5-1-1995

Diagonalization arguments: Standardization and implications.

Robert R. Goldberg

SCHOOL OF COMPUTER SCIENCE
AND INFORMATION SYSTEMS

TECHNICAL REPORT
Number 83, May 1995

Diagonalization Arguments: Standardization and Implications

Robert R. Goldberg
Computer Science Department
Queens College
65-30 Kissena Boulevard
Flushing, NY 11367-0904

Natalie Hammerman
Department of Mathematics
Pace University - Westchester
and
Department of Computer Science
CUNY Graduate Center

PACE
UNIVERSITY

Dr. Robert R. Goldberg is a tenured Assistant Professor at the Computer Science Department of Queens College. His research interests span the realms of automata, computer graphics, computer vision, expert systems, and graph theory and routing problems. He received his Ph.D. from the Courant Institute of New York University under an Office of Naval Research graduate fellowship. He currently serves on the ONR Fellowship Panel as well as on the Research Awards Panel for Computer Science at the Research Foundation of City University of New York.

Natalie Hammerman, an adjunct Assistant Professor in the Mathematics Department, is a doctoral student in computer science at the City University of New York Graduate Center.

Diagonalization Arguments: Standardization and Implications

Robert Goldberg
Computer Science Department
Queens College of CUNY
65 - 30 Kissena Blvd.
Flushing, New York 11367

and

Natalie Hammerman
Department of Mathematics
Marks Hall
Pace University
861 Bedford Road
Pleasantville, NY 10570

Email: goldberg@qcunix.acc.qc.edu

Abstract

In theoretical computer science it is often necessary to distinguish sets based on their cardinality. The Church-Turing thesis, for example, formulates the capabilities of computation by asserting that algorithms over countable sets are effectively computable. However, algorithms for computation over uncountable sets are not possible due to to the infinite precision required to distinguish arbitrary elements of the set. Cantor introduced the diagonalization technique for proving infinite sets uncountable. Typically, diagonalization proofs vary depending on the particular domain of the set. Over time, the significance of the diagonal has eroded from this proof by contradiction method. Thus, developing the ability to apply the technique to new cases may be elusive.

In this paper, a template for standardizing diagonalization proofs is developed. This template is applied to a number of specific examples to illustrate the proper application of the diagonalization argument. With the use of data structures implementing sets, we present a formal analysis of the template proof to show that countable sets do not provide adequate information for the contradiction required by diagonalization proofs. This further elucidates the associated computer and mathematical theory concepts.

1. Introduction

Computability theory provides the computational process with mathematical and logical foundations. The basis for a computational model is is the Church-Turing thesis which states that any algorithm over the set of natural numbers can be implemented on a (Turing) machine. As such, categorizing infinite sets based on their cardinality plays an intrinsic role in understanding the capabilities and limitations of computation on modern machines.

Cantor (1936) introduced the diagonalization method to determine whether an infinite set is uncountable. His original proof was applied to the set of total functions over $\mathbb{N}$. However, the diagonalization method has since been applied in different formats to accommodate domains which are not sets of functions. This resulted in an unclear presentation of how to transfer the technique to further cases. It has reached a point where the notion of "diagonal" has been eroded from these approaches. In this paper, Cantor's original proof is abstracted by providing a standardized template proof that is readily available for applications to different infinite sets. This is accomplished by requiring a characterizing function to be defined for each element of the set under consideration. Examples are presented to illustrate the strengths of the method. The critical points are identified; these must be handled carefully in order to guarantee a valid proof. By using data structures to implement set constructs, we show that countably infinite sets do not provide enough information for a contradiction in the diagonalization argument. Although the template was tailored for understanding diagonalization arguments, it also provides an important educational tool for clarifying difficult concepts in both mathematical and computer science theory.

In the next section set constructs are analyzed. A proven 1-1 correspondence between $\mathbb{N}$ and the set of finite subsets of $\mathbb{N}$ uses a representation for these sets which is extended to provide a similar representation for infinite sets. A diagonalization argument involves a proof by contradiction, which also requires an enumeration of all the elements of the given set. In both the countably infinite and uncountable cases, it is necessary to consider a 1-1 correspondence between the set under consideration and $\mathbb{N}$; however, in the uncountable case, the diagonalization argument is used to prove the nonexistance of such an enumeration. Because our academic experience has indicated student difficulty in grasping these concepts, a template is provided in section 3 for standardizing diagonalization arguments based on Cantor's original proof to show that the set of total functions over $\mathbb{N}$ is not countable. With this, P($\mathbb{N}$) and $\mathbb{R}$ (power set of $\mathbb{N}$ and real numbers, respectively) are shown to be uncountable. These examples are used to indicate the care needed in specifying a proper diagonalizing function within the diagonalization argument in order to yield a valid proof. With the template, the notion of the "diagonal" is clearly defined. Students regularly raise the question as to why the diagonalization argument fails for a countably infinite set; therefore, section 4 presents a mathematical formalization that explains why diagonalization will not succeed when applied to a countable infinite set. Further examples involving elements of the set of rational numbers are used to indicate subtle flaws which can invalidate a diagonalization argument; these flaws are elusive to students studying computability. Based on this analysis, a characterization of countable and uncountable sets is presented that enhances the assertion of the Church-Turing thesis.

2. Set Constructs

To analyze the properties of (un)countable sets, the set of (in)finite subsets of $\mathbb{N}$ will be considered. In this context it is necessary to determine the elements of E_j, the j^{th} finite subset of $\mathbb{N}$.

A set is an unordered collection of elements. Associated with each set is its membership function, M(x,S) ≡ (x ∈ S), that is

$$M(x,S) := \begin{cases} 1, & x \in S \\ 0, & x \notin S. \end{cases}$$

While the elements of sets are unordered, theorems about their cardinality will require an imposed order by putting the elements in a one-to-one correspondence with a subset of the natural numbers. The recursive enumeration of the finite subsets of $\mathbf{N}$ provides such an order. The one-to-one correspondence between $\mathbf{N}$ and the finite subsets of $\mathbf{N}$ will then be demonstrated by the lemma, which provides for an actual implementation of these set constructs.

