ADAPTING COMPUTATIONAL DATA STRUCTURES TECHNOLOGY TO REASON ABOUT INFINITY

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

Natalie Hammerman
Department of Mathematics and Computer Science
Molloy College
1000 Hempstead Avenue
Rockville Centre, New York 11571
nhammerman@molloy.edu

ABSTRACT

The NCTM [6] curriculum states that students should be able to "compare and contrast the real number system and its various subsystems with regard to their structural characteristics." In evaluating overall conformity to the 1989 standard, NCTM [11] requires that "teachers must value and encourage the use of a variety of tools rather than placing excessive emphasis on conventional mathematical symbols." Finally, the Educational Testing Service in PRAXIS II [2] "assess(es) the subject matter knowledge necessary for a beginning teacher of secondary school mathematics" to demonstrate competence in "the concept of countability as related to infinite sets" and this test "conforms to the NCTM standards" of 1989 and 1991.

In compliance with the above standards and goals, this paper presents to educators a deeper understanding of the concept of countability and as to which sets are countable and which are not. Cantor [3] introduced the diagonalization method to determine whether an infinite set of real numbers is uncountable. This paper presents a template for standardizing diagonalization type proofs by borrowing notions of applied software engineering technology to "store" infinite sets. This template is applied to a number of specific examples to illustrate the proper application of the diagonalization argument.

With the abstract use of computational data structures for describing infinite 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

In 1989, the National Council of Teachers of Mathematics (NCTM) put forth an extensive standard for mathematical curricula and education in elementary, middle-school, and secondary (pre-college) courses. The end goal of this mathematics educational process is to "provide students with opportunities to acquire the mathematical knowledge, skills, and modes of thought needed for daily life and effective citizenship, to prepare students for occupations that do not require formal study after graduation, and to prepare students for postsecondary education, particularly college" [7].

Amongst the many topics covered, the standard expected that "students have experiences with the concepts and methods of discrete mathematics," which is defined there as "the study of mathematical properties of sets and systems that have a countable number of elements." [8] In contrast with the countable number systems, "the mathematics curriculum should include the study of mathematical structure so that all students can compare and contrast the real number system and its various subsystems with regard to their structural characteristics" [9].

In 2000, the NCTM updated and elaborated on the standard. "Whereas middle-grades students should have been introduced to irrational numbers, high school students should develop an understanding of the system of real numbers. They should understand that given an origin and a unit of measure, every point on a line corresponds to a real number and vice versa. They should understand that irrational numbers can only be approximated by fractions or by terminating or repeating decimals. They should understand the difference between rational and irrational numbers. Their understanding of irrational numbers needs to extend beyond π and $\sqrt{2}$ " [12].

The formulation of the real number system and the explanation as to why irrational numbers cannot be exactly computed intrigues students due to the seeming never-ending computational "power" that computer systems make available at the touch of a button or click of a mouse. However, it should be explained to the student that the fundamental difference between the natural number system and the real number system, between countable sets and uncountable sets, differentiates that which can be represented in and calculated by a computing device and between those number systems that cannot. This paper incorporates basic computational data structures to elucidate this difference. By expressing these mathematical concepts within the context of computational structures, the discussion in this paper will be consistent with the NCTM document in that "computers, courseware, and manipulative materials are ... used in instruction" [10].

Computability theory models the computational process with mathematical and logical foundations. The basis for a computational model is the Church-Turing thesis [4, 16] that 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.

In 1895 Cantor [3] introduced the diagonalization method to determine whether an infinite set is uncountable. An infinite set is uncountable if there does not exist a 1-1 correspondence between the set and a subset of the natural numbers. His original proof was applied to the set of (total) functions over N, the set of natural numbers. (Note: This paper includes zero as a member of N, based on the axioms of Peano Arithmetic [17].) Subsequently, the diagonalization method was applied to different domains, which are not sets of functions. (For example, Atallah and Fox [1] describe diagonalization as "a proof technique for showing that a given language does not belong to a given complexity class.") This resulted in an unclear presentation of how to extend the technique to further cases. It has reached a point where the notion of "diagonal" has 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 set domains. 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 template proof and to identify the critical points necessary for a valid proof. By adapting data structure concepts for describing infinite set constructs, we show within the template proof that countably infinite sets do not provide enough information for a contradiction in the diagonalization argument. The template proof thus provides an important educational tool for clarifying difficult concepts in both mathematical and computability theory.

