Symbolic Set Theory for Scientists: A Computer Science Perspective
Paula Medina version

Maurice Karnaugh, Paulina Medina, Charles C. Tappert and Avery Leider
Seidenberg School of Computer Science and Information Systems, Pace University
Pleasantville, NY 10570, USA
Email: symbolicsets@yahoo.com, {paula.hernandezmedina, ctappert, aleider}@pace.edu

Abstract*—Computer scientists perceive that they have limited needs of set theory. Set theory is used in problem complexity descriptions that characterize algorithms and that is the end of it. However, the computing environment has changed fundamentally in two ways. One, techniques to compute with extremely large numbers - possibly infinite numbers - are revealing immense value in artificial intelligence, increasing the demand to be ever better at this. Two, computational ability is passing Moore's Law's constraints, as Moore's law becomes trivial once we pass the matter-to-energy boundary of quantum computing. To adapt to these changes, we present here that symbolic set theory gives us tools to expand our understanding and enable ideas of greater experience in computer science. Classical axiomatic set theory is based upon the Zermelo-Fraenkel axioms (ZF) or ZF plus the axiom of choice (ZFC). ZF and ZFC theory creates unimaginably large sets (ULS), populated by mostly undefinable elements. The mathematical "real line" is such a set. This is useless. Computer scientists need symbolic models for real phenomena. Symbolic sets are well defined sets of symbolic elements which we construct with suitably constrained axioms. The definite (pronounced de-fine'-ate) real line is such a set. It contains just the well defined elements of the real line and it is sufficient for numerical scientific computation. This paper takes a constructive point of view that makes it less abstract and more accessible and is influenced by the subsequent development of computer science beyond the 19th century. To use symbolic set theory, the computer scientist needs an acquaintance with discrete mathematics and with convergent sequences, and we shall define here the other concepts.

***Index Terms*—ZF, ZFC**

I. Introduction

Progress in mathematics has always been motivated by practical concerns. However, with the study of logic and formalization of deductive proofs these circumstances begun to change. Logical foundations for mathematics are presently not well established but a large body of interesting mathematical theory has resulted. An important result of this work has been the divergence between mathematical set theory and the applications of mathematics to scientific theory. Much of this divergence can be ascribed to differences in method as it applies to the selection of axioms. The mathematical logicist view is that one can be free to introduce any axiom that is sufficiently interesting and which (one believes) cannot be proved or disproved by the preexisting axiom set. Having done that, one can then proceed to deduce some of its consequences, however counter-intuitive they may be. Logicism has introduced some new philosophic problems. For example, in the standard form of modern set theory, sets of elements can be so large that most of their elements cannot be defined by any text. In addition, the axiom of choice (i.e., ZFC) tells us that these undeniable elements can be well ordered (a specific type of order which will be explained in this paper). These counter-intuitive things need not be dismissed forever. In quantum computing the biggest current challenges can be resolved by just understanding set theory.

The scientific method is much more constrained. This takes some explaining. Up to this time, the goal of scientific theory has been to develop symbolic theories that adequately describe the observable properties of real phenomena. These properties are determined by observation and, when possible, by conducting experiments. If one chooses axioms that are too simple or unnecessarily broad in scope, one increases the chance that in the near future experiments will reveal some discrepancies of theory with observation. A common view among scientists is that a theory should be as simple as possible but not more so. Simplicity alone is an inadequate criterion. One might argue that Newtonian physics is simpler than general relativity but general relativity is more correct in predicting certain critical observations. One's intuition is also not invariably a correct guide. Einstein, for example, found it difficult to accept quantum theory because he believed that "God does not play at dice with the world." Nevertheless, quantum theory is the most accurate model for many physical observations. To sum up these simple remarks, nature is the arbiter of the selection of natural laws - not simplicity, not intuition.

II. Methodology

Internal references, if used, will be decimals (eg. [3.2] points to the second numbered item in section 3). [3.0] points to section 3 but not to any numbered item in it. Two textbooks are referenced [1] and [2], for those who desire to study ZFC [ZermeloFraenkel set theory]. The first one is brief but quite abstract. The second one is also abstract but much more comprehensive and well designed for lengthy self study. Our approach is much less abstract than the textbooks'.

This printed tutorial is in a language we call a meta-language. The alphabet of this meta-language includes three fonts of the English alphabet and a large collection of punctuation, formatting and logical and mathematical characters. This is a large but finite alphabet. The existence of a sufficiently expressive meta-language is essential for the practice of philosophy, science, mathematics and for this paper. It is usually taken for granted but we must be careful to observe its limitations.