The recursive enumeration of the finite subsets of $\mathbf{N}$ can be formulated as follows :

BASE CASE :

$$S_0 = \{\varnothing\} = \{E_0\}.$$

RECURSIVE STEP :

$$\begin{aligned} S_{i+1} &= S_i \cup \{E_j \cup \{i\} \mid E_j \in S_i, j = 0, 1, \cdots |S_i| - 1\} \\ &= \{E_j \mid j = 0, \cdots, 2|S_i| - 1\}. \end{aligned}$$

In this enumeration, the union operator assumes ordered sets for its arguments and returns an ordered set. Thus, in the set S_{i+1} (the set of all subsets of $\{0, \ldots ,i\}$), the elements of S_i appear first, followed by those elements in the same order but with natural number i added. A subsequent lemma demonstrates that $E_j \in S_i$ (the j^{th} finite subset of the recursive enumeration) contains precisely those elements n that have a bit set to one in the n^{th} position of the binary representation of j. The reader is referred to figure 1 for examples of $i = 0,1,2,3$.

Consider the binary representation of $j = b_k b_{k-1} \cdots b_2 b_1 b_0$. Let $\mathrm{M}(n, S_j) = b_n$, $0 \le n \le k$ and $\mathrm{M}(n, S_j) = 0, n > k$. This induces a finite subset of $\mathbf{N}$ containing precisely those numbers n such that the digits $b_n = 1$ in the associated binary representation of j. An extension of this bitstring construct allows for a representation of the infinite subsets of $\mathbf{N}$ as well; however, in this case there is no finite k such that for $n > k$, $M(n, S_j) = 0$. This construct permits the retrieval/storage of the set associated with any bitstring. In particular, for the finite subsets of $\mathbf{N}$, this permits the retrieval/storage of the set associated with any natural number. For example, to determine the elements of E_{50}, the binary representation of 50 is needed: $50_{10} = 110010_2 = b_5b_4b_3b_2b_1b_0$. Since b_1, b_4, and b_5 are 1, $E_{50} = \{1,4,5\}$. This process can easily be reversed. Given a finite subset of **N**, the number representing this set can easily be obtained. For the set $\{1,2,4\}$, $b_1 = b_2 = b_4 = 1$ and all other $b_i = 0$. The decimal value of $b_4b_3b_2b_1b_0 = 10110_2 = 22_{10}$ is therefore the unique natural number corresponding to the set.

The next theorem guarantees the one-to-one correspondence between **N** and the finite subsets of **N**. The significance of this theorem in our discussion is threefold. First, the finite subsets of **N** are countable. Second, the theorem emphasizes the nature of a proof which guarantees the one-to-one correspondence between an infinite set and **N**. This is important when trying to categorize the cardinality of an infinite set. Third, this theoretical construct actually provides a practical implementation of (in)finite sets for computer science theory: namely, a bitstring data structure.

Theorem. $\exists$ a 1-1 correspondence between **N** and the finite subsets of **N**.

Proof. The following lemma demonstrates that after the i^{th} iteration of the above recursive enumeration, S_i is an ordered power set of the natural numbers < *i*. In addition, the order in S_i of the elements E_j (subsets of $\{0, \cdots, i-1\}$) is given by the binary representations of the sequence [0 ... 2^i-1]. Thus, this defines a 1-1 correspondence between [0 ... 2^i-1] and the elements in the ordered set S_i. This lemma also indicates that each $E_j \in S_i$ contains precisely those numbers such that the n^{th} digit ($0 \le n \le k$) of the binary representation of $j = b_k \cdots b_0$ is 1. Since this is true for all *i*, the above recursion enumerates the finite subsets of **N** creating a 1-1 correspondence between **N** and the finite subsets of **N**.

□

Lemma. After applying the above recursive union i times the following results hold true. S_i contains an ordered power set of the natural numbers $<i$. This order is enumerated by the sequence [0 ... 2^i-1]. Furthermore, each $E_j \in S_i$ contains precisely those numbers such that the n^{th} digit of the binary representation of j is 1.

Proof. By classical induction.

BASE CASE :

$i = 0$. By definition, $S_0 = \{\emptyset\}$, the power set of { }. S_0 is enumerated by $[0 \ldots 2^0-1] = [0 \ldots 0] = [0]$. Hence, the only element of S_0 is E_0. The binary representation of the index of this set is 000...0. Thus, $\forall k, b_k=0$, corresponding to E_0 having no elements. The lemma holds true in this case.

INDUCTION HYPOTHESIS :

Assume that S_i contains the ordered power set of the natural numbers $<i$ and each element $E_j \in S_i, j=0, \cdots ,2^i-1$ contains precisely those numbers such that the n^{th} digit of the binary representation of j is 1.

INDUCTIVE STEP :

Prove that S_{i+1} contains an ordered power set of $\{0, \cdots ,i\}$ and each element $E_j \in S_{i+1}, j=0 \cdots 2^{i+1}-1$ contains precisely those numbers such that the n^{th} digit of the binary representation of j is 1.

The power set of $\{0, \cdots ,i\}$ partitions into the subsets which do not contain i and those subsets which do. By the induction hypothesis, the former set of subsets S_i is precisely the first $|S_i|$ elements of S_{i+1}. Any subset of $\{0, \cdots ,i\}$ which contains i, consists of a subset of $\{0, \cdots ,i-1\}$ union $\{i\}$. Since $\{E_j \cup \{i\} \mid E_j \in S_i, j=0, \cdots ,2^i-1\}$ produces the last $|S_i|$ elements of S_{i+1}, this set contains precisely all of these remaining subsets. Hence, S_{i+1} is an ordered power set of $\{0, \cdots ,i\}$.

