CS 2742 (Logic in Computer Science) – Winter 2014 Lecture 18

Antonina Kolokolova

March 7, 2014

5.1 Cardinalities and diagonalization

It is easy to compare sizes (in set theory terminology, cardinalities) when sets are finite: whichever set has more elements, that set is larger. In a way, what we are doing in that comparison is creating a correspondence between the elements of the sets by matching elements of one set to elements of another one-by-one. If this correspondence is both one-to-one and onto, then our sets are of equal size. For example, to match $\{10, 20, 30\}$ with $\{a, b, c\}$ we can define a bijection $f(x)$ by $f(10) = a, f(20) = b, f(30) = c$. If sets have different sizes, then either f cannot be one-to-one (if the second set is smaller) or it is not onto (if the second set is large).

This is the idea behind the comparison of infinite sets. We declare two sets to have the same size (cardinality), if there is bijection (that is, a function which is both one-to-one and onto) from one set to the other. To extend this definition, we say that a (possibly infinite) set A has cardinality $|A| \leq |B|$ if there is a one-to-one function from A to B. Note that this gives us another way of proving that two sets have equal cardinality: give two one-to-one functions, one from A to B and another from B to A. Rephrasing pigeonhole principle: for sets such that $|A| < |B|$ there is no one-to-one function from B to A (no onto function from A to B).

But this definition gives us some strange consequences. With infinite sets, it is possible that a proper subset has the same cardinality as the whole set (although it is not possible for a subset to be larger than the whole: identity function gives the proof). Example: $\mathbb{N} \subset \mathbb{Q}$, $Even \subset \mathbb{N}$, $\mathbb{N} \subset \mathbb{N} \times \mathbb{N}$.

Definition 1. *We call sets that have the same cardinality as* $\mathbb{N}$ countable *sets. The sets that have larger cardinality such as* $\mathbb{R}$*, we call* uncountable*.*

Example 1. The set of positive rational numbers is countable.

Note that we already know an injective function from $\mathbb{N}$ to $\mathbb{Q}^+$: it is an identity function. So the interesting part is to give an injective function from $\mathbb{Q}^+$ to $\mathbb{N}$.

For that, look at the following table:

1/1	1/2	1/3	1/4	1/5	...
2/1	2/2	2/3	2/4	2/5	...
3/1	3/2	3/3	3/4	3/5	...
4/1	4/2	4/3	4/4	4/5	...
5/1	5/2	5/3	5/4	5/5	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Now, start counting the elements of the table as follows:

1: 1/1, 2: 1/2. 3: 2/1, 4: 1/3, 5: 2/2, 6: 3/1, 7: 1/4 and so on (that is, list all elements with the sum of numerator and denominator equal 2, then all with sum =3 and so on). Note that every rational number is represented in this table, multiple times, too (for example, 2/4 and 1/2). So we just constructed a one-to-one function from a set larger than $\mathbb{Q}^+$ to the set of natural numbers. Therefore, the cardinality of $\mathbb{Q}^+$ is at most the cardinality of $\mathbb{N}$. Since it is also at least $|\mathbb{N}|$, we conclude that $|\mathbb{Q}^+| = |\mathbb{N}|$.

This kind of argument is often used to show that a certain set is countable.

5.2 Diagonalization

At this point you may ask – is it true, then, that all infinite sets have the same size? What about a powerset of natural numbers $2^{\mathbb{N}}$? In this case, we can show that the resulting set is actually strictly larger than $\mathbb{N}$. For this, we will use a technique called *diagonalization*, due to Cantor, who used it to show that the set of real numbers is larger than the set of natural numbers.

The idea of the proof is as follows. The proof proceeds by contradiction. Assume, for the sake of contradiction, that your set in question (i.e., $2^{\mathbb{N}}$ or $\mathbb{R}$) is countable, that is, there is function from the elements of this set to natural numbers. Make a table in which rows correspond to elements being enumerated in the order of that assumed enumeration (e.g, rows are real numbers represented as strings of digits or rows are (infinite) strings encoding the powerset of natural numbers). Now, construct a new element which belongs to the set and which is different from any row in the table.

