Foundations of Computer Science Comp109

University of Liverpool

Boris Konev

konev@liverpool.ac.uk

Olga Anosova

O.Anosova@liverpool.ac.uk

Last lecture recap: Induction.

Strong induction:

-
- **then** it holds for $n = m + 1$.

DIY Example from last lecture: Number of multiplications

For any integer $n \geq 1$, if x_1, x_2,..., x_n are n numbers, then no matter how the parentheses are inserted into their product, the number of multiplications used to compute the product is $n - 1$.

Proof continued

Part 2. (Naive) Set Theory

Comp109 Foundations of Computer Science

Reading

- S. Epp. *Discrete Mathematics with Applications* Chapter 6
- K. H. Rosen. *Discrete Mathematics and Its Applications* Chapter 2

Contents of the Set theory topic

- Notation for *sets*.
- Important sets.
- What is a *subset* of a set?
- When are two sets *equal*?
- *Operations* on sets.
- *Algebra* of sets.
- Bit strings.
- *Cardinality* of sets.
- Russell's paradox.

Notation

A *set* is a collection of objects, called the *elements* of the set. For example:

- $\{7, 5, 3\}$;
- $\{\text{Liverpool}, \text{Manchester}, \text{Leeds}\}$.

We have written down the elements of each set and contained them between the *braces* $\{\ \}$.

We write $a \in A$ to denote that the object a is an element of the set A:

$$7 \in \{7, 5, 3\}, \;\; 4 \notin \{7, 5, 3\}.$$

Order and repetitions in a set

- The order of elements does not matter
- Repetitions do not count

Predicate notation

For a large set, especially an infinite set, we cannot write down all the elements. We use a predicate P instead.

$$A = \{x \in S \mid P(x)\}$$

denotes the set of objects x from S for which the predicate $P(x)$ is true.

Examples: Let $A = \{1, 3, 5, 7, \ldots\}$. Then

$$A =$$

Very informal notation:

$$A = \{2n - 1 \mid \ n \text{ is a positive integer } \} =$$

Examples

Find descriptions of the following sets by listing their elements:

- $A = \{x \mid \; x \text{ is a day of the week not containing "}u\text{" }\}$
- $B = \{n^2 \mid \; n \text{ is an integer }\} = \{k \in \mathbb{Z} \mid \exists n \in \mathbb{Z} \text{ such that } n^2 = k\}$
- $C = \{x \in \mathbb{Z} \mid \; x^2 + 4x = 12\}$

Important sets (notation)

The empty set has no elements. It is written as $\emptyset$, $\varnothing$ or as $\{\}$.

We have seen some other examples of sets in Part 1.

- $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ (the natural numbers)
- $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ (the integers)
- $\mathbb{Z}^+ = \{1, 2, 3, \ldots\}$ (the positive integers)
- $\mathbb{Q} = \{x/y \mid x \in \mathbb{Z}, y \in \mathbb{Z}, y \neq 0\}$ (the rationals)
- $\mathbb{R}$: (real numbers, used to measure continuous quantities)
  - $[a, b] = \{x \in \mathbb{R} \mid a \leq x \leq b\}$ the set of real numbers between a and b (inclusive)

Computing: Sets in programming languages

Sets are the 'most elementary' data structures: they don't always map well into the underlying hardware, but can speed up some procedures (can you guess which ones?).

Some modern programming languages feature sets.

- For example, in Python one writes

```
empty = set()
m = {'a', 'b', 'c'}
n = {1, 2}
print('a' in m)
```

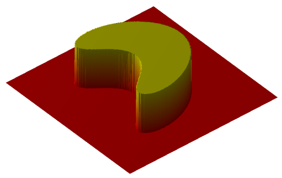
Ordered sequences

Only finite sets can be represented.

- Another common data type: List *Java&Python do differently*

- If all elements of A are drawn from some ordered sequence $S = \langle s_1, \ldots, s_n \rangle$: the characteristic vector (indicator vector, dummy variable, one-hot-encoding) of A is the sequence $[b_1, \ldots, b_n]$ where

$$b_i = \begin{cases} 1 & \text{if} \quad s_i \in A \\ 0 & \text{if} \quad s_i \notin A \end{cases}$$



Indicator function from here

Sequences of zeros and ones of length n are called bit strings of length n (*bit vectors*, *bit arrays*).

Example

Let $S = \langle 1, 2, 3, 4, 5 \rangle$, $A = \{1, 3, 5\}$ and $B = \{3, 4\}$.

- The characteristic vector of A is $\chi_A =$
- The characteristic vector of B is $\chi_B =$
- The set characterised by $[1, 1, 1, 0, 1]$ is
- The set characterised by $[1, 1, 1, 1, 1]$ is
- The set characterised by $[0, 0, 0, 0, 0]$ is

Subsets

Definition A set B is called a subset of a set A if every element of B is an element of A. This is denoted by $B \subseteq A$.

Examples:

$$\{3,4,5\} \subseteq \{1,5,4,2,1,3\},\ \{3,3,5\} \subseteq \{3,5\},\ \{5,3\} \subseteq \{3,5\}.$$

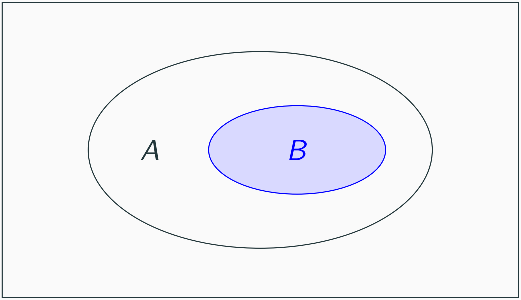


Figure 1: Venn diagram of $B \subseteq A$.

Empty set

Does an empty set $\emptyset$ belong to the set $A = \{1, 2, 3\}$?

Subsets in Python

```
def isSubset(A, B):
    for x in A:
        if x not in B:
            return False
    return True
```

Testing the method:

```
m = {'a', 'b', 'c'}
n = {1, 2}


print(isSubset(n,m))
```

But then there is a built-in operation:

```
print(n<m)
```

Subsets and bit vectors

Let $S = \langle 1, 2, 3, 4, 5 \rangle$, $A = \{1, 3, 5\}$ and $B = \{3, 4\}$.

- Is $A \subseteq B$?

- Is the set C, represented by $[1, 0, 0, 0, 1]$, a subset of the set D, represented by $[1, 1, 0, 0, 1]$?

Equality

Definition A set A is called equal to a set B if $A \subseteq B$ and $B \subseteq A$. This is denoted by $A = B$.

Examples:

$$\{1\} = \{1, 1, 1\},$$

$$\{1, 2\} = \{2, 1\},$$

$$\{5, 4, 4, 3, 5\} = \{3, 4, 5\}.$$

Equality and bit vectors

Let $S = \langle 1, 2, 3, 4, 5 \rangle$, $A = \{1, 3, 5\}$ and $B = \{3, 4\}$.

- Is $A = B$?

- Is the set C, represented by $[1, 0, 0, 0, 1]$, equal to the set D, represented by $[1, 1, 0, 0, 1]$?

Summary

- A set is
- When dealing with some ordered sequence, a characteristic vector (indicator vector, dummy variable, one-hot-encoding)
- Sets A and B are equal iff
- $\emptyset \subseteq A$ for