\mathscr{P}A$. It is a useful exercise to check the two conditions for a set given at the beginning of this handout:

1. The elements of a power set are "identifiable" because set theory postulates clearly what it means for two sets to be equal or not equal (the formula in Box 67), and this turns sets into "identifiable things."
2. The criterion for a set B to be a member of $\mathscr{P}A$ is that B is a subset of A. This in turn has a precise definition: every element of B must also be an element of A.

It is also quite important that we have said that the elements of a set count only once, because typically there are many different ways to specify a subset. These different specifications of a subset are not important from the point of view of the power set, only the elements that each subset contains.

61. Practical advice. In the exam I expect you to

- know the set-theoretic symbols $\in$, $\subseteq$, $\emptyset$, $\cup$, $\cap$, $\overline{\ \ }$, and $\setminus$;
- know the traditional symbol for sets of numbers $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$;
- know the symbol for the set of strings Σ^* over an alphabet Σ;
- read subset definitions written in logic;
- write subset definitions in logic;
- read and evaluate nested expressions that use $\cup$, $\cap$, $\overline{\ \ }$, and $\setminus$;
- draw Venn diagrams of such expressions.

Notes

- To show that $A \subseteq B$, prove that every element of A belongs to B, too.
- To show that $A = B$, prove $A \subseteq B$ and $B \subseteq A$.