In the next section, set constructs are analyzed. A proven 1-1 correspondence between N and the set of finite subsets of N suggests a computer science implementation (data structure) for these sets that is extended to provide a similar (theoretical) representation for infinite sets. A template proof is then constructed in Section 3 for standardizing diagonalization arguments based on Cantor's original proof. Examples are provided to indicate the care needed in specifying a proper function within the diagonalization argument in order to yield a valid proof. With this template, the notion of the "diagonal" is clearly defined. Section 4 provides a mathematical formalization to explain why diagonalization arguments do 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 that can invalidate a diagonalization argument; these flaws are elusive to those learning computability concepts. Based on the above analysis, a characterization of countable and uncountable sets is discussed (section 5) 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 N will be considered. Let set S be an unordered collection of elements. Associated with each set is its membership function,

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

While the elements of sets are unordered, theorems about their cardinality will require an imposed order by putting the elements in a 1-1 correspondence with a subset of the natural numbers.

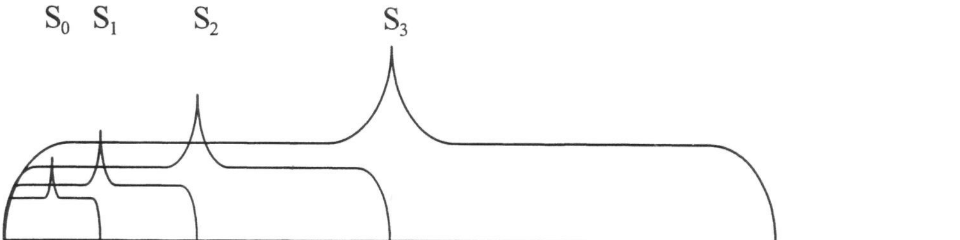
Consider the finite subsets of N. In this context it is necessary to determine the elements of E_j, the j^{th} finite subset of N. The binary representation of $j = b_k b_{k-1} \cdots b_2 b_1 b_0$. The bitstring data structure permits for the efficient retrieval/storage of a subset of N. Here, $M(i, E_j) = b_i$, $0 \le i \le k$, and $M(i, E_j) = 0$, $i > k$. A chart listing some of these elements is in Figure 1. A more elaborate example is presented next.

Finite Subset of N	Elements in E_i	BIT Representation $k \ldots 43210$	n in Base 10
E_0	$\{\ \}$	$0 \ldots 00000$	0
E_1	$\{0\}$	$0 \ldots 00001$	1
E_2	$\{1\}$	$0 \ldots 00010$	2
E_3	$\{0, 1\}$	$0 \ldots 00011$	3
E_4	$\{2\}$	$0 \ldots 00100$	4
E_5	$\{0, 2\}$	$0 \ldots 00101$	5
E_6	$\{1, 2\}$	$0 \ldots 00110$	6
E_7	$\{0, 1, 2\}$	$0 \ldots 00111$	7

Figure 1. The bitstring implementation of E_j, the j^{th} finite subset of N, is the binary representation of j.

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; in fact, the set will be E_{22}.

This example indicates that a natural number j encodes a finite subset of N embedded in the binary representation of j. The elements of this subset are precisely the column indices where the digit 1 appears in the binary representation of j. An extension of this bitstring construct allows for a representation of the infinite subsets of N as well; however, in this case there is no finite k as a length of the bitstring representation of the general subset and such that for $i > k$, $M\left(i, S_j\right)=0$.



{ }	{0}	{1}	{0,1}	{2}	{0,2}	{1,2}	{0,1,2}	...	$\{k: b_k=1\}$

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

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

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

$$S_3 = \{E_0, E_1, E_2, E_3, E_4, E_5, E_6, E_7\} = \left\{\begin{matrix} E_0, E_1, E_2, E_3, E_0 \cup \{2\}, \\ E_1 \cup \{2\}, E_2 \cup \{2\}, E_3 \cup \{2\} \end{matrix}\right\}.$$

Figure 2. Visual presentation of the recursive enumeration of sets (S_n) maintaining the finite subsets (E_j) of N.

