Maggie Johnson Handout #17
CS109

Infinite Sets and Countability

Key topics:

* Infinite Sets
* Diagonalization
* Russell's Paradox
* The Beginnings of Theoretical Computer Science

Infinite Sets

In a **finite** set, we can always designate one element as the first member, another as the second member, etc. If there are k elements in the set, then these can be listed in the order we have selected:

$$s_1, \; s_2, \; \ldots, \; s_k$$

Therefore, a finite set is one which can be enumerated by the positive integers from 1 up to some integer k. More precisely, A is finite if there is a positive integer k such that there is a one-to-one correspondence between A and the set of all natural numbers less than k. We define one-to-one correspondence between the elements of a set P and the elements of a set Q if it is possible to pair off the elements of of P and Q such that every element of P is paired off with a distinct element of Q. We are defining a function that maps the elements of P to the elements of Q; this function must be both one-to-one and onto (or bijective for you function enthusiasts out there).

If a set is infinite, we may still be able to select a first element s_1, and a second element s_2, but we have no limit k. So the list of chosen elements may look like this:

$$s_1, \; s_2, \; s_3 \ldots\ldots$$

Such an infinite set is called **denumerable**. Both finite and denumerable sets are **countable** sets because we can count, or enumerate the elements in the set. Being countable, however, does not always mean that we can give a value for the total number of elements in the set; it just means we can say "Here is a first one, here is a second one.....". In other words, a denumerable set is one where we can define a one-to-one correspondence between the elements in the set and $\mathbb{Z}^+$.

Thus, the set $\mathbb{Z}^+$ is, in a sense, the most basic of all infinite sets. The reason for this is that the one-to-one correspondence can be used to "count" the elements of an infinite set. If f is a one-to-one and onto function from $\mathbb{Z}^+$ to some infinite set A, then f(1) can be designated as the first element of A, f(2), the second, and so forth. Because f is one-to-one, no element is ever counted twice; and because f is onto, every element of A is counted eventually.

To prove denumerability, we need only exhibit a counting scheme, i.e., if starting from a particular element, we can sequentially list all the elements in the list, (because such a listing will yield a one-to-one correspondence between the elements in the set and $\mathbb{Z}^+$). The counting scheme is the function that maps $\mathbb{Z}^+$ to some other infinite set.

Example 1

The set of positive even integers {0, 2, 4, 6, 8...} is denumerable because there is an obvious one-to-one correspondence between these integers and the positive integers (2k). The set of all integers, positive and negative is denumerable because we can list them as follows: {0, 1, -1, 2, -2, 3, -3 ...} (f(n) = n/2 if n is even, -(n-1/2) if n is odd). But these examples seem to go against common sense. The set of positive even integers must be 1/2 the size of the set of positive integers; the set of all integers must be twice the size of the set of positive integers. We have certainly defined functions for these two sets that are both one-to-one and onto, so by the basic definitions given above, $\mathbb{Z}^{+}_{\text{even}}$ and $\mathbb{Z}$ have the same cardinality as $\mathbb{Z}^{+}$. Surprising, but true!

Example 2

The set $\mathbb{Q}^{+}$ (positive rational numbers) is denumerable. We assume that each positive rational number is written as a fraction of positive integers. We can write all such fractions having the numerator 1 in one row, all those having a numerator 2 in the second row, and so on:

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...
…				

To show that the set of all fractions in this table is denumerable, we will thread an arrow through the table beginning with 1/1. Following the arrow gives an enumeration of the set: 1/1, 2/1, 1/2, 1/3, 2/2, 3/1, 4/1... (We have to skip numbers that were already counted so the function will be one-to-one). Therefore, the set represented by the table is denumerable.