Our method is largely constructive, based upon recursive generators. The presentation may be described as informal, minimally abstract and strongly influenced by computer science.

III. Literature Review

This work is an update from an earlier draft, posted online [3]. The classic set theory that uses ZF and ZFC (the C is for Choice) comes from the work of Ernst Zermelo and Abraham Fraenkel. They followed the lead of Georg Ferdinand Ludwig Philipp Cantor.

References

[1] P. R. Halmos, "Relations," in *Naive Set Theory*. Springer, 1974, pp. 26–29.

[2] D. Goldrei, *Classic Set Theory: For Guided Independent Study*. Routledge, 2017.

[3] M. KARNAUGH, "Symbolic sets and the real line."

IV. Classes

A collection of different identifiable elements that usually have some common properties is called a class. When we speak of classes, we usually have to assume that all the elements contained in a class are contained in a well defined largest class that is called the universe of discourse. The collection of elements in a class is called the extension of a class. In the extension, the elements are unique and counted only once. This extension is a very important property, since other properties like cardinality (number of elements in a class) can be obtained from the extension.

A. Creating infinite classes of elements

In order to create classes, for example an alphabet, we must consider two main ideas, concatenation and segmentation. Concatenation is the formation of a new number by concatenating its numerals. For example, we can concatenate a 1 to a 1 to produce the string 11. Concatenating another 1 will result in 111 and so on. Segmentation is used to avoid losing information. For example, if we have the string 1111 there is no way to know if we concatenated 1 to 111 or 11 to 11. This arises the segmentation problem and results in loss of information. To solve this, we need to use at least two primal objects like 0 and 1 so when we concatenate we don't lose information. It is possible to create a segmented class of infinitely many characters by a process called simple finite recursion. As we know, recursion begins with a foundation element or with a finite sequence of elements called foundation sequence. If we begin with a foundation element 0, the successor will be 01. Let x be any element, then the right concatenation of a 1 will be x1, this last element being the successor of x. If we want to create some algorithm to do this we can do something like *suc(x) = concat¡x,1¿*.If we keep repeating this process then we have a simple finite recursion where each element is generated after a finite number of recursions even if there is no last element. A sequence of distinct elements with a first element but no last element will be called an infinite sequence. In the previous example the resulting class is called segmented unary class, SU, and the primal element 0 indicates the start of a new character. However, if instead of doing the right concatenation of a 1 we start doing the left, the result 0, 10, 110... will be called unary segmented class, US. If we take a look at the class with the binary form of the natural numbers we will see that it is not segmented (0, 1, 01, 11, 011). In order to fix this we would need to include a third primal object as delimeter. Now that finite recursion has been defined, we must add that to create an infinite class of elements using finite recursion, two necessary conditions must be met. The first one is that the successor function has to be inherited by each element, so if successor is applicable to x, it must also be applicable to the successor of x element. The second condition is that each new element created must be unique. There must be no loop in the character sequence. Therefore every sequence created is infinite and has the transitive property.

Creating an infinite class of characters from a finite class of primal characters is an important mathematical fact. We can now introduced the concept of infinity and the concept of completion. Completion is no small matter. Although each element of a simple finite recursion is created after a finite number of successor operations, we must axiomatically assume that the completion of the recursion exists in order to assume that the class exists.

B. Order

When we generate infinite classes by finite recursion, we are giving this classes an additional property, order. For every character that we create, there is a unique successor. If we choose any two created characters, one must have been created before the other. This last idea gives the property of strict ordering where two distinct characters x, y are either x¡y or y¡x, but never both. The order of the natural numbers will be called NN ordering which is an important type of well ordering.

C. Cardinality

Now that we can create a class and we have defined order, we need to create a method for quantifying the size of different classes to compare them. Consider a non-empty class X of distinct elements. We do not know how many elements X contains so we employ the following procedure, which we shall call countable reification.

1) The foundation family. We create two classes: the initial residue class, XR_0 = X and the initial cumulative class, CX_0 = (), which is empty.
2) The successor operation is *count*¡RX_k, CX_k¿, where k is a representation of any natural number. This is a

function whose argument is two natural number indexed classes which returns a family of two modified classes, RX_{k+1} and CX_{k+1}. The modification of the residue class, RX_k is to reify one of its elements (it does not matter which one) and name it x_{k+1} and remove it from RX_k so that RX_{k+1} = *difference*¡ RX_k, x_{k+1}¿. The modification of the cumulative class is CX_{k+1} = *union*(CX_k, x_{k+1}).
3) The recursion. The count function is applied recursively until, for some n, the residue class RX_n is empty.