An interesting pattern emerges from the table in Figure 1 (above). The empty set is appropriately the initial subset of N. Then, the set containing a new element, {0} is unioned with all subsets constructed up to this point (here, just the empty set so {0} remains). Then, {1} is unioned with the empty set and the subset {0} to yield {1} and {0, 1}, the next two subsets in the table. Finally, for this table, the next singleton subset {2} is unioned with the four previous subsets obtained, yielding

{2},{0, 2},{1, 2},{0, 1, 2} as in the table of Figure 1. The significance of this pattern is that computable functions over countable sets are called recursive [14]. Here, the above description using bitstrings is not only constructive but also recursive, as now described.

The recursive enumeration of sets (S_n) maintaining the finite subsets (E_j) of N can be formulated as follows (see Figure 2 for a visual representation of this):

Base case:

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

Recursive step:

$$S_{n+1} = S_n \cup \left\{E_j \cup \{n\} \mid E_j \in S_n, j = 0, 1, 2, \ldots, |S_n| - 1\right\},$$

$$= \left\{E_j \mid j = 0, 1, 2 |S_n| - 1\right\},$$

where $|S_n|$ is the number of elements in set S_n.

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

The above construction guarantees the 1-1 correspondence between N and the finite subsets of N. The significance of this in our discussion is threefold. First, the bitstring data structure actually provides a practical implementation of finite sets and can be extended (theoretically) to "implement" infinitely countable sets. Second, this emphasizes the nature of a 1-1 correspondence between an infinite set and N in that any infinite set where an individual element can be uniquely described (encoded) in a binary string where the last 1 in the encoding appears in a finite position (as above) is countable by the identical construction above and the fact that subsets of countable sets are countable (proof outlined in next paragraph). This is important when trying to categorize the cardinality of an infinite set. Third, the finite subsets of N are countable (as opposed to the power set of N, as in Section 3).

For completion, an informal proof that subsets of countably infinite sets are countable is as follows. Consider $S_1 = \{a_i\} \subseteq S_2 = \{b_j\}$, a countably infinite set. One can effectively count the elements of S_1 by a modified count of S_2. Since S_2 is countable, there is a 1-1 correspondence between b_j and the natural numbers so that a natural number, which will be represented by the index j, can uniquely identify each b_j. A second counter variable i will keep the actual count of the elements a_i in S_1. The idea is simply to list the elements of S_2 by counting through the natural numbers, retrieving the element b_j of S_2 that corresponds to the particular natural number just counted, and testing for membership in S_1. If that element is in S_2, then, i is incremented, thus "counting" the elements of S_1. A formal proof of this is given in Orr [13].

The above correspondences were described in this section based on an effective enumeration of all the elements of a countable set by the use of the bitstring implementation for the set data structure (adapted to handle infinite sets). In the next section, diagonalization analysis of uncountable sets [3] will involve a proof by contradiction that also requires an effective enumeration of all the elements of the given set. However, in the uncountable case, the diagonalization argument is used to prove the nonexistence of such an enumeration. This paradoxical use of the effective enumeration at times can obfuscate the proper application of set constructs. Thus, Section 3 will provide a standardized template proof of diagonalization arguments to enable proper applications of diagonalization techniques to infinite sets.

3. STANDARDIZATION OF DIAGONALIZATION PROOFS

Cantor [3] introduced the diagonalization method to determine the uncountability of infinite sets. Consider the set F of functions over the natural numbers, N. Assume that F is countable. Then, there exists a 1-1 correspondence between $F=\{f_i \mid f_i : N \to N\}$ and N. Construct f_{new} such that $f_{\text{new}}(i) \neq f_i(i)$ and $f_{\text{new}} \in F$. Cantor used $f_{\text{new}}(i) = f_i(i)+1$. Since $f_i(i) \in \mathrm{N}$ is uniquely defined, the same is true for $f_{\text{new}}(i)$ and hence, f_{new} is a total function on N. But, then $f_{\text{new}} \in F$; hence, $f_{\text{new}} = f_k$ for some k by virtue of the 1-1 correspondence between F and N. In particular, $f_{\text{new}}(k) = f_k(k)$. However, $f_{\text{new}}(k) \neq f_k(k)$ by the construction of f_{new} This contradiction indicates that no natural number k can be found for $f_{\text{new}} \in F$. Since there exists an $f_{\text{new}}($ F that cannot be counted, F is not countable.

