International Journal of Mathematics and Consciousness

Volume 5, Number 1
2019-2020

Editorial Introduction to the Journal
Tony Nader, MD, PhD, MARR

Proving That Wholeness Is Indestructible
Paul Corazza, PhD

Natural Language Processing with Semantics and Logic: A Necessity Strongly Supported by Maharishi Vedic Science
Emdad Khan, PhD

Maharishi International University
Fairfield, Iowa

International Journal of Mathematics and Consciousness

Volume 5, Number 1
2019–2022

CONTENTS

Editorial Introduction to the Journal
Tony Nader, MD, PhD, MARR i

Proving That Wholeness Is Indestructible
Paul Corazza, PhD 1

Natural Language Processing with Semantics and Logic: A Necessity Strongly Supported by Maharishi Vedic Science
Emdad Khan, PhD 43

A Publication of Maharishi International University
Fairfield, Iowa

ISSN 2376-6751

The International Journal of Mathematics and Consciousness is an annual journal dedicated to the mathematical description and understanding of consciousness.
Address correspondence to: *International Journal of Mathematics and Consciousness*, Department of Mathematics, Maharishi International University, Fairfield, Iowa, 52556, USA; email IJMAC@miu.edu.

EDITORIAL INTRODUCTION TO THE JOURNAL

Throughout history, natural phenomena have been ultimately mysterious. Some of these phenomena were explained by religious belief, others by philosophical analysis. Since the 17th century, the modern scientific approach has found that many phenomena in nature obey clearly describable physical laws. This success greatly widened the ambit of scientific inquiry beyond the physical into the realm of what previously had been considered metaphysical or nonmaterial. Today, the territory of scientific inquiry has expanded to include how matter leads to consciousness.

Most common and popular models of consciousness share the postulate that physical activity in the brain is prior to consciousness. No current theory, however, has been able to resolve the problem of how physical processes in the brain give rise to subjective experiences. Even quantum mechanical theories, while suggesting potential mechanisms that might create "unexplainable" phenomena, fall short of answering the fundamental questions about subjective experience. This gap—between the objective, material brain and the intimately known, private qualia of subjective experience, or "what it is like" to experience something—has so far not been bridged. Some thinkers have even rejected qualia out of hand, asserting that we have insufficient knowledge of the physical world to evaluate their existence.

Some believe that early *Homo sapiens* depended entirely on sensory experience as a reference for what does and does not exist, and that only as our understanding evolved did we come to challenge the evidence of our senses. Certainly, the discoveries of modern science changed the way we looked at the world. They gave us intellectual models of the universe that often seemed to contradict our sensory model but which provided in fact more accurate pictures and were eventually confirmed by experimental observation.

Perhaps the most notable example is the shift from a geocentric to a heliocentric view of the cosmos as a result of the work of Copernicus, Kepler, and Galileo in the 16th and 17th centuries. More recently, inquiry into very small and very large time and distance scales in relativity theory, quantum mechanics, quantum field theory, and cosmology has radically changed our beliefs about the nature of matter and physical phenomena as our senses perceive and our intellects apprehend them. We may ask, what actually exists for us? And we may agree that everything is continuously changing; we may even agree that whatever appears not to change is only one of an infinite number of simultaneously existing possibilities. For example, in some models a particle can be everywhere at once, and the fact that we find it here and now suggests either that we have collapsed the infinitude of its possibilities in a single act of conscious experience or that it continues to exist everywhere in an infinite number of universes parallel to the one in which we experience it.

In all this uncertainty, one fact seems undeniable: the fact of our own awareness. Without awareness, we can neither perceive nor apprehend, neither see nor think nor dream. Commonly, this awareness is called consciousness: the observer, the witness, the experiencer. If indeed this is the one undeniable fact, then it is timely that a scientific journal be dedicated to the study of consciousness as primary.