If this recursion stops for some natural number *n*, then the class X is finite and its cardinality, the number of its elements, is *n*.

If this recursion doesn't stop, then the class X has infinite cardinality. We can't specify the cardinality of X but we can say that X is an infinite sequence that has a first element but no last element. We can also say that CX_k has a clear one-to-one relationship with a subclass of the elements of X and it also has a clear one-to-one relationship with the class of natural numbers. It follows that every infinite class includes a subclass that has a one-to-one relationship with the natural numbers. The cardinality of the natural number is denumerable.

Two classes with equal cardinality (*card*(X) = *card*(Y)) are said to be equipollent, equinumerous or to have equal power. They are represented as $X \approx Y$, and this relation is reflexive, symmetric and transitive. All classes that can be generated by means of simple finite recursions or by countable reification form an equivalence family with respect to cardinality. The cardinality of this equivalence family has been given the conventional name, $\aleph_0$ (aleph null). All classes with $\aleph_0$ cardinality are denumerable and all classes that are denumerable or finite are countable. Consider a denumerable family of classes, each of which contains a denumerable class of elements. We assume that no two of the classes have any element in common, that is, they are disjoint. Then the union of this classes is also denumerable.

D. Comparing infinite classes

How are we to compare the cardinalities of infinite classes? One reasonable approach is to extend what we know about finite classes but it must be done with care because, unlike finite classes, infinite classes can be in a 1:1 correspondence with some of their proper subclasses. For example, the positive integers can be in a 1:1 correspondence with the even positive integers. In that case the bijection is f(j) = 2j. It is also clear that there is a bijection between the natural numbers and the positive integers. In this case the bijection is f(j)=(j+1).

We need to follow some rules in order to compare infinite classes.

1) Two classes are equinumerous and have equal cardinality if and only if they have a 1:1 correspondence.
2) If a class B can be placed in a 1:1 correspondence with a subclass of a class C, then we say that C dominates B and write $B \preceq C$. This enables us to say that the cardinality of B is less than or equal to that of C. It follows that the cardinality of subclass S of a class X is no greater than that of X.
3) If C dominates B but B does not dominate C then we say that C strictly dominates B, $B \prec C$, and that the cardinality of C is strictly greater than the cardinality of B.
4) If two sets are equinumerous then each of them dominates the other.
5) If $B \preceq C$ and also $C \preceq B$ then it appears reasonable to equate their cardinalities, but in that case it should be possible to prove that there is a 1:1 relationship possible.

That mutual domination implies the existence of a 1:1 relation has been proved for all classes. Since every infinite class has a subclass of cardinality $\aleph_0$, this is the smallest infinite cardinality and every infinite class that is dominated by the class on natural numbers has cardinality $\aleph_0$. Every denumerable class can, by definition, be put in a 1:1 relationship with the natural numbers. Therefore, every denumerable class can be given NN ordering. Consequently, every denumerable class can be well ordered.

E. Three Important Classes

There are three main classes of which their cardinalities should be studied for the purpose of this paper. The natural numbers, the class of all finite binary strings, the class of all symbols in the metalanguage. As we have previously discussed, the class of natural numbers is generated by a simple finite recursion in which the *suc* function is implemented by the binary addition 1 to the content of the register. The class of all finite binary strings is also generated by a simple finite recursion in which the *suc* function is implemented by the addition of 1 to to the content of the register in number base n unless the register contains a string of all n 1's. In that case the register is given a string of all 0's that is one bit longer. This procedure generates all strings of length k ¿ 0 before going on to length k + 1. The cardinality of this class is the same as the cardinality of the natural numbers, $\aleph_0$. The cardinality of the class of all symbols follows from this. The finite alphabet of the metalanguage can be encoded by a class of finite binary strings. Therefore, the class of all symbols is dominated by the class of all finite binary strings and is less than or equal to $\aleph_0$. Since the class is infinite its cardinality must be $\aleph_0$.

One important theorem of $\aleph_0$ we can obtain from this three classes is the following:

The cardinality of a denumerable class of pairwise disjoint denumerable classes is denumerable. Every infinite class that is dominated by a denumerable class is also denumerable because that is the smallest infinite cardinality.