The induction hypothesis then indicates that each element $E_j \in S_i, j=0, \cdots ,2^i-1$ contains precisely those numbers from $\{0, \cdots ,i-1\}$ whose corresponding positions in the binary representation of j is set to one. Then, after one iteration of the recursive union above, the first $|S_i| = 2^i$ elements of the ordered power set S_{i+1} must have the elements of S_i in the same order as S_i, since $S_{i+1} = S_i \cup \cdots$ uses the union operator that returns an ordered set. Thus, the first 2^i elements of S_{i+1} are enumerated by the binary representations of [0 ... (2^i - 1)] in order. The next $|S_i| = 2^i$ elements formed by the union in the loop $\{E_j \cup \{i\} \mid E_j \in S_i, j = 0 \cdots 2^i - 1\}$ have the same enumeration of binary representations but with an extra bit set to one for the number i (see figure 1). The value added to each of the numbers of the corresponding sequence is 2^i. Thus, the actual values corresponding to the new elements added are [(0+2^i) ... ((2^i -1)+2^i)] = [2^i ... (2^{i+1} -1)]. Hence, the total sequence of elements in S_{i+1} has in order the finite subsets whose binary representations correspond to the ordered sequence [0 ... (2^i−1), 2^i ... (2^{i+1} -1)] = [0 ... (2^{i+1} - 1)].

By the recursive enumeration, the first half of S_{i+1} is identical to S_i. Since for S_i, $M(n, E_j) = b_n$, $0 \le j \le 2^i - 1$, where the binary representation of j is $b_k \cdots b_0$ ($k < i$ and $b_k = 1$), the membership function for the sets in the first half of S_{i+1} is

$$M(n, E_j) = \begin{cases} b_n, & n \le k \\ 0, & n > k. \end{cases}$$

Furthermore, by the construction of S_{i+1}, $E_{j+2^i} = E_j \cup \{i\}$, $0 \le j \le 2^i - 1$. Therefore, for the sets in the second half of S_{i+1} the membership function is

$$M(n, E_{j+2^i}) = \begin{cases} b_n, & n \le k \\ 0, & n > k \text{ and } n \ne i \\ 1, & n = i. \end{cases}$$

Thus, each element $E_j \in S_{i+1}, j = 0, \cdots, 2^{i+1} - 1$ contains precisely those numbers such that the n^{th} digit of the binary representation of j is 1. □

3. Standardization of Diagonalization Proofs

Cantor introduced the diagonalization method to determine the uncountability of infinite sets. Consider the set F of total functions over the natural numbers, $\mathbf{N}$. Assume that F is countable. Then, there exists a 1-1 correspondence between $F = \{f_i \mid f_i : \mathbf{N} \rightarrow \mathbf{N}\}$ and $\mathbf{N}$. Construct f_{new} such that $f_{new}(i) \ne f_i(i)$ and $f_{new} \in$ F. Cantor used $f_{new}(i) = f_i(i) + 1$. Since $f_i(i) \in \mathbf{N}$ is uniquely defined, the same is true for $f_{new}(i)$ and hence, f_{new} is a total function on $\mathbf{N}$. But, then $f_{new} \in$ F ; hence, $f_{new} \equiv f_k$ for some k by virtue of the 1-1 correspondence between F and $\mathbf{N}$. In particular, $f_{new}(k) \equiv f_k(k)$. However, $f_{new}(k) \neq f_k(k)$ by the construction of f_{new}. This contradiction indicates that no natural number k can be found for $f_{new} \in$ F. Since there exists an $f_{new} \in F$ which cannot be counted, F is *not* countable.

Thus, to apply the diagonalization method to any uncountable set Q, it is necessary to

D1) define a property P which allows for an element to be a member of a set, say Q; (in the above proof, P = "total functions over $\mathbf{N}$")

D2) define and count (by indexing) total functions f_i that distinguish (*characterize*) the elements of Q; (since the above set F only contains total functions, these functions f_i are automatically defined)

D3) define $q_{new} \in Q$ with property P by designing its characterizing function f_{new} such that $f_{new}(i) \neq f_i(i)$; (f_{new} above was shown to be total, that is it possesses property P since $\forall i, f_{new}(i) \in \mathbf{N}$; also, the inequality condition was provided for by f_{new}).

Having proved that this set F is not countable, a standard template for applying the diagonalization argument to any uncountable set now emerges. This template clearly demonstrates what this method incorporates from the "diagonal." To prove that a set Q containing elements with property P is not countable, prove that the associated set F of characterization functions for property P distinguishing each element of Q is **not** countable.

DIAGONALIZATION PROOF TEMPLATE

1) Let Q be an (uncountable) infinite set with elements containing property P.

2) Proof by contradiction. Assume that set Q is countable.

3) By definition of a countable set, there exists a one-to-one correspondence between the elements of Q and $\mathbf{N}$.

4) Construct F, the set of characterizing functions distinguishing each element of Q. Since F represents the countable set Q, F is also countable. Consequently, an index number (subscript) can be assigned to each characterization function $f_i \in$ F corresponding to each $q_i \in Q$ having property P.

5) Construct lookup tables for the outputs of the characterization functions $\{f_i \mid i \in \mathbf{N}\}$ with the elements of $\mathbf{N}$ as inputs. This creates a two-dimensional matrix with the f_i on the vertical and the natural numbers on the horizontal; the outputs $f_i(n)$ are stored in the matrix cells. (See figure 2.)

6) Design $f_{new} \in$ F, the characterizing function of $q_{new} \in Q$ with property P such that $f_{new}(i) \neq f_i(i)$. (It is obvious from figure 2 that the main diagonal of the matrix is being utilized to define f_{new}.)

7) Since q_{new} contains property P, the position of f_{new} can be obtained in the countable list of such elements by the one-to-one correspondence guaranteed in step 3 and constructed in step 4. Let f_{new} be the k^{th} element in the ordered listing of F.

8) Compute $f_{new}(k)$. This equals $f_k(k)$ because of its position on the list of elements in F. However, $f_{new}(k) \neq f_k(k)$ by the diagonalization construction in step 6. For any k, $f_{new}(k)$ differs from the diagonal element of row k in the matrix of lookup tables. Thus, an $f_{new} \in$ F (and hence the corresponding $q_{new} \in Q$) has been found which was not counted.