	0	1	2	$\ldots$	i	$\ldots$
f_0	$f_0(0)$	$f_0(1)$	$f_0(2)$	$\cdots$	$f_0(i)$	$\cdots$
f_1	$f_1(0)$	$f_1(1)$	$f_1(2)$	$\cdots$	$f_1(i)$	$\cdots$
f_2	$f_2(0)$	$f_2(1)$	$f_2(2)$	$\cdots$	$f_2(i)$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
f_i	$f_i(0)$	$f_i(1)$	$f_i(2)$	$\cdots$	$f_i(i)$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Figure 3. Lookup tables (step 5 below) for the outputs of the characterization functions with the elements of N as inputs.

Having proved that this set F is not countable, a standard template consisting of ten steps for applying the diagonalization argument to any uncountable set now emerges:

1) Theorem. An (uncountable) infinite set T with elements containing property P is not countable.
2) Proof by contradiction. Assume that set T is countable.
3) By definition of a countable set, there exists a 1-1 correspondence between the elements of T and N.
4) Construct F, the set of total characterizing functions distinguishing each element of T. This associates an index i to each characterizing function $f_i \in F$ corresponding to some $t_i \in T$.
5) Construct lookup tables (the rows of Figure 3) for the outputs of the characterization functions with the elements of N as inputs. This creates a two-dimensional matrix with the $f_i \in F$ on the vertical and the natural numbers $n \in N$ on the horizontal; the outputs $f_i(n)$ are stored in the matrix cells.
6) Design $f_{\text{new}} \in F$ such that $f_{\text{new}}(i) \notin f_i(i)$, and characterizes some $t \in T$ with property P. For any i, $f_{\text{new}}(i)$ differs from the diagonal element of row i in the matrix of lookup tables.
7) The position (row) k of f_{new} in the indexed list of F can be obtained by the 1-1 correspondence provided for in step 4.
8) Compute $f_{\text{new}}(k)$. This equals $f_k(k)$ because of its position on the list of elements in F. However, $f_{\text{new}}(k) \neq f_k(k)$ by the diagonalization construction in step 6. Thus, an $f_{\text{new}} \in F$ has been found which cannot be counted in the indexed list.
9) The contradiction indicates that a 1-1 correspondence cannot exist between the elements of F and N.
10) Conclusion: F is not countable; therefore T is not countable.

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

D1) to define a property P which allows for an element to be a member of a set, say T; (in the above proof, P=(total) function over N)

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

D3) to define $q_{new} \in T$ 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 $\forall i$, $f_{new}(i) \in$ N; also, the inequality condition was provided for by f_{new}).

Referring back to the general template diagonalization proof above, criterion D1 is stated in step 1; criterion D2 is provided for in step 4; criterion D3 is defined in step 6. These three steps are the only parts of the diagonalization proof template that depend on the specific application domain. This template clearly demonstrates what this method incorporates from the "diagonal." The diagonal of the lookup table of characterizing functions (step 4) is utilized to define f_{new}, the function that will allow for the contradiction (step 6). Thus, to prove that a set T containing elements with property P is not countable, prove that the associated set F of total characterization functions, distinguishing each element of T, is not countable.

As mentioned at the end of the previous section, the power set of N is not countable. To use this template to show that the power set of N is not countable, the three definitions above (D1-D3) have to be stated. First, define P as "a subset of N" (criterion D1). Then, the characterizing function f_i is the infinite bitstring implementation of the membership function (criterion D2). Finally, the diagonalizing function $f_{new}(i)$ is simply $\sim f_i(i)$ (read "NOT $f_i(i)$"); the elements of q_{new} are precisely those j such that $f_{new}(j)=1$ (criterion 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). Criterion 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 representation of the real numbers between 0 and 1.