The explanation of this theorem comes from the idea that each denumerable class can be identified by a natural number 0,1,2,3..... so that the union of two denumerable disjoint classes that are both identified by a natural number, must be an ordered pair of natural numbers. Having explained this, it is clear to see that the cardinality of the union is no less than $\aleph_0$.

V. Sets

Sets are a collection of elements, similar to classes. However, sets are elements of a set theoretic universe and they can contain other sets as elements. There exist three main distinctions between sets and classes:

1) Any collection of elements within a universe is a class but, while sets are collections of elements, not every collection of elements can be the extension of a set.
2) Sets are elements but classes cannot be elements. Classes that cannot be elements are usually called proper classes. All of the classes we shall construct will be proper classes.
3) Because sets are elements, the extension of a set may include other sets

To construct sets, we are going to use a function called *wrap* which takes a regular class of elements (a,b,c) and transforms it into a set a,b,c. Its inverse is the function *unwrap* that takes a set as input and returns a class. There is also a function called *multiwrap*. If X is a regular class and all of its proper subclasses are also regular classes then *multiwrap*(X) is the class containing wrap(X) and the wraps of all of the proper subclasses of X as elements. For finite classes, the multiwrap is algorithmically computable.

A. Russell's Paradox

Philosopher and logician Bertrand Russell was troubled with a paradox in set theory. The set of all sets would have as a subset the set of all sets that do not contain themselves as elements. However, if we assume that this subset does not contain itself then it must; if we assume that it does contain itself then it must not. Clearly this presents an unavoidable contradiction also called a paradox . This problem makes it clear why the *wrap* function is needed. By limiting the domain of wrap to classes that are regular we can limit the sets of the universe to sets that are well founded and which avoid the paradox.

B. Ordering classes of symbols

We shall discuss universes (of discourse) of sets whose foundation classes are classes of symbols. In the recursive construction of these universes by successive classes, the names of these classes will be indexed by ordinal numbers. These ordinals will be a well ordered class of symbols. Therefore, it will be useful to discuss some ways of well ordering classes of symbols. All classes of symbols that are created by simple finite recursion are NN ordered by the order of generation and all NN ordered classes are well ordered. To lexicographically order any two finite strings of linearly ordered alphabetical characters, we begin with the leftmost character of each string. If these are not the same then the one with the lesser first character is the lesser of the two strings. If they are the same then we go one step to the right and compare the next pair of characters and so on until one of the following is true: either one of the two strings has ended in which case it is the lesser of the two or else the two strings have different characters. in some position, in which case the one with the lesser character in the first such position is the lesser of the two. If the two strings have the same character sequence and terminate at the same position then they are identical. Lexicographic ordering is not a well ordering for the class of all finite strings of an alphabet of two or more characters. The second type of ordering will be called canonical. To find the least element of a class of symbols in canonical order, we first select the subclass of shortest length, by which we mean the subclass of symbols having the fewest alphabetical characters. This subclass will always exist because the length is a natural number and the natural numbers are well ordered. Then we select the least of these by means of lexicographic ordering. Scanning the class of alphabetical characters at each position from left to right we will always find a least character, assuming a well ordered alphabet. The symbols being compared are all of the same finite length, k, so the process will end at position k. At that point, there will be one or more least symbols remaining. If there are more than one, they must be identical, which cannot occur in a true class.

VI. Ordinals and ordinality

The natural numbers can be used as ordinal indices to reify the generated classes in a finite recursive class generator (e.g., $C_0, C_1, C_2, \ldots$). This allows us to follow up the order in which the classes were created and to determine which of any family of classes has the least ordinal. Ordinals are well ordered classes, so every subclass of ordinals has a least element. When we use them to index the classes in a recursive generator, there is always a next ordinal available. The next ordinal is the least one of those that have not yet been used. In a finite recursion, the first limit class is the one corresponding to the union of all previous classes after a denumerable sequence of recursions. It is not a successor class but it contains all elements of the preceding classes. The customary symbol for this ordinal is (omega). In ZF theory, transfinite recursion is possible and it will be convenient to have transfinite ordinals to reason about it and to define the universe. We will limit out attention to classes with symbolic ordinals since we have limited our attention only to symbolic sets so we don't need transfinite ordinals beyond +1. Transfinite symbolic orbital exist. The least of them is which is the limit for any finite recursion and represents the least ordinal greater than any natural number. If k is any natural number, $+ k$ is the k^{th} ordinal that follows . The ordinal number of recursions in a generative procedure can be limited by specifying a limit ordinal to be the least upper bound (lub) to the permitted ordinals.