9) The assumption that Q and hence F is countable proves false because the contradiction indicates that a one-to-one correspondence cannot exist between the elements of F and $\mathbf{N}$.

10) Conclusion: F is not countable; therefore Q is **not** countable.

To use this template to show that the power set of $\mathbf{N}$ is not countable, the three definitions above (**D1-D3**) have to be stated. First, define P as the "subsets of $\mathbf{N}$" (criteria **D1**). Then, the characterizing function f_i is the infinite bitstring implementation of the membership function (criteria **D2**). Finally, the diagonalizing function $f_{new}(i)$ is simply

$\neg f_i(i)$ (read "NOT $f_i(i)$"); the elements of q_{new} are precisely those j such that $f_{new}(j)=1$ (criteria **D3**). The proof can then proceed identically with the template provided.

This bitstring implementation with its corresponding diagonalizing function in fact suggests that two other sets are not countable. The bitstring as an abstract data type implements not only subsets of **N**, but also the Boolean predicates on **N**. Each row of **0**'s and **1**'s, which in the previous example represented a subset of **N**, now represents the outputs of P_i, the i^{th} predicate function (criteria D1 and D2). Criteria D3 is satisfied identically to the previous example. This set is thus proven not countable by a similar use of the template proof. Also the bitstring may represent the binary digits of the real numbers between 0 and 1. However, it will be shown that for **R**, a base higher than two must be employed for the template to produce a valid proof.

As stated above, three definitions (**D1-D3**) are required to make the template proof problem specific; this enables the diagonalization method to apply to different set domains. However, the last two (D2 and D3) are sensitive points in that they provide fertile ground for creating an invalid proof. Specifically, by enumerating the f_i (criteria **D2**), an order is imposed on the q_i. The elements of Q must have a 1-1 correspondence with **N** implying that no element of Q is included more than once nor precluded from the list by the imposed ordering. For example, for definition **D1**, let P be "the real numbers between 0 and 1". For simplicity, this subset of the reals is considered. If this set of numbers is uncountable, then so is **R**.

Rationals are elements of **R** and many rationals have two decimal formats (terminating and repeating). For example, $2/5 = 0.4 = 0.3999... = .3\bar{9}$. It is important to count each rational number with this property only once; otherwise this violates the 1-1 correspondence between Q and **N** (step 3 in the template proof). Therefore, when a rational number, and hence a real number, has two decimal representations, only one of them may be used, say the repeating decimal format. Arbitrarily select the numbers, one at a time, and assign them sequential indices. Let $f_i(j)$ represent the j^{th} digit of real number q_i (**D1**).

However, using the repeating decimal format alone is not sufficient to yield a valid proof. A first attempt at defining f_{new} (criteria **D3**) might adapt Cantor's original diagonalization function for f_{new} to the set of reals by using modular arithmetic to guarantee that each output is in the set of digits {0,...,9}; thus, $f_{new}(i)=(f_i(i)+1)\ mod\ 10$. But, under this definition of $f_{new},\ q_{new}$ may be in terminating decimal format, that is q_{new} could possibly end with an infinite string of zeroes. Thus, q_{new} may already be on the list in its repeating decimal format. This would invalidate the proof. To overcome this problem, correct

$$f_{new}(i)=\begin{cases} f_i(i)+1, & f_i(i)\neq 9 \\ 1, & f_i(i)=9. \end{cases}$$

This definition guarantees that q_{new} is a real number between 0 and 1, and is not in terminating decimal format. Satisfying criteria **D3** in this way yields a valid proof. Similarly, note what would happen if the bitstring (binary) representation for members of $\mathbf{R}$ were used. Then, $f_{new}(i)=(f_i(i)+1)\ mod\ 2\equiv f_i(i)$. Again, under this definition of f_{new}, the associated q_{new} may be in terminating decimal format, ending with an infinite string of zeroes. But, in base 2, no other digit exists besides 0 and 1 to correct the problem. For definition **D2**, counting the terminating binary representation rather than the repeating one, when both exist, will not alleviate the problem. Thus, some other base greater than 2 must be used to create a valid proof using the template.

While it is relatively simple to make sure that for $i\in\mathbf{N}, f_{new}(i)\neq f_i(i)$, guaranteeing that the corresponding q_{new} has property P may be more difficult (criteria **D3**). For example, to prove that the monotonically increasing functions are uncountable, let Q be the set of those functions (criteria **D1** and **D2**). After setting up the matrix as in figure 2, to ensure that $f_{new}(i)\neq f_i(i)$, a first attempt for f_{new} might be $f_{new}(i)=f_i(i)+1$; but this does not guarantee that f_{new} is monotonically increasing. Therefore, this results in an invalid proof. Defining the diagonalizing function, (Sudkamp, 1989)

$$f_{new}(i)=\begin{cases} f_i(i)+1, & i=0 \\ max(f_{new}(i-1), f_i(i)) + 1, & i>0 \end{cases}$$

will result in f_{new} being monotonically increasing and differing from each element on the diagonal of the table in figure 2 (**D3**); this yields a valid proof. These two examples of uncountable sets emphasize that even when a valid diagonalization proof exists, a poor choice for f_{new} results in an invalid proof. In the next section, specific cases of countably infinite sets are utilized to illustrate what can go wrong if the template guidelines are not strictly adhered to. The flaws in these invalid proofs are elusive to most students.

4. Improper Application of Diagonalization Proofs

It has been shown that while the set of *finite* subsets of **N** are countable, P(**N**), the set of *all* subsets (power set) of **N** is not. However, through computer science constructs of sets (bitstrings), similar characterization functions f_i are implemented for elements of both set types. This seems to question the validity of the diagonalization method: can countable infinite sets be successfully pushed through the template proof? By answering this question, an insight into diagonalization arguments will surface which will also shed light on the essential difference between countably infinite and uncountable sets, and their corresponding cardinalities of **N** and **R**.