To be truly scientific requires that the journal obey rigorous methods of logic, research, and experimentation. At the same time, this requires that no *a priori* or unproven points of view stand in the way of new original postulates, previously explored theories revisited with new insights, or unconventional axioms.

The *International Journal of Mathematics and Consciousness* is founded in part to fulfill this need. The *Journal* opens the door to all mathematicians, scientists, and thinkers to present their theories of consciousness and the consequences thereof. With the requirement that such theories follow strict mathematical, logical argumentation and respect proven facts and observations, articles can be submitted for review, without restriction to their proposed axioms and postulates. The *Journal* also welcomes carefully reasoned articles that challenge commonly held, but not fully established, theories and beliefs.

1. Consciousness and "Consciousness at work"

Abstract concepts and subjective experiences such as love, friendship, beauty, devotion, happiness, inspiration, pain, despair, and deception, are, in and by themselves, hard to study scientifically because of their innate, subjective, personal nature. Even more difficult to study is the more abstract consciousness that seems to be like a screen on which these emotions, notions, and sensations are projected and experienced.

Modern cognitive neuroscience identifies various neural correlates of these mental states. The discipline of psychology attracted great thinkers who proposed various theories and methods of investigation, mostly focusing on the manifestations, observable or subjectively reportable signs and symptoms, and causes and effects of such inner experiences. Physicists recently have attempted to bridge the gap between the physical world and conscious experience through various quantum mechanical models.

Philosophy, metaphysics, and spiritual and religious studies delve into ontological, epistemological, and other fundamental questions, using more or less formal logic or a wide variety of opinions and postulates. In contrast, art forms such as music, painting, and fictional writing are outer expressions of inner experiences and creative thinking.

All theories, concepts, and creative work, whether scientific, psychological, philosophical, artistic, or spiritual are the manifestations of "consciousness at work." While it might be challenging to study "consciousness" as such, in and by itself, it may be easier to study "consciousness at work"—its dynamics and its manifestations.

The postulates that can be made about consciousness as an abstract phenomenon or epiphenomenon are most amenable to investigation by scientifically analyzing and studying "consciousness at work." The *International Journal of Mathematics and Consciousness* invites analyses of consciousness at work from various perspectives with a particular emphasis on mathematics.

2. Mathematics

Mathematics studies abstract forms, patterns, relationships, and transformations in an exact, systematic, and logical way. Forms and shapes like circles and triangles are the subject of geometry and topology. Patterns of number and operations lead to algebra. Relationships that change in time form the basis of calculus. Mathematics also includes the study of mathematics itself. The study of mathematical reasoning is undertaken by logic. Even questions about the limits of the mathematical method and the nature of mathematical knowledge can be addressed using the methodology of mathematics.

Using mathematical models of experimental observations of the physical world makes it possible to give a purely abstract formulation of real-life phenomena. Subjective mathematical reasoning, which is nevertheless entirely rigorous, applied to these models leads to new descriptions and predictions about the world.

Mathematics is fundamentally a method that finds patterns of orderliness in the subjective field of human intelligence and thought. Based on sets of axioms and postulates that are accepted without proof, mathematics gives a structure to the way our minds and intellects operate. It systematizes how individual human awareness perceives, discriminates, organizes, and expresses its own patterns of functioning. In our opinion, mathematics is certainly one of the most useful and scientifically manageable methods to study the interface between consciousness and physical phenomena.

Mathematics is in essence a subjective discipline that nevertheless allows us to organize and make sense of the physical universe in which we exist. Though subjective, it is precise and effective in objective scientific explorations. It is a fundamental and indispensable tool of all sciences, and at the same time, it is an expression of abstract human awareness and intellect.

3. Mathematics and Consciousness

The *International Journal of Mathematics and Consciousness* takes the position that methods of mathematics and mathematical modeling provide especially appropriate tools for studying the interface between consciousness and physical phenomena. As we have pointed out above, mathematics is a fundamental and indispensable tool of all sciences, and at the same time an expression of abstract human awareness and intellect. It is therefore the most precise scientifically reliable tool in the exploration of the dynamics of consciousness. It can be seen as the precise abstract representation of consciousness at work.