Diagonalization

Are all infinite sets countable? To prove there are **uncountable** infinite sets, we need a new proof technique. We introduce this technique with an example: Three boys, Mark, Matt & Max have very particular tastes when it comes to ice cream. Of course, they all like Ben & Jerry's, but only certain flavors as illustrated in the table below:

	Chunky Monkey	Cherry Garcia	NY Super Fudge Chunk
Mark	Y	N	Y
Matt	N	N	Y
Max	Y	Y	Y

We can make a trivial observation from this table: Suppose we are told there is a boy (we don't know if he is one of the three we know or not), who disagrees with Mark about Chunky Monkey (Mark likes it, so this other boy hates it). This other boy also disagrees with Matt about Cherry Garcia (Matt does not like it so this other boy likes it); and finally, this boy disagrees with Max about NY Super Fudge Chunk. Clearly, this other boy must be Mike, a totally different boy, since he disagrees with each of the original three boys on at least one of the flavors.

This little example illustrates a **diagonal** argument in which we assert that a certain object is not one of the given objects, using the fact that this new object is different from each of the given objects in at least one way. This new object *must* be different from the given objects.

As an example of an uncountable infinite set, we will show using **diagonalization**, that the real numbers between 0 and 1 are not countable. This very famous proof was first done by Cantor in the late 19th century. As before, to prove that something is denumerable, we need to show a one-to-one correspondence between the enumeration of the elements of the set and the natural numbers. So, we need a scheme. Any member of the set can be written as a 0 followed by a decimal point and then an infinite series of numbers representing the decimal fraction. If the set of reals between 0 and 1 is countable, we can exhaustively list them one after another in decimal form where r_1, r_2, r_3... have a one-to-one correspondence with the natural numbers:

r_1	0. d_{11} d_{12} d_{13} ....
r_2	0. d_{21} d_{22} d_{23} ....
r_3	0. d_{31} d_{32} d_{33} ....
	.....
r_i	0. d_{i1} d_{i2} d_{i3} ....
	....

where d_{ij} {0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }. For example, if r_1 = 0.23794102... then d_{11} = 2, d_{12} = 3, d_{13} = 7...).

But will such a listing include all the reals between 0 and 1? By using the diagonal of the table, we can always come up with a number that is not in the list. Suppose we construct a new real number (0. p_1 p_2 p_3...) as follows: p_i is always chosen to be 5 if d_{ii} <> 5, and 6 if d_{ii} = 5. So, if the enumeration is as follows:

r_1	0.342134...
r_2	0.257001...
r_3	0.546122...
r_4	0.716525...
.....	

Since d_{11} = 3, then p_1 = 5; d_{22} = 5, so p_2 = 6; d_{33} = 6 so p_3 = 5; d_{44} = 5 so p_4 = 6. So the new number starts as 0.5656.... If we compare this to the enumeration, this new number is definitely between 0 and 1, but it *must* differ from all the members of the enumeration in just the same way that Mike could not be one of the three original boys. The crucial observation is: for each integer i, the new number d differs in the ith position from the ith number in the list. Consequently, we conclude that no matter what enumeration we devise, we can always come up with another number using diagonalization. Therefore, the real numbers between 0 and 1 are not countable.

Russell's Paradox

Cantor's discovery that the set of real numbers between 0 and 1 is uncountable was just the first of many paradoxes in mathematics in the late 19th and early 20th centuries. In 1902, Bertrand Russell, a philosopher and logician, discovered a "paradox" (really a contradiction) of set theory.

Can a set be an element of itself? Most sets are not. For example, the set of all integers is not an integer; the set of all gorillas is not a gorilla. On the other hand, the set of all abstract ideas is itself an abstract idea.

Russell's Paradox: Let A be the collection of all sets which are not elements of themselves (like integers or gorillas). In formal set lingo: A = { S A | S S }. Is A an element of this set, i.e., is

A A? The answer is yes and no. If A is an element of this set, then by the definition of A, it is impossible for A to belong to A. On the other hand, if A is not an element of this set, then A must be an element of A. Neither case can be true which is a contradiction.

Russell's paradox had a profound impact on mathematics for a couple reasons. First, abstract set theory as developed by Cantor was not consistent unless some "workarounds" were provided. Second, even though his contradiction could be made to disappear by more careful definitions, its existence caused people to wonder what other contradictions remained.