Suppose an attempt is made to apply the diagonalization template proof for uncountability to the **finite** subsets of **N**, which has already been proven countable. Then P in step 1 of the template proof is the "finite subsets of **N**" (criteria **D1**). For step 3, set up the one-to-one correspondence and the corresponding bitstring implementation shown before for the finite subsets of **N** (criteria **D2**). This means that the order in the list of functions implies the actual elements of its corresponding set and that these elements can be identified. This specific order allows for the flaw in the proof to be located. Note that in this case each row in the matrix is the binary representation of *i* in reverse order followed by an infinite sequence of zeroes. This reversal simply follows from the fact that in a number, the positions of digits are written from most significant (left) to least significant (right), whereas columns in matrices increase from lowest rank (left) to highest (right). So, for example the binary representation of 6 is 110 and yet the corresponding set q_6 will be stored as 011 → 01100000... in the row of the lookup matrix for f_6. This row represents $M(j, q_6), j \in \mathbf{N}$.

Consider step 6 where q_{new} is defined by $B_0 B_1 \cdots$ where $B_i = f_{new}(i)$. Since $f_i(j)$ can only be 0 or 1 and $f_{new}(i) \neq f_i(i)$, $B_i = f_{new}(i) = {}^{\sim} f_i(i)$ is the only choice for f_{new}. Then, the proof seems to proceed as normal. The next theorem indicates the subtle flaw in this argument. Based on the characterizing functions for the elements of this countable set, it has been shown that only a *finite* amount of information is necessary to distinguish the elements of this **countable** (infinite) set; an *infinite* amount of information will be needed to distinguish the elements of an **uncountable** set. This indicates where the proof fails: there will not be enough significant information in any given row of the lookup table for a contradiction to occur at the diagonal element.

Theorem. If $b_0 b_1 \cdots b_d \cdots$ is the infinite bitstring characterization of finite subset q_d of $\mathbf{N}$ in the ordered list, then $b_i = 0, i \geq d$.

Proof. q_0 represents the empty set and appears as the first element ($d = 0$) in the ordered list. Its bitstring representation is an infinite string of zeroes; therefore, $b_i = 0, \forall i \in \mathbf{N}$. For all $d > 0$, $2^k \leq d < 2^{k+1}$ for some $k \in \mathbf{N}$. The number 2^k is the first (least) natural number to require $k+1$ bits to represent it. Similarly, the number 2^{k+1} is the first (least) natural number to require $k+2$ bits. Hence, d requires $k+1$ bits to represent it; let the binary representation of $d = b_k b_{k-1} \cdots b_0$. A simple argument can be developed to show that $k < 2^k, k \in \mathbf{N}$; therefore, $2^k \leq d$ implies that $k < d$. It then follows that the associated infinite bitstring representation for d, represented by $b_0 b_1 \cdots b_{k-1} b_k \cdots b_i \cdots$ in the lookup table, contains zeroes for all other digits $b_i, i > k$. Therefore $\forall i \geq d, b_i = 0$. □

The result of this theorem is now analyzed within the context of the application of the template proof to the set of finite subsets of $\mathbf{N}$. Specifically, this theorem implies that the $k+1$ bits needed to represent positive d occur before the bit b_d in the infinite bitstring corresponding to q_d (and when $d=0$, $b_d=0$). Therefore, $b_i = 0$ whenever $i=d$. Hence, $(\forall i \in \mathbf{N})$ $(f_i(i) = 0)$ in the lookup matrix of step 5 of the template proof (figure 2). Therefore, $B_i = f_{new}(i) = \tilde{\ } f_i(i) = 1, i \in \mathbf{N}$. Thus, q_{new} is characterized by the bitstring 1111... and represents **N**, which is **not** a finite subset of **N**. This example stresses a very important point which was likewise noted in the previous section. When using the template proof for uncountability, one must be absolutely sure that the f_{new} constructed for step 6 of the proof actually defines an element with the desired property P; otherwise a seemingly "valid" proof has no validity. If it had not been observed that the f_{new} in this example represents an infinite set, it would appear that a proof exists showing that the finite subsets of **N** are uncountable. It was the special ordering imposed on the elements of Q which permitted the deduction that $q_{new} \notin Q$.

The previous attempt to prove that the finite subsets of **N** are uncountable failed because the essential bits of information to define each finite subset of **N** are stored before the diagonal element on each row of the lookup table; all other bits were set to zero to fill the remaining cells of the lookup table. A similar observation can be noted when attempting to apply the diagonalization template to any other countable set **C**. Since $\exists$ a 1-1 correspondence between **C** and **N**, the characterizing functions for **C** can be based on the corresponding index in **N**. This allows for the special ordering imposed on **C** to permit the above discussion to be applicable to any countable set. Hence, the flaw in the proof will be detectable.

However, there exist countable sets which have legitimate infinite representations. For example, consider the set of rational numbers, which is countable. Each rational number can be represented by a repeating decimal. Some rationals have a decimal representation only as a repeating decimal, for example 2/11 = .181818... = $.\overline{18}$. Other rationals can be represented as a terminating decimal, but such numbers also have a repeating decimal representation, for example 2/5 = .4 = .399999... = $.3\overline{9}$. This representation bypasses the limitation of having only a finite amount of essential data to represent each number. The question arises as to whether this countable set can be forced through the template proof for uncountability using the infinite representation for its elements.