The ways in which human beings explore and express the experience of consciousness are as varied as nature itself. The following list contains some of the relevant sciences and other forms of human inquiry:

(1) Physics and chemistry (physical/quantum mechanical theories of consciousness at work)
(2) Biology and cognitive neuroscience (biological/electro-chemical/neural correlates of consciousness at work)
(3) Mathematics (abstract representation of consciousness at work)

(4) Psychology and cognitive sciences (objectification of subjective experiences of consciousness at work)
(5) Economics, particularly behavioral economics (production, distribution, and consumption of resources as models of the dynamics of consciousness at work)
(6) Philosophy (discursive representation of consciousness at work)
(7) Arts (subjective creative representation of consciousness at work)
(8) Religion (individual/group belief in the origins and dynamics of consciousness and consciousness at work)
(9) Spirituality (personal and totally subjective experience of consciousness at work)
(10) Study of pure consciousness itself (the field or screen "phenomenon" on which or by which all aspects of consciousness at work take place)

The *International Journal of Mathematics and Consciousness* maintains the position that of all such pursuits, mathematics, because of its rigor, depth, and effectiveness, is the most suitable discipline to study the interface between consciousness and the physical world. This *Journal* is devoted to exploring this interface using the rigorous approach of mathematics. We invite all mathematicians, scientists, and thinkers to submit papers using a mathematical approach to consciousness and "consciousness at work" in all its aspects.

Tony Nader, MD, PhD, M.A.R.R.

PROVING THAT WHOLENESS IS INDESTRUCTIBLE

Paul Corazza

ABSTRACT. In earlier research, we proposed a new axiom—the Wholeness Axiom—to be added to the currently accepted list of foundational axioms for all of mathematics. The Wholeness Axiom describes the self-interacting nature of the wholeness of the mathematical universe and gives mathematical expression to many principles of wholeness described in Maharishi Vedic Science. This article provides an accessible summary of recent research showing that the Wholeness Axiom is *indestructible*: Once the Wholeness Axiom is introduced in the mathematical universe, the Wholeness Axiom continues to hold true in all alternative universes that are created in independence proofs using the technique of forcing. This discovery provides strong evidence for the correctness of the Wholeness Axiom; it also shows that the Wholeness Axiom incorporates principles of Maharishi Vedic Science sufficient to give expression to the indestructible and invincible character of wholeness.

1. Introduction

An epic piece of eternal wisdom concerning the nature of wholeness, of pure consciousness, is articulated in the *Bhagavad-Gita* [14]:

> *Know That to be indeed indestructible by which all this is pervaded. None can work the destruction of this immutable Being.*
> – Maharishi, *Bhagavad-Gita*, II.17.

Indestructibility of wholeness means that it is not vulnerable to the conditions and vicissitudes of manifest existence; this point is made in the following verse [14]:

> *He is uncleavable; he cannot be burned; he cannot be wetted, nor yet can he be dried. He is eternal, all-pervading, stable, immovable, ever the same.*
> – Maharishi, *Bhagavad-Gita*, II.24.

Being indestrucible, wholeness exhibits an *invincible* quality: The dynamics to which wholeness is subjected do not disturb its nature; wholeness rests forever, invincibly, in its own nature, always as wholeness. In describing this invincible quality of pure consciousness, Maharishi says [15]:

> Nothing can disturb or disrupt the perfect balance and absolute order of this field of pure existence since everything that exists is a part of its structure and an expression of its own self-interacting dynamics.

Received by the editors March 18, 2020.