There were other unsettling questions that set theory raised: e.g., how can there be different sizes of infinity? All this left a cloud over mathematics that needed to be resolved. David Hilbert, a brilliant mathematician, wanted to put mathematics back on track by putting it on the same footing as Euclidean geometry (which is characterized by precisely specified direct proofs). Hilbert felt that the mathematics that had developed after Euclid did not follow this standard of precision; if mathematics were put back on this standard, the paradoxes would go away. Hilbert wanted to develop a complete set of general axioms that can be applied to any type of mathematical thinking. Then, he wanted to develop complete, guaranteed, easy-to-follow sets of instructions (algorithms) for solving whole classes of mathematical problems. This would put the precision and exactness back in mathematics. *Maybe*, he could even come up with an algorithm to solve all mathematical problems of any kind in a finite number of steps.

Many mathematical logicians tried to develop algorithms using Hilbert's system. Many others tried to prove that Hilbert's system was *complete* and *consistent*.

> complete: a system where any true statement can be proven true (i.e., all required axioms are included with no extras)
> consistent: a system where no false statement can be proven true

But in 1931, Kurt Gödel's **Incompleteness Theorem** proved that no worthwhile mathematical system could be developed that is both complete and consistent.

"This statement is false" is an example of what are called self-referential undecidable statements. Gödel proved that there are statements of this nature in any mathematical system. Thus, since such statements exist in any mathematical system, he proved that a system could not be complete without being inconsistent.

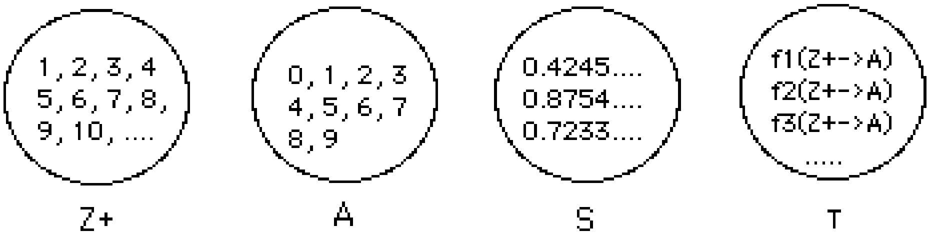
This had serious ramifications in mathematics. It showed that any branch of mathematics has its limitations, i.e., no mathematical system with a finite number of axioms and based on rules of inference, can contain a full depiction of reality.

The Beginnings of Theoretical Computer Science

In computer science, researchers continued working on Hilbert's algorithms despite Gödel and his theorem, and we have the beginning of the type of problem-solving we learn in programming. These researchers learned a lot about the nature of algorithms, and they also were able to begin defining some of the self-referential undecidable questions as they applied to algorithms (rather than mathematical systems). The first definition of an algorithm was proposed by Alonzo Church. Using his definition, he and Stephen Kleene, and independently, Emil Post, were able to prove that there were indeed problems that no algorithm could solve.

Another important researcher in this area was Alan Turing who applied aspects of Gödel's theorem to the idea of algorithms. He was interested in coming up with a "mechanical" method for implementing algorithms. A Turing Machine is a theoretical model of a machine that could implement algorithms. In terms of Gödel's theory, a computer is viewed as having an initial state (registers and data) which represents the definitions and axioms in a mathematical system. The computer than performs operations using the data and registers in a manner analogous to how we use logic to develop new theorems in math. It turns out that this analogy works - there are undecidable problems in computer science.

It's not surprising that there are problems that no algorithm can solve. Think of it in terms of a set of functions: Suppose you have a set T of all the functions that map the positive integers Z^+ to the set A of digits (which equals {0,1,2,3,4,5,6,7,8,9}). Let S be the set of all real numbers between 0 and 1. As noted earlier, any number in S can be represented in the form $0.\ d_1\ d_2\ d_3 \ldots\ d_n$ where each d_i is an integer from 0 to 9 and n is a positive integer. So we have four sets:



Define a function F: S -> T, by the rule: $F(0.\ d_1\ d_2\ d_3 \ldots\ d_n)$ = the subset of functions of T that maps each positive integer i to d_i (i.e., we are mapping the location of the element in the decimal (e.g. the 24th number to the right of the decimal) to its value using the functions in T). For example, consider the real number 0.3128..... If we apply F(0.3128....) we end up with the following set of functions from T: (f(1->3), f(2->1), f(3->2), f(4->8), ...). We have just set up a one-to-one correspondence between T and the set of reals between 0 and 1. So, the set of functions is uncountable.