In the attempt to apply the template proof to the rational numbers, it is only necessary to look at the subset $Q = \{p/q \mid p,q \in \mathbf{N}$ and $0 < p \leq q\}$, that is the rationals between 0 and 1. If this set of numbers is (un)countable, then so is the set of all rationals. To apply the template proof to Q, the three items must be defined. First, let property P be the "rational numbers between 0 and 1" (criteria **D1**). Next the characteristic function is defined; for $q_i = .d_{(i,0)}d_{(i,1)}d_{(i,2)}\cdots \in Q$, $f_i(j)=d_{(i,j)}$, where q_i is in its repeating decimal format (criteria **D2**). Finally, the diagonalizing function is constructed (criteria **D3**). Since the repeating decimal format is used, f_{new} can be chosen as in the real number case:

$$f_{new}(i)=\begin{cases} f_i(i)+1, & f_i(i)\neq 9 \\ 1, & f_i(i)=9. \end{cases}$$

This definition provides for a q_{new} that is not in terminating decimal format, that is q_{new} does not end with an infinite string of zeroes. However, this proof cannot be completed because there is no guarantee that q_{new} defined by f_{new} is actually a repeating decimal; it could be irrational. To overcome this problem a special ordering of Q is thus required to assure that q_{new} is a repeating decimal. For example, when setting up the one-to-one correspondence between Q and $\mathbf{N}$, let $q_i = .d_{(i,0)}d_{(i,1)}d_{(i,2)}\cdots \in Q$, only if $d_{(i,i)} = i$ mod 10. Then, f_{new} as previously defined for the rationals, will produce q_{new} as a repeating decimal, namely $.\overline{1234567891}$. But, the proof is invalid with this scheme. There are many repeating decimals, besides q_{new}, which will not appear on the ordered list, for example $.\overline{1234567890}$, $.\overline{9876543210}$, and $.\overline{2252222222}$, etc. It should be noted that *any* special ordering forcing q_{new} to be a repeating decimal will result in some rational numbers being omitted from the list. These numbers are not counted because the selection of the correspondence between Q and $\mathbf{N}$ omitted them, rather than because the set Q is not countable. Thus, it is imperative that any scheme which is used to count the elements of Q not preclude any elements of Q.

Applying the diagonalization template proof to an uncountable set yields a q_{new} that cannot be found on the ordered list due to the fact that Q contains more elements than can be counted. Applying the template proof in error to a countable set causes q_{new} not to be found on the ordered list because either the ordering scheme does not properly list all elements of Q or q_{new} does not have property P. Since the sets of rationals and finite subsets of $\mathbb{N}$ are countable, all efforts to apply the template proof to these sets will fail to meet some requirement of the proof, but finding the flaw in such attempts can be elusive.

5. Conclusion

The standardization of Cantor's original proof provides an important educational tool. First, it introduces the concept of a diagonalization argument to prove uncountability. Second, it can be used to highlight common errors that invalidate a diagonalization proof. Third, this template proof exemplifies the incorporation of computer science constructs to explain mathematical theoretical concepts. In particular, the bitstring implementation defined the membership function for sets. As a result, the inherent difference between countably infinite and uncountable sets can thus be exhibited. A correct understanding of this difference permits for the application of the diagonalization technique to appropriate infinite sets. This also supports the Church-Turing thesis and enhances its interpretation.

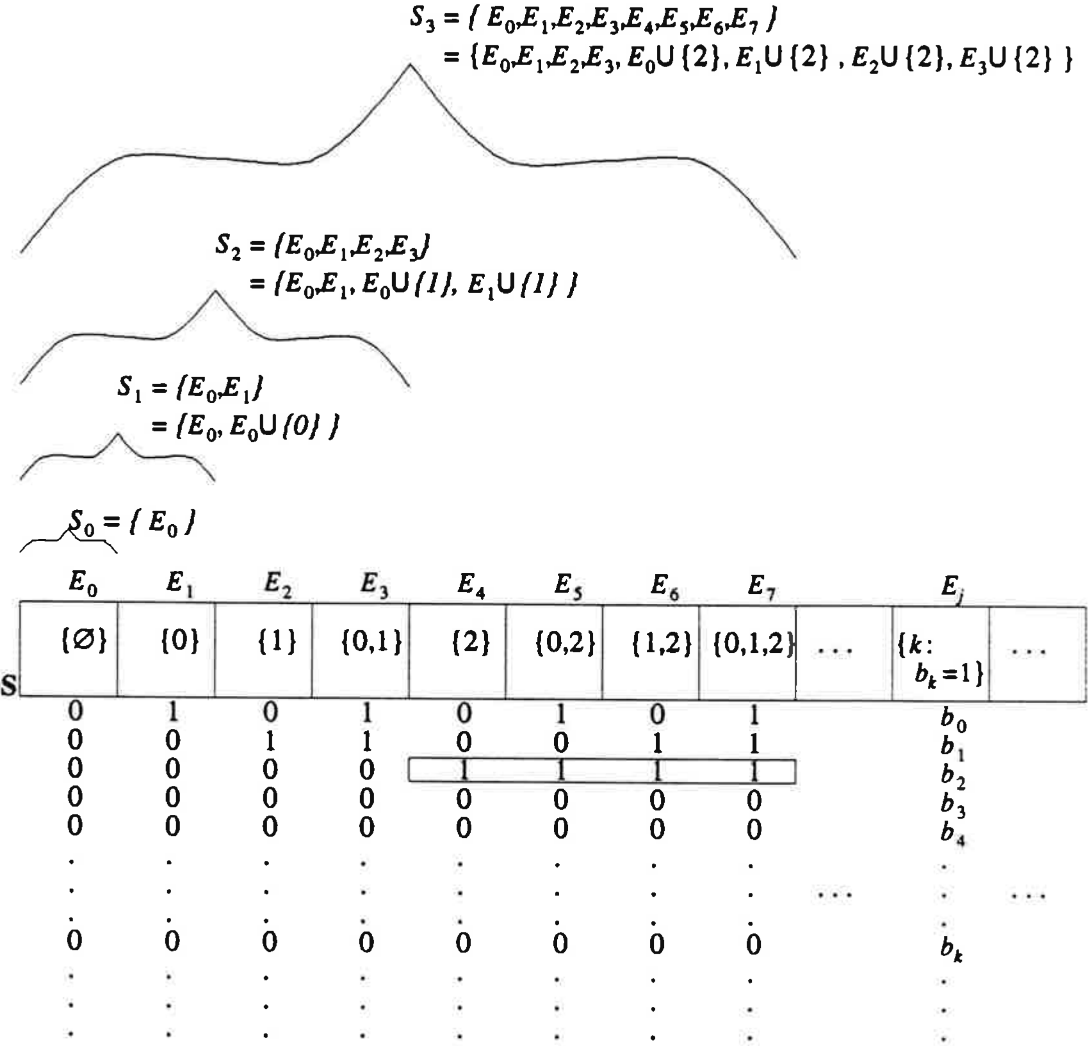
By analyzing Cantor's original diagonalization argument to prove that the set of total functions over $\mathbb{N}$ is uncountable, a standardized template was constructed for proving infinite sets uncountable. Characterization functions were implemented representing the different elements of specific infinite sets. Using data structures implementing the elements, two critical points of the proof were observed that could allow for a seemingly valid proof which in actuality is flawed. First, to properly categorize the countability of an infinite set, an order is imposed on the elements by putting them in a 1-1 correspondence with the natural numbers $\mathbb{N}$. This order must be carefully designed so that elements of the set are neither precluded nor counted more than once in different formats.