The fact that it is possible for an individual to tap the field of wholeness, of pure consciousness, through the process of transcending[1] opens the door to the possibility of establishing this remarkable quality of invincibility in individual life [17, pp. 138–143]. Progress in this direction appears as improvements in mental clarity, health, emotional fulfillment, and deep wisdom about one's true nature. Even more importantly, the invincible individual is established unshakeably in the Self, the deepest level of his own consciousness; no event in the field of manifest life could undermine this realization. The ignorance that appears to separate a person from his true nature has been, in the invincible individual, dispelled once and for all.[2]

In this article, we discuss a striking analogy between indestructibility of wholeness, as described in Maharishi Vedic Science (and indeed in many other ancient traditions of knowledge), and indestructibility of wholeness in the domain of mathematics. It has long been known that the usual foundational axioms for mathematics do not say anything about the wholeness of the universe of mathematics. In earlier research, the author proposed a new foundational axiom, the *Wholeness Axiom*, that does give expression to the characteristics of wholeness of the mathematical universe. This article discusses a recent discovery that the Wholeness Axiom cannot be destroyed by even the radical changes to the universe that are imposed when mathematicians create new *alternative* universes, which are used in independence proofs. We will see how our mathematical formulation of wholeness captures many of the essential characteristics of wholeness discussed in Maharishi Vedic Science. We will also see how the mechanics of structuring invincibility in the life of the individual and society find expression in this new indestructibility result in the domain of pure mathematics.

We will begin the discussion with a review of the technical issue that originally motivated the Wholeness Axiom. The Wholeness Axiom was designed to provide an account of mysterious, extraordinarily large sets, called *large cardinals.* Large cardinals[3] started to appear in mathematical research as far back as 100 years ago, but, unlike virtually all other mathematical concepts, large cardinals were found to be *underivable* from the known foundational axioms. We will then review the solution to this problem obtained by introducing the Wholeness Axiom as a new foundational axiom. We will point out the central role played by the principles of Maharishi Vedic Science, embodied in the new axiom, in arriving at this solution. Finally, we will turn to the main topic of this article: the discovery that the Wholeness Axiom is indestructible.

[1]The most widely researched technique for transcending is Maharishi's Transcendental Meditation program (see `https://www.tm.org/research-on-meditation` for a survey of some of this research). Maharishi remarks [19]:

> The process of Transcendental Meditation brings the active wandering mind to this state of Transcendental Consciousness, the seat of all Laws of Nature in their unified (Saṁhitā) state. (p. 30)

[2]See for example [14, pp. 333–4].

[3]An accessible introduction to large cardinals can be found in [6].

2. The Mathematical Infinite and Large Cardinals

To appreciate the Wholeness Axiom in context, we need to review briefly the efforts of mathematics to come to grips with the concept of the "Infinite." Nineteenth-century mathematician Georg Cantor began the real mathematical study of the concept of the Infinite with the following question: We know that the set $\mathbb{N} = \{1, 2, 3, \ldots\}$ of natural numbers and the set $\mathbb{R}$ of real numbers are infinite. Which collection is bigger? Or do they have the same size? Cantor showed, remarkably enough, that there are more points on the real number line than there are natural numbers:

$$\mathbb{N} < \mathbb{R}.$$

Cantor went on to show how to obtain even larger infinite sets than $\mathbb{R}$. Recall that for any set X, a set A is a *subset* of X if every element of A is also an element of X. Cantor made an important observation about the set of all subsets of a given set X, called the *power set of* X and denoted $\mathcal{P}(X)$. Consider for example the set $X = \{1, 3\}$: The set $\mathcal{P}(X)$ of all subsets of this set is $\{\{1\}, \{3\}, \{1, 3\}, \{\}\}$. We notice that $\mathcal{P}(X)$ has four elements while X has only two. Cantor proved that this relationship between a set and its power set always holds true: For any set, finite or infinite, the set $\mathcal{P}(X)$ is bigger than X. In particular, if we start with $\mathbb{R}$, then $\mathcal{P}(\mathbb{R})$ must be a larger infinity than $\mathbb{R}$. This leads to the remarkable fact that there is an endless hierarchy of ever larger infinities, which we can obtain by repeatedly applying the power-set operation $\mathcal{P}$:

$$\mathbb{N} < \mathcal{P}(\mathbb{N}) < \mathcal{P}(\mathcal{P}(\mathbb{N})) < \cdots.$$