For example, here is how the table looks for proving that the powerset of natural numbers is uncountable. Represent each subset S of natural numbers by an infinite string s with 0 in bit i when number i is not in the set S and 1 in bit i if $i \in S$. Now, list these strings as rows of the table, according to that alleged enumeration. The table would look similar to this:

1	**0**	0	1	1	0	1	1	0	1	....
2	1	**1**	1	1	1	0	0	1	1	....
3	1	0	**0**	0	0	1	1	1	1	....
4	1	1	0	**1**	1	0	0	1	1	....
5	0	0	1	1	**1**	1	1	0	0	....
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
-	**1**	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**1**	....

Here, inside of the table is the alleged listing of all possible subsets of natural numbers, represented as strings. The string under the table differs from the first line in the table in the first bit, from the second line of the table in the second bit and so on. Since this string looks like the diagonal of the table inverted, the method is called diagonalization.

How do we show that this diagonal string that we constructed is not in the listing? Suppose it is, then it is in the table as a row number k for some k. But this is not possible, because our diagonal string differs from the string in row k in its k^{th} bit. Therefore, this string is not in the enumeration. Since this works for any possible enumeration of subsets of natural numbers (just construct a corresponding string for each enumeration), we conclude that $2^{\mathbb{N}}$ is not countable. However, it has the same size as the set of all real numbers (exercise: see how you can prove that). It makes sense to ask is there a set with cardinality strictly between the two, that is, some set A such that $|\mathbb{N}| < |A| < |\mathbb{R}|$. And the answer is... not only we don't know, but either way would not contradict the axioms of mathematics.

This is called the *Continuum Hypothesis*. It says that there is no set whose size is strictly between that of natural numbers and that of real numbers (that is, between $\mathbb{N}$ and $2^{\mathbb{N}}$. This hypothesis is not provable (or disprovable) from the current axioms of mathematics (of Zermelo-Fraenkel set theory ZFC).

Another interesting application of this method is to show that there are problems that cannot be solved by any (say, Java) program. Think about the simplest kind of problems, classification of inputs – or even simpler, determining whether a given input is a number belonging to a specified set. For example, such membership problem can be determining, given a natural number, whether that number is prime or whether it is a square of another number. As you are already noticing from the definition of the problem, here a "problem" (often called "language" in this setting) corresponds exactly to a subset of natural numbers discussed above. So as before the rows of our table will encode subsets of natural numbers, with each column corresponding to a number that can be given to a program as an input.

Now, we want to enumerate the rows of the table by Java programs, where a Java program associated with a row should correctly compute all values in this row (for example, if the row corresponds to 1s for all prime numbers, the Java program should correctly determine whether a given number is prime). Note that any Java program can be written as a finite binary string, and we already said that finite binary strings (with 1 in front) form a bijection with natural numbers. In particular, there are not more Java programs than natural numbers. Now, applying diagonalization exactly the same way as we did above, we obtain a language that differs from the language computed by i^{th} Java program on the i^{th} number. Therefore, this language is not computable.

At this point you may ask if there is a specific, concrete example of a language not computed by any Java program. Indeed there is, and to construct it, we will use intuition akin Russell's paradox $A = \{x \mid x \notin A\}$. This particular example is called the Halting Problem, and it corresponds to a very natural task of checking whether a given program goes into an infinite loop on a given input or eventually stops (halts). More precisely, let CheckHalt be an algorithm such that CheckHalt(M,x) prints "halts" if M terminates on input x, and "loops" if M does not terminate. Let $Diag(X) = \neg CheckHalt(X, X)$. Such a $Diag(X)$ gives a paradox (just think what $Diag$ would do if presented with its own code as an input).