Generally, an arbitrary ordering of the elements of the set Q is used. The ability or need to impose a special ordering seems to create fertile ground for a flaw to arise in the proof. For an uncountable set, any ordering will provide a valid proof, once the characterizing and diagonalizing functions have been properly defined. Second, the applicability of diagonalization does not depend on the existence of a representation of members of a set using an infinite amount of data to define each element of the set. Rather, it relies on the nonexistence of a representation allowing a finite amount of data to define each member of the set. While some elements of an uncountable set may be well defined with a finite amount of data, the characterization of such elements is extended to provide a uniform representation for all elements of the set.

Set Q is uncountable when an *infinite* amount of data is mandatory to define membership for elements of Q. Proofwise this means that the template proof or any version of Cantor's diagonalization is applicable, and a careful choice of f_{new} will produce a valid proof. It is this infinite storage requirement for distinguishing the elements of the set that prevents computation on computing machines, since any procedure which must process an infinite amount of information will not halt. For example, many real numbers must have an infinite sequence of digits, and infinite subsets of $\mathbf{N}$ must have an infinite sequence of bits to define the membership function. Consequently, these two sets, $\mathbf{R}$ and P($\mathbf{N}$), are uncountable and hence, procedures for computation over these sets are not effectively computable.

Set Q with property P is countable when all (but a finite number of) $q \in Q$ can be represented by a *finite* amount of data to define the elements that have property P. Proofwise this means a one-to-one correspondence does exist between the elements of Q and $\mathbf{N}$; the q_{new} as defined in the template proof (step 6) which allows for a contradiction cannot be found. Computation over an infinite set in which each element can be characterized by a finite amount of information is effective over the entire (countable) set. For example each member of the finite subsets of $\mathbf{N}$ can be defined with a finite number of bits, and each rational number can be defined using only two integers; these two sets are thus countable and hence by the Church-Turing thesis, algorithms over these sets are computable.

Acknowledgement: The authors would like to thank Dr. Jerry Waxman for his guidance and useful comments.



$S_3 = \{ E_0,E_1,E_2,E_3,E_4,E_5,E_6,E_7 \}$
$= \{E_0,E_1,E_2,E_3, E_0\cup\{2\}, E_1\cup\{2\}, E_2\cup\{2\}, E_3\cup\{2\} \}$

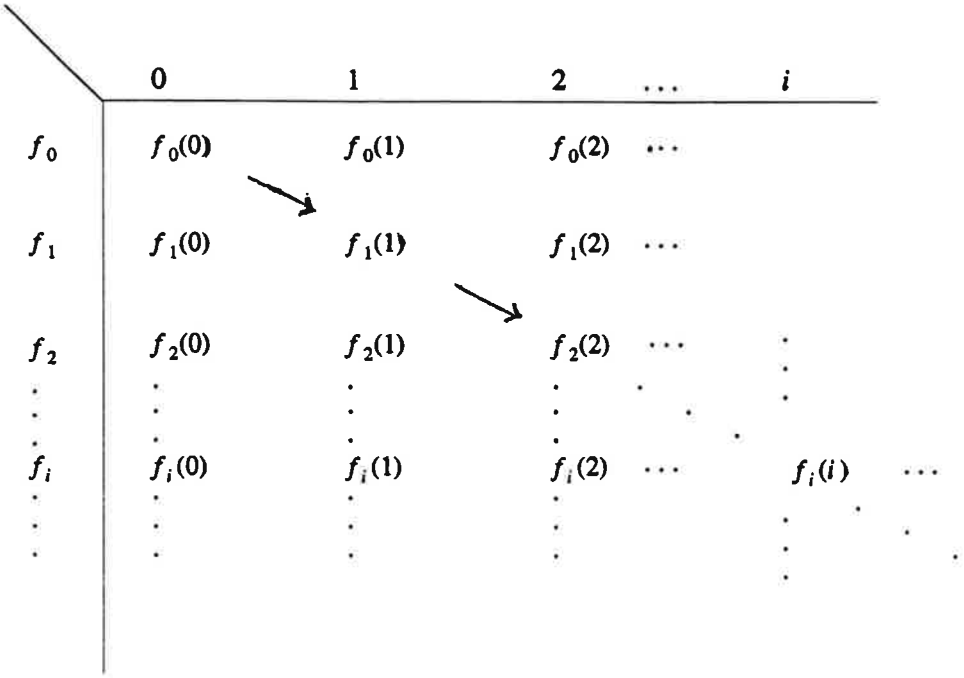
$S_2 = \{E_0,E_1,E_2,E_3\}$
$= \{E_0,E_1, E_0\cup\{1\}, E_1\cup\{1\} \}$

$S_1 = \{E_0,E_1\}$
$= \{E_0, E_0\cup\{0\} \}$

$S_0 = \{ E_0 \}$

	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7		E_j	
POS	$\{\varnothing\}$	$\{0\}$	$\{1\}$	$\{0,1\}$	$\{2\}$	$\{0,2\}$	$\{1,2\}$	$\{0,1,2\}$	...	$\{k:$ $b_k=1\}$	...
0	0	1	0	1	0	1	0	1		b_0	
1	0	0	1	1	0	0	1	1		b_1	
2	0	0	0	0	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$		b_2	
3	0	0	0	0	0	0	0	0		b_3	
4	0	0	0	0	0	0	0	0		b_4	
.	.	.	.	.	.	.	.	.		.	
.	.	.	.	.	.	.	.	.	...	.	...
.	.	.	.	.	.	.	.	.		.	
k	0	0	0	0	0	0	0	0		b_k	
.	.	.	.	.	.	.	.	.		.	
.	.	.	.	.	.	.	.	.		.	
.	.	.	.	.	.	.	.	.		.	