Cantor's work led to the insight that there is a great variety of infinite sizes, called *infinite cardinals*. Cantor provided names for these cardinal numbers. Finite cardinal numbers are already familiar from experience: $0, 1, 2, 3, \ldots$. The cardinal number representing the size of $\mathbb{N}$ (the smallest infinite set) is ω (sometimes written ω_0). The next larger infinite cardinal is ω_1, and the pattern of increasing the value of the index of these cardinals (from 0 to 1 to 2, and so forth) continues. The first few infinite cardinals are shown here:

$$\omega_0, \omega_1, \omega_2, \omega_3, \ldots.$$

It is known that every infinite set has size that is a cardinal number lying somewhere on Cantor's long list of infinite cardinals.

Something unexpected happens when we begin to study another kind of infinity known as *large cardinals*. Examples of large cardinals began to arise just a few years after Cantor's work; they arose as theorists sought to determine whether certain combinations of properties of ordinary infinite cardinals could belong to a single infinite cardinal.[4] Some of these combinations turned out to be so potent that any cardinal exhibiting such properties would have to be extraordinarily large.

Many such infinities were invented and, over time, two things happened. First, it was discovered that some of the really big cardinal notions that theorists came upon in those days could not be proven to exist at all; in other words, it was found to be impossible to prove that certain unusual combinations of properties could be found in any particular infinite cardinal on Cantor's list. Secondly, some of those same cardinal notions turned out to be key elements for solving research problems in many areas of mathematics. These extraordinary cardinals came to be known as *large cardinals*: Large cardinals are cardinals possessing such a potent combination of properties, they cannot be proven to exist.

And where exactly on Cantor's list of cardinals can a large cardinal be found? It is a fact that every infinite size, including large cardinals, must lie somewhere in Cantor's list. However, in the case of large cardinals, it is impossible to specify *which* of the cardinals in Cantor's list are actually large. Indeed, if we could devise a procedure for locating a large cardinal in this way, this procedure would itself be a proof that a large cardinal exists!

The fact that large cardinals cannot be proven to exist on the basis of the known axioms of mathematics led the community of experts in foundations to recognize the need for new axioms to supplement the currently accepted list of axioms; such new axioms would make it possible to derive the known large cardinals, just as all other known mathematical concepts and truths can be derived from those axioms today.

3. The Wholeness Axiom

To understand the need for a new axiom, we need to understand what "axioms" are. Axioms are first principles that are taken to be fundamental truths. All mathematical objects can be represented as sets, so the fundamental axioms for mathematics are axioms about *sets*. The axioms at the basis for all of mathematics are known collectively as the *Axioms of ZFC*; "ZFC" stands for Zermelo Fraenkel set theory with the axiom of Choice.

The ZFC axioms tell us the fundamental properties that sets must have and provide a kind of "instruction manual" for building a universe of sets.

---

[4] One of the early efforts in this direction, which gave rise to the first type of large cardinal ever to be studied, attempted to combine the property of *regularity*, which some cardinals have, with the property of being a *fixed point*. In this footnote, we briefly discuss each of these properties.

Regularity is a property that is easily observed in the infinity represented by the set $\mathbb{N}$ of natural numbers: It is impossible to obtain a *finite* sequence of *finite* subsets $A_1, A_2, \ldots, A_k$ of $\mathbb{N}$ so that their union $A_1 \cup A_2 \cup \cdots \cup A_k$ includes every element of $\mathbb{N}$. To "fill up" $\mathbb{N}$, one either needs *infinitely many* subsets of $\mathbb{N}$ or else one needs at least one of the subsets to be *infinite*. One says for this reason that ω (which is the size or cardinality of $\mathbb{N}$) is a *regular cardinal*. In general, an infinite cardinal λ is said to be *regular* if, for any set X of size λ, it is impossible to obtain a collection $\mathcal{C}$ of fewer than λ subsets of X, each of size less than λ, so that the union of the sets in $\mathcal{C}$ is equal to X.