VII. Cantor's ladder of infinities

The term "continuum", c, is a name frequently applied to the cardinal of the real numbers. It is also, due to the work of Georg Cantor on set theory, considered to be a larger cardinal than 0 (pronounced "aleph null"), which represents the cardinal of the natural numbers. In this section, we present one of Cantor's proofs of this and examine its applicability to the real numbers as they are used in arithmetic computation. In order to explain this proof we are going to start by creating a universe of sets using recursion. We know that the empty set exists, which will be the foundation for this recursion. Then we know that if a set X exists then its power set PS(X), is the set of all subsets. We also know that if X is a set in our recursion then the successor of X is its power set. Now suppose that the sets in our procedure are indexed by conventionally represented ordinals, so that the empty set is reified as X_0 and its successor as X_1, and so on. This has no end, but we know that if any X_k is in the set, its power set is also in the there. Now assume that we reach the first limit point with ordinal . All of the existing sets are finite and indexed by finitely generable ordinals. Now, we can apply the theorem of *wrap* to the class of all existing sets. These are all finite but the cardinality of the resulting class is $_0$ and the cardinality of its wrap is also $_0$. This is the first infinite set in our construction. We can then apply the recursion once more, from ordinal to the next limit point, $\cdot$ 2. It will be sufficient to assume that $\cdot$ 2 is the lub of the ordinals for this exercise. Now lets examine the cardinalities of the infinite sets we just created. To do that, we need Cantor's power set theorem that says that every set is strictly dominated by its power set. If X is a set and P S(X) is the set of all subsets of X then $X \prec PS(X)$. This is equivalent to saying *card*(X) ¡ *card*(P S(X)). This theorem has been proved to be true for finite and infinite sets. Let's return to our recursive generation process. The foundation set is the empty set, , which is of cardinality 0. The successor set is P S(). The symbol for this is obtained by wrapping a list of all subsets of the empty set in curly brackets but the empty set has only one subset, which is the empty set. Therefore, the power set of the empty set is , which is of cardinality 1. The successor of this set is , which has cardinality 2. We can repeat the recursive application of the power set creation endlessly. Each time we create a successor set with exponentially greater cardinality, but they are all of finite cardinality because if k is finite then 2^k is also finite. At the completion of this simple finite recursion we have created an infinite subclass of the finite binary strings. We then take the *wrap* of the union of all of the finite sets that have already been created. This is a set of denumerable cardinality. Its power set a set having the cardinality, c, of the continuum. Then we can continue to apply the power set successor function endlessly. This will create a ladder of ever increasing infinite cardinalities. By simple modifications of the axioms, some of the ideas that have been developed for set theory will be useful in creating a symbolic set theory, and we can construct the definate real line with only well defined elements of the real line.

VIII. Constructing Real Numbers

Now we are going to construct the real numbers without using the previous axioms. Starting with rational fractions (cardinality $\aleph_0$) where each element is represented in the form on an infinite nary string followed by a nary point using recursion on a long division algorithm. However, other infinite nary strings can be generated in other recursive ways. That is how we have identified many irrational numbers such as pi, e, irrational trigonometric numbers,and irrational algebraic numbers. Each of this and others can be said to be well defined and to exist if a text in the metalanguage exists which makes possible a procedure for generating the appropriate series of numerals. It is noteworthy that any nary string of numerals that can ever be well defined, will be a real number. Due to this previous conclusion, it is clear to see that the real numbers cannot exceed the cardinality of the class that define them, the metalanguage text, and therefore its cardinality cannot exceed $\aleph_0$.

IX. Universes having non-empty foundations.

To fully understand this section we must know the definition of rank. The rank of an element is the ordinal of the least cumulative class of which it is a member. Now assume that we have a non-empty countable foundation class of symbols. In set theory these symbols are called urelements because they are not created from the empty set. They can represent whatever we choose or they can be just symbols. The construction of a universe with this foundation treats them just as symbols. The cumulative classes will be indexed by ordinals. We employ well ordered, symbolic ordinals. The foundation class, CC_0, may or may not be given some particular order. We shall assume a finite linear order at this time. We shall deal with infinite foundations later. The constructive procedure is as follows. The classes constructed will be cumulative classes. Given any cumulative class of ordinal *k*, the class of its ordinal successor, k^+, is the union of two classes:

$$CC_{k^+} = \textit{union(multiwrap}(CC_K), (CC_0). \tag{1}$$

If CC_k is finite then CC_0 must also be finite and CC_{k^+} is finite. Cantor's theorem tells us that *multiwrap*(CC_k), which is the class containing all elements of ranks 1 through k^+ and which contains a dominant set and all of its subsets, has a greater cardinality than CC_k. We have already seen that, when a class is infinite, taking the union of this class with, a countable class, does not change its cardinality. If *multiwrap*(CC_k) is finite then so is CC_0, and the union will result in a greater finite cardinality. The family of cumulative classes in this universe is indexed by their ordinals. The successor classes contain all elements of rank equal to or less than their own ordinals. The limit classes contain the union of all classes of lower ordinality and we have seen that their cardinalities are strictly greater than the cardinalities of their predecessors.

If instead of staring with a countable foundation, we begin with a denumerable class of symbols the class indices can be given a least value of and a greatest value of the ordinal: lub or lub + 1. This would be consistent with transfinite constructions having a finite foundation class. When the foundation class is denumerable and its ordinal is and the ordinal lub is $\cdot$ 2, the universe contains cumulative classes having an infinite ladder of Cantor's increasing infinite cardinalities.

X. Shrinking the universe

We have constrained this presentation of set theory to sets that are symbolically founded and which have symbolic class ordinals. All cumulative classes of sets having cardinalities larger than $_0$ have transdenumerably more elements than we can define by any metalanguage texts and there are denumerably many such classes of sets having an infinite ladder of greater cardinalities, even in a universe generated by a finite foundation and trans-denumerable recursion up to ordinal $\cdot 2$. When we have constructed a denumerable class of symbols, even one application of the multiwrap function will give us a class of sets which has cardinality c, having trans-denumerably more elements than we can well define. What might be more desirable for scientific purposes is a universe containing only all of the elements that can be well defined. Such a universe could have only denumerable cardinality but it would contain every element that science could define, including irrational numbers.

XI. Symbolic sets

Symbolic sets universes contain only symbolic ur-elements and well defined sets whose elements are all symbols. We shall now modify the definitions related to the *wrap* and *multiwrap* functions so as to define universes of sets such that all sets are well defined and the elements of all sets are symbols. A class is ss-regular if and only if it is well defined and all of its elements are symbols. This is a significant restriction. All finite classes of symbols are regular but most infinite subclasses of an NN ordered class of symbols are not regular because they are not well defined. The well defined infinite subclasses are important but they are not symbols. Therefore, no class that contains the ss-*wrap* of one of them as an element is regular. This will limit the recursion. The ss-*wrap* of a class exists and is a set if and only if the class is ss-regular. It should be clear that *multiwrap*, applied to finite classes is not the same function as *multiwrap* applied to regular denumerable classes. For example, when applied to a finite class, it returns a larger finite class which is well defined and regular and it can be algorithmically constructed. None of these things are true when it is applied to a denumerable class. We need a new function when applying it to a denumerable class and it is sensible to change its name. We want to restrict *multiwrap* so as to avoid creating undefinable sets. The ss-*multiwrap o*f an ss-regular class, X, is a class whose only members are the wraps of all of the well defined subclasses of X. If we start the generation of a symbolic sets universe with an empty or finite foundation class then the generation will proceed as before up to the first limit point. The class at CC will contain all of the hereditarily finite sets and it will be ss-regular. Applying ss-multiwrap, the successor class at ordinal (+1) will not be regular and the recursion must terminate at that point. The universe will contain all sets that have been created, including the finite sets and the well defined infinite sets.

XII. Conclusion

Computer Scientist use Set Theory in many of their usual operations. They use sets to think about the data structures as well as using set operations to work with them. The relation between databases is seen as a relation over sets. Operations like Union, Intersection and Concatenation of strings are widely used in daily tasks and proofs are necessary to come up with reliable computer science theory. However, in this project, besides giving meaning to some of this basic set theory applications we have gone deeper into set theory. We have defined and explain the meaning of infinities, we have defined symbolic sets and we have pointed out how finite recursion is needed to create infinite classes. We believe that with the present constraints that computer scientist are encountering this paper explains and gives basic foundations to deal with this new changes. Understanding infinities, classes and sets should help every computer scientist to deal with the amount of data that we are currently dealing with.