Figure 1. Recursive construction of the finite subsets of $\mathbb{N}$ with corresponding bitstring implementations. The example in this figure is that of the ordered set $S_3 = S_2 \cup \{E_j \cup \{2\} \mid E_j \in S_2\}$. Thus, the top 2 bits for E_4,E_5,E_6,E_7 are the same as those for E_0,E_1,E_2,E_3 respectively and the 2nd bit **POS**ition is set to one (highlighted by the box) in the ordered elements E_4,E_5,E_6,E_7 indicating that $2 \in E_4,E_5,E_6,E_7$. For E_j, j is the decimal equivalent of $b_k b_{k-1} \cdots b_0$ where b_k contains the most significant bit in the column for E_j.



$$
\begin{array}{c|cccccc}
 & 0 & 1 & 2 & \cdots & i & \\
\hline
f_0 & f_0(0) & f_0(1) & f_0(2) & \cdots & & \\
f_1 & f_1(0) & f_1(1) & f_1(2) & \cdots & & \\
f_2 & f_2(0) & f_2(1) & f_2(2) & \cdots & \vdots & \\
\vdots & \vdots & \vdots & \vdots & \ddots & & \\
f_i & f_i(0) & f_i(1) & f_i(2) & \cdots & f_i(i) & \cdots \\
\vdots & \vdots & \vdots & \vdots & & \vdots & \ddots
\end{array}
$$

Figure 2: Two-dimensional matrix of lookup tables for the characterizing functions.

Definitions

cardinality: the number of elements in a set.

characteristic function: the membership function of a set.

characterizing function: for element $q \in$ Q, a function f on **N** that uniquely defines q based on some general representation describing the elements of Q.

Church-Turing thesis: Any algorithm over the natural numbers is effectively computable on a (Turing) machine.

computability theory: the analysis of the computational process.

countable set: a set that has a one-to-one correspondence with a subset of **N**.

diagonalization: a method to prove an infinite set uncountable using a diagonalizing function.

diagonalizing function: given an infinite set Q and enumerated elements $q_i \in$ Q with corresponding characterizing functions f_i, a function **f** on **N** that assigns a value to **f**(i), $\forall i \in$ **N**, based on the value of $f_i(i)$.

infinite set: a set containing a proper subset with equal cardinality.

N = {0, 1, 2, ... }.

one-to-one correspondence: a mapping between two sets such that each element in one set has a unique associated element in the other set, and vice versa.

partition: $\{P_1, \cdots, P_n\}$ is a partition of set S if $\bigcup_{i=1}^{n} P_i = S$ and $\forall i, j \leq n$, $P_i \cap P_j = \emptyset$ when $i \neq j$.

power set: the set of *all* subsets of a set.

recursive enumeration: an enumeration which will index every element of a set S such that

a) one element e_0 is selected first;

b) each $e_{i+1} \in S, i \in$ **N**, is defined as a function of the previously enumerated elements.

total function: a function f:A→B such that $\forall a \in$ A, f(a) $\in$ B.

uncountable set: a set that does not have a one-to-one correspondence with **N**.

References

Church, A. (1936). "A Note on the Entscheidungs Problem," *Journal of Symbolic Logic*, vol. 58, pp. 345-363.

Davis, M. and Weyuker, M. (1983). *Computability, Complexity and Languages.* Academic Press: New York, New York.

Maurer, S.B. and Ralston, A. (1991). *Discrete Algorithmic Mathematics.* Addison-Wesley Publishing Company: New York, New York.

Myhill, J. (1972). "What is a Real Number?" *American Mathematical Monthly,* 79(7), pp. 748-754.

Rogers, H. Jr. (1987). *Theory of Recursive Functions and Effective Computability.* MIT Press: Cambridge, Mass.

Sudkamp, T. (1989). *Languages and Automata.* Addison-Wesley Publishing Company: New York, New York.

Linearity of Equations, Graphs, and Functions in 2-D
≈ or ≈
The Little Truths We Sometimes Forget

by

Ronald I. Frank
Department of Computer Science

Last month the challenge was posed to find the error in this table:

This is false.

	EQUATIONS	GRAPHS	FUNCTIONS
EQUATIONS	----	The equation representation of a linear graph is the form: *y = mx + b*.	The equation representation of a linear function is the form: *y = mx + b*.
GRAPHS	The graph representation of a linear equation is a straight line.	----	The graph representation of a linear function is a straight line.
FUNCTIONS	The function representation of a linear equation is the form: *y(x) = mx + b*.	The function representation of a linear graph is the form: *y(x) = mx + b*.	----

The definition of a linear function y(x) is:

HOMOGENEITY: $y(kx) = k\,y(x)$ LINEARITY: $y(x + u) = y(x) + y(u)$

<u>Note: The linear equation $y(x) = mx + b$ does not define a linear function!!</u>

HOMOGENEITY: $y(kx) = m(kx) + b \neq k[mx + b] = m(kx) + kb$

LINEARITY: $y(x+u) = m(x+u) + b = [mx + b] + [mu] \neq [mx + b] + [mu + b]$

[The "b" term is sometimes called the "inhomogeneous" term, which it certainly is.]