An infinite cardinal ω_α is a *fixed point* if its index α is equal to the cardinal ω_α itself. An example of a fixed point is given in [6, p. 73].

A cardinal that is both regular and a fixed point is called *weakly inaccessible*; weakly inaccessible cardinals are the weakest type of large cardinal known.

Here are a few examples of these axioms:

Some ZFC Axioms

Pairing Axiom. If X and Y are sets, there is another set Z that has X and Y as its only elements. (Notation: $Z = \{X, Y\}$.)

Powerset Axiom. If X is a set, the collection $\mathcal{P}(X)$ of all subsets of X is also a set.

Axiom of Infinity. There is an infinite set. (Equivalently: The natural numbers $1, 2, 3, \ldots$ can be collected together to form a set.)

The ZFC Axioms tell us how to build the universe V of all sets by piecing together a vast collection of ever larger *partial universes* $V_0, V_1, V_2, \ldots$.

$$\begin{array}{lcl} V_0 & = & \emptyset \qquad \text{(the empty set)} \\ V_1 & = & \mathcal{P}(V_0) = \{\emptyset\} \\ V_2 & = & \mathcal{P}(V_1) = \{\emptyset, \{\emptyset\}\} \\ V_3 & = & \mathcal{P}(V_2) \\ \vdots & & \vdots \end{array}$$

Piecing all these parts together[5] gives us the universe V (Figure 1):

$$V = V_0 \cup V_1 \cup V_2 \cup \cdots .$$

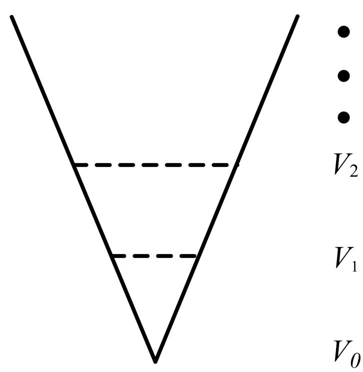


Figure 1: The Universe V of Sets

[5]It should be noted that, although each of the stages $V_0, V_1, V_2, \ldots$ displayed here is only finite, the stages eventually achieve arbitrarily big infinite size as indices of the stages progress beyond the whole numbers $0, 1, 2, \ldots$.

The axioms of ZFC provide mathematics with a unified foundation; nearly every mathematical fact can be demonstrated by formulating the fact in the language of sets and then deriving it directly from the ZFC axioms. However, the ZFC axioms are unable[6] to derive the existence of large cardinals.

It is generally agreed in the set-theory community that a new axiom, to be added to the ZFC axioms, is needed in order to account for the apparently inevitable presence of large cardinals in the universe, but attempts at formulating such an axiom have been only partially satisfactory. What is needed is a deep insight into the structure of the Infinite.

An important observation about the ZFC axioms that led the author to formulate the Wholeness Axiom is that the ZFC axioms talk about "the parts" of the mathematical universe only; they do not address the nature and character of its *wholeness*. Moreover, it seemed clear that the wholeness represented by V should exhibit the same essential characteristics as the wholeness that is spoken of in Maharishi Vedic Science.

