Discrete Mathematics in Computer Science

B8. Cantor's Theorem

Malte Helmert, Gabriele Röger

University of Basel

October 30, 2024

Discrete Mathematics in Computer Science

October 30, 2024 — B8. Cantor's Theorem

Reminder: Cardinality of the Power Set

Theorem

Let S be a finite set. Then $|\mathcal{P}(S)| = 2^{|S|}$.

B8.1 Cantor's Theorem

Countable Sets

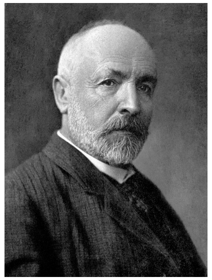
We already know:

- Sets with the same cardinality as $\mathbb{N}_0$ are called countably infinite.
- A countable set is finite or countably infinite.
- Every subset of a countable set is countable.
- The union of countably many countable sets is countable.

Open questions (to be resolved today):

- Do all infinite sets have the same cardinality?
- Does the power set of an infinite set S have the same cardinality as S?

Georg Cantor



- German mathematician (1845–1918)
- Proved that the rational numbers are countable.
- Proved that the real numbers are not countable.
- Cantor's Theorem: For every set S it holds that $|S| < |\mathcal{P}(S)|$.

Our Plan

- Understand Cantor's theorem
- Understand an important theoretical implication for computer science

Cantor's Diagonal Argument Illustrated on a Finite Set

$S = \{a, b, c\}$.

Consider an arbitrary function from S to $\mathcal{P}(S)$.
For example:

	a	b	c	
a	1	0	1	a mapped to $\{a, c\}$
b	1	1	0	b mapped to $\{a, b\}$
c	0	1	0	c mapped to $\{b\}$
	0	0	1	nothing was mapped to $\{c\}$.

We can identify an "unused" element of $\mathcal{P}(S)$.
Complement the entries on the main diagonal.

Works with every function from S to $\mathcal{P}(S)$.
$\rightarrow$ there cannot be a surjective function from S to $\mathcal{P}(S)$.
$\rightarrow$ there cannot be a bijection from S to $\mathcal{P}(S)$.

Cantor's Diagonal Argument on a Countably Infinite Set

$S = \mathbb{N}_0$.

Consider an arbitrary function from $\mathbb{N}_0$ to $\mathcal{P}(\mathbb{N}_0)$.
For example:

	0	1	2	3	4	...
0	1	0	1	0	1	...
1	1	1	0	1	0	...
2	0	1	0	1	0	...
3	1	1	0	0	0	...
4	1	1	0	1	1	...
⋮	⋮	⋮	⋮	⋮	⋮	⋱
	0	0	1	1	0	...

Complementing the entries on the main diagonal again results in an "unused" element of $\mathcal{P}(\mathbb{N}_0)$.

Cantor's Theorem

Theorem (Cantor's Theorem)

For every set S it holds that $|S| < |\mathcal{P}(S)|$.

Proof.

Consider an arbitrary set S. We need to show that

1. There is an injective function from S to $\mathcal{P}(S)$.
2. There is no bijection from S to $\mathcal{P}(S)$.

For 1, consider function $f : S \to \mathcal{P}(S)$ with $f(x) = \{x\}$.
It maps distinct elements of S to distinct elements of $\mathcal{P}(S)$. ...

Cantor's Theorem

Proof (continued).

We show 2 by contradiction.
Assume there is a bijection f from S to $\mathcal{P}(S)$.

Consider $M = \{x \mid x \in S, x \notin f(x)\}$ and note that $M \in \mathcal{P}(S)$.

Since f is bijective, it is surjective and there is an $x \in S$ with $f(x) = M$. Consider this x in a case distinction:

If $x \in M$ then $x \notin f(x)$ by the definition of M. Since $f(x) = M$ this implies $x \notin M$. $\rightsquigarrow$ contradiction

If $x \notin M$, we conclude from $f(x) = M$ that $x \notin f(x)$. Using the definition of M we get that $x \in M$. $\rightsquigarrow$ contradiction

Since all cases lead to a contradiction, there is no such x and thus f is not surjective and consequently not a bijection.

The assumption was false and we conclude that there is no bijection from S to $\mathcal{P}(S)$. □

B8.2 Consequences of Cantor's Theorem

Infinite Sets can Have Different Cardinalities

There are infinitely many different cardinalities of infinite sets:

- $|\mathbb{N}_0| < |\mathcal{P}(\mathbb{N}_0)| < |\mathcal{P}(\mathcal{P}(\mathbb{N}_0))| < \dots$
- $|\mathbb{N}_0| = \aleph_0 = \beth_0$
- $|\mathcal{P}(\mathbb{N}_0)| = \beth_1 (= |\mathbb{R}|)$
- $|\mathcal{P}(\mathcal{P}(\mathbb{N}_0))| = \beth_2$
- $\dots$

Existence of Unsolvable Problems

There are more problems in computer science
than there are programs to solve them.

There are problems that cannot be solved by a computer program!

Why can we say so?

Decision Problems

"Intuitive Definition:" Decision Problem

A decision problem is a Yes-No question of the form
"Does the given input have a certain property?"

- "Does the given binary tree have more than three leaves?"
- "Is the given integer odd?"
- "Given a train schedule, is there a connection from Basel to Belinzona that takes at most 2.5 hours?"

- Input can be encoded as some finite string.
- Problem can also be represented as the (possibly infinite) set of all input strings where the answer is "yes".
- A computer program solves a decision problem if it terminates on every input and returns the correct answer.

More Problems than Programs I

- A computer program is given by a finite string.
- A decision problem corresponds to a set of strings.

More Problems than Programs II

- Consider an arbitrary finite set of symbols (an alphabet) Σ.
- You can think of $\Sigma = \{0, 1\}$ as internally computers operate on binary representation.
- Let S be the set of all finite strings made from symbols in Σ.
- There are at most $|S|$ computer programs with this alphabet.
- There are at least $|\mathcal{P}(S)|$ problems with this alphabet.
  - every subset of S corresponds to a separate decision problem
- By Cantor's theorem $|S| < |\mathcal{P}(S)|$, so there are more problems than programs.

B8.3 Sets: Summary

Summary

- Cantor's theorem: For all sets S it holds that $|S| < |\mathcal{P}(S)|$.
- There are problems that cannot be solved by a computer program.

B8.4 Outlook: Finite Sets and Computer Science

Enumerating all Subsets

Determine a one-to-one mapping between numbers $0, \dots, 2^{|S|} - 1$ and all subsets of finite set S:

- Consider the binary representation of numbers $0, \dots, 2^{|S|} - 1$.
- Associate every bit with a different element of S.
- Every number is mapped to the set that contains exactly the elements associated with the 1-bits.

$S = \{a, b, c\}$

decimal	binary abc	set
0	000	$\{\}$
1	001	$\{c\}$
2	010	$\{b\}$
3	011	$\{b, c\}$
4	100	$\{a\}$
5	101	$\{a, c\}$
6	110	$\{a, b\}$
7	111	$\{a, b, c\}$

Computer Representation as Bit String

Same representation as in enumeration of all subsets:

- Required: Fixed universe U of possible elements
- Represent sets as bitstrings of length $|U|$
- Associate every bit with one object from the universe
- Each bit is 1 iff the corresponding object is in the set

Example:

- $U = \{o_0, \dots, o_9\}$
- Associate the i-th bit (0-indexed, from left to right) with o_i
- $\{o_2, o_4, o_5, o_9\}$ is represented as: 0010110001

How can the set operations be implemented?