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 (criteria D2 and D3) are sensitive points in that they provide fertile ground for creating an invalid proof. Specifically, by enumerating the f_i (criterion D2), an order is imposed on the q_i. The elements of T must have a 1-1 correspondence with N implying that no element of T is included more than once nor precluded from the list by the imposed ordering. For example, for definition D1, let P be "a real number between 0 and 1." Since this subset of numbers is uncountable, then so is R, the set of all real numbers. But, since fractions with terminating decimals have two representations, q_{new} could possibly be on the list in an alternative form.

While it is relatively simple to make sure that for $i \in$ N, $f_{\text{new}}(i) \neq f_i(i)$, guaranteeing that the corresponding q_{new} has property P may be more difficult (criterion D3). For example, to prove that the monotonically increasing functions are uncountable, let T be the set of those functions (criteria D1 and D2). After setting up the matrix as in figure 2, to ensure that $f_{\text{new}}(i) \neq f_i(i)$, a first attempt for f_{new} might be $f_{\text{new}}(i) = f_i(i)+1$; but this does not guarantee that f_{new} is monotonically increasing. Therefore, this results in an invalid proof. Defining the diagonalizing function [15]

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

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

The previous sections differentiated between the set of finite subsets of N which are countable, and P(N), the set of all subsets (power set) of N which is not. Yet, theoretically adapting 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 set of all 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" (criterion D1). For step 3, set up the one-to-one correspondence and the corresponding bitstring implementation shown before for the finite subsets of N (criterion 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 $\rightarrow$ 01100000... in the row of the lookup matrix for f_6. This row represents $M(j, q_6), j \in$ N.

Consider step 6 where q_{new} is defined by B_0B_1 ... where $B_i = f_{\text{new}}(i)$. Since $f_i(j)$ can only be 0 or 1 and $f_{\text{new}}(i) \neq f_i(i)$, $B_i = f_{\text{new}}(i) = \sim f_i(i)$ 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 $b_0b_1 \ldots b_d \ldots$ is the infinite bitstring characterization of finite subset q_d of 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, $b_i=0$, $\forall i \in$N. For all $d>0$, $2^k \leq d < 2^{k+1}$ for some $k \in$ 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 $d=b_k b_{k-1} \ldots b_0$. A simple argument can be developed to show that $k<2^k$, $k \in$N; therefore, $2^k \leq d$ implies that $k<d$. It then follows that the associated infinite bitstring representation for d, represented by $b_0 b_1 \ldots b_{k-1} b_k \ldots b_i \ldots$ in the lookup table, contains zeroes for all other digits b_i, $i>k$. Therefore $\forall i \geq d$, $b_i=0$. □

Hence $f_{\text{new}}=111 \ldots$, and therefore q_{new} is not a finite subset of N.

5. CONCLUSION

This paper adapts a computer science construct as a means of explaining mathematical theoretical concepts. In particular, the bitstring data structure allows for the definition of the membership function for (in)finite 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.

Set T is uncountable when an infinite amount of data is mandatory to define membership for elements of T. Proofwise this means that the template proof 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 that 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 N must have an infinite sequence of bits to define the membership function. Consequently, these two sets, R and P(N), are uncountable and hence, procedures for computation over these sets are not effectively computable. See Myhill [5] who shows that, in fact, any radix based representation (base 2 in this paper) of real numbers will yield simple computations that to compute even the first digit of the result would require an inspection of an infinite amount of the operand's digits.

Set T with property P is countable when all (but a finite number of) $t \in T$ can be represented by a finite amount of data to define the elements that have property P. Proof wise this means a 1-1 correspondence does exist between the elements of T and 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 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.