Now, consider that an algorithm or computer program can be viewed as nothing more than a finite-length string of symbols created using a finite alphabet. Imagine the binary equivalent string of any program (just use the binary ASCII codes to represent each char). Let P be the set of all computer programs in some language. Given any program in P translated to 0's and 1's, the program is just one long string of 1's and 0's. Order these strings by length, putting shorter before longer. Then, order all the strings of a given length by regarding each string as an actual binary number - write the numbers in ascending order. Either P is infinite or P is finite. If P is finite, it is countable. If P is infinite, we can define a function f(n) = the nth program in the list. This function maps all computer programs to the natural numbers. Therefore, the set of computer programs is countable. There is no one-to-one correspondence between the integers and the reals, so there must exist functions with no corresponding algorithm. There are just too many functions and only a countable number of algorithms. So, uncomputable functions must exist.

Bibliography

Of historical interest (Note that nearly all of the papers on computability are reprinted in Davis):

G. Cantor, *Contributions to the Foundations of the Theory of Transfinite Numbers,* New York: Dover, 1947 (reprinted from the original of 1895).

A. Church, "An Unsolvable Problem of Elementary Number Theory," *American Journal of Mathematics,* 58 (1936), 345-363.

M. Davis, *The Undecidable*, Hewlett, NY: Raven Press, 1965.

K. Gödel, "Über Formal Unentscheidbare Sätze der Principia Mathematica und Verwandter Systeme, I," *Monatshefte für Mathematik und Physik*, 38 (1931), 172-198. (On Formally Undecidable Propositions of Principia Mathematica and Related Systems)

D. Hilbert, *Grundlagen der Geometrie*, Leipzig, 1899. (The Foundations of Geometry is regarded as the first to display Hilbert's axiomatic method.)

S. Kleene, "General Recursive Functions of Natural Numbers," *American Journal of Mathematics*, 57 (1935) 153-173, 219-244.

S. Kleene, *Introduction to Metamathematics*, New York: Van Nostrand, 1952.

E. Post, "Finite Combinatory Processes-Formulation I," *Journal of Symbolic Logic*, 1 (1936), 103-105.

E. Post, "A Variant of a Recursively Unsolvable Problem," *Bulletin of the American Mathematical Society*, 52 (1946), 246-268.

B. Russell, A. Whitehead, *Principia Mathematica*, Cambridge: Cambridge University Press, 1910.

A. Turing, "On Computable Numbers with an Application to the Entscheidungsproblem," *Proceedings of the London Mathematical Society*, 2, No. 42 (1936), 230-265. (Entscheidungsproblem means decision or decidable problems)

Epp has a thorough section on countability, (the above section on infinite sets was adapted from this). For more on paradoxes of set theory see Stoll, and Smullyan for general paradoxes. For a good introduction to computability see Minsky (a classic text in CS); Hopcroft & Ullman (another classic), Lewis and Papadimitriou and Martin give a more detailed treatment.

S. Epp, *Discrete Mathematics with Applications*, Belmont, CA: Wadsworth, 1990.

J. Hopcroft, J.D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Reading, MA: Addison-Wesley, 1979.

H. Lewis, C. Papadimitriou, *Elements of the Theory of Computatio*n, Englewood Cliffs, NJ: Prentice Hall, 1981.

J. Martin, *Introduction to Languages and the Theory of Computation*, New York: McGraw-Hill, 1991.

M. Minsky, *Computation: Finite and Infinite Machines*, Englewood Cliffs, NJ: Prentice Hall, 1967.

R. Smullyan, *What is the Name of This Book - The Riddle of Dracula and Other Logical Puzzles*, Englewood Cliffs, NJ: Prentice Hall, 1978.

R.R. Stoll, Sets, *Logic and Axiomatic Theories*, 2nd ed., San Francisco, CA: W.H. Freeman, 1974.