REFERENCES

1. Atallah, M.J. and Fox, S., *Algorithms and Theory of Computation Handbook,* CRC Press, Boca Raton, FL, pp. 21-7 (1998).
2. Brunsman, Bethany A. and Carlson, Robert E., Jr., *Annual Report on the Praxis and SSAT Examinations in English, Mathematics, and Social Science, December 1995 - June 1998*, California Commission on Teacher Credentialing: Sacramento, CA. Accessed from http://www.ctc.ca.gov/aboutctc/agendas/may_1999/perf/perf1appendices.html on August 24, 2003, 8:57pm (1999).
3. Cantor, G., *Contributions to the Foundations of the Theory of Transfinite Numbers,* Reprinted by Dover, 1947: New York, NY (1895).
4. Church, A., "A Note on the Entscheidungs Problem," *Journal of Symbolic Logic,* Vol. 1, pp. 40-41 (1936).
5. Myhill, J., "What is a Real Number?", *American Mathematical Monthly*, Vol. 79, No. 7, pp. 748-754 (1972)..
6. National Council of Teachers of Mathematics, *Principles and Standards for School Mathematics: Curriculum and Evaluation of GRADES 9-12: Standard 14 - Mathematical Structure*, National Council of Teachers of Mathematics: Reston, VA. Accessed from http://standards.nctm.org/Previous/CurrEvStds/9-12s14.htm on August 24, 2003, 5:31pm (1989a).
7. National Council of Teachers of Mathematics, *Principles and Standards for School Mathematics: Curriculum and Evaluation of Grades 9-12: Curriculum Standards For Grades 9-12*. National Council of Teachers of Mathematics: Reston, VA. Accessed from http://standards.nctm.org/Previous/CurrEvStds/currstand9-12.htm on August 25, 2003, 10:08am (1989b).
8. National Council of Teachers of Mathematics, *Principles and Standards for School Mathematics: Curriculum and Evaluation of GRADES 9-12: Standard 12 - Discrete Mathematics,* National Council of Teachers of Mathematics: Reston, VA. Accessed from http://standards.nctm.org/Previous/CurrEvStds/9-12s12.htm on August 25, 2003, 10:17am (1989c).
9. National Council of Teachers of Mathematics, *Principles and Standards for School Mathematics: Curriculum and Evaluation of Grades 9-12: Standard 14 - Mathematical Structure*. National Council of Teachers of Mathematics: Reston, VA. Accessed from http://standards.nctm.org/Previous/CurrEvStds/9-12s14.htm on August 25, 2003, 10:23am (1989d).

10. National Council of Teachers of Mathematics, *Principles and Standards for School Mathematics: Curriculum and Evaluation of Grades 9-12: Evaluation: Standard 12 - Curriculum and Instructional Resources,* National Council of Teachers of Mathematics: Reston, VA. Accessed from http://standards.nctm.org/Previous/CurrEvStds/evals12.htm on August 25, 2003, 11:03am (1989e).
11. National Council of Teachers of Mathematics, *Principles and Standards for School Mathematics: Professional Standards for Teaching: Standard 4 - Tools for Enhancing Discourse*, National Council of Teachers of Mathematics: Reston, VA. Accessed from http://standards.nctm.org/Previous/ProfStds/TeachMath4.htm on August 24, 2003, 5:44pm (1991).
12. National Council of Teachers of Mathematics. "Number and Operations Standard for Grades 9-12," in Chapter 7 of *Principles and Standards for School Mathematics,* National Council of Teachers of Mathematics: Reston, VA. Accessed from http://standards.nctm.org/document/chapter7/numb.htm on August 25, 2003, 10:56am (2000).
13. Orr, John Lindsay, *Analysis WebNotes*. Accessed from http://www.math.unl.edu/~webnotes/classes/classAppA/lemA1.htm on August 25, 2003, 12:57pm (1996).
14. Rogers, H., Jr., *Theory of Recursive Functions and Effective Computability,* MIT Press, Cambridge, MA (1987).
15. Sudkamp, T., *Languages and Automata*, Addison-Wesley Publishing Company, New York, NY (1997).
16. Turing, A. M., "On Computable Numbers, with an Application to the Entscheidungs problem", *Proceedings of the London Mathematical Society, Series 2*, Vol. 42, pp. 230-265; Corrections (1937). Ibid. Vol 43, pp. 544-546 (1936).
17. Weisstein, Eric, *World of Mathematics*, CRC Press: Boca Raton, FL. Accessed from http://mathworld.wolfram.com/PeanosAxioms.html on August 26, 2003, 10:16pm (1999).