In the formulation of an axiom that could fill the need, the intention was to give mathematical expression to many of the most salient features of wholeness as desribed in Maharishi Vedic Science. In particular, we attempted to capture the following principles in the mathematical formulation.[7]

> Wholeness, by nature, moves within itself, knows itself, and undergoes a wide range of transformations in the unfoldment of existence, and yet remains unchanged by these transformations. Moreover, the

[6] The inability to derive the existence of large cardinals from the ZFC axioms is a consequence of Gödel's Second Incompleteness Theorem, which says that no consistent foundational theory (like ZFC) can prove its own consistency. Gödel also showed that a theory like ZFC is consistent if and only if there is a model, or universe of sets, in which all the axioms of the theory hold true. It is known that if a large cardinal exists, it immediately gives rise to a model of set theory. Therefore, if ZFC *could* prove the existence of a large cardinal, it would at the same time prove the existence of a model of ZFC, which, by the Second Incompleteness Theorem, is impossible. See the appendices to Chapter 10 in [8] for an accessible treatment of this topic.

[7] This summary of principles is inspired by the following descriptions in Maharishi Vedic Science:

> The self-referral state of consciousness is that one element in nature on the ground of which the infinite variety of creation is continuously emerging, growing, and dissolving. The whole field of change emerges from this field of nonchange, from this self-referral, immortal state of consciousness. [17, p. 25]
>
> The deepest level of every grain of creation is the self-referral field, the transcendental level of pure intelligence, the self-referral state of Unity—pure wakefulness, pure intelligence—*Chiti Shaktiriti*—as expressed by the last *Yog-sūtra*—that self-referral intelligence which is the common basis of all expressions of Natural Law. [19, p. 425]
>
> The essential and ultimate constituent of creation is the absolute state of Being or the state of pure consciousness. This absolute state of pure consciousness is of unmanifested nature which is ever maintained as that by virtue of the never-changing cosmic law. Pure consciousness, pure Being, is maintained as pure consciousness and pure Being all the time, and yet it is transformed into all the different forms and phenomena. Here is the cosmic law, one law which never changes and which never allows absolute Being to change. Absolute Being remains absolute Being throughout, although it is found in changed qualities here and there in all the different strata. [16, p. 12]

> tranformational dynamics of wholeness are present at every point in creation. [17, p. 25]

Our formulation of these points in a mathematical context is the following (Figure 2):

> **Wholeness Axiom (WA)**. There is an elementary embedding $j : V \to V$ for which there is a least cardinal κ moved by j; that is, $j(\kappa) \neq \kappa$. Moreover, for every set X in the universe, the restriction of j to X is a set that also belongs to the universe. The embedding j is called a *WA-embedding*.

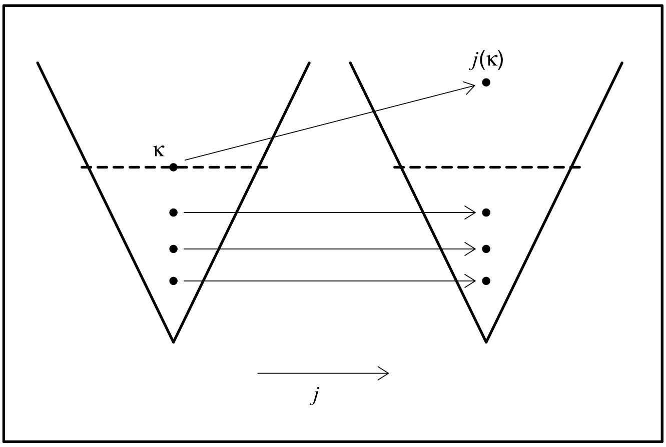


Figure 2: Elementary Embedding $j : V \to V$ with κ the First Cardinal Moved

The Wholeness Axiom embodies many of the characteristics and dynamics attributed to wholeness, pure consciousness, in Maharishi Vedic Science. Here are a few points:

(1) V naturally represents wholeness in the mathematical context: Every "existent thing" belongs to V, yet V is "bigger than the biggest"—it is too big to be itself a set.

(2) $j : V \to V$ represents the dynamics of wholeness. The map j is described as an *elementary embedding*. This means that every relationship among sets that is found in V is preserved by j. For instance, if X, Y, and Z are sets in V and it happens that X is an element of Y and Z is the power set of Y, then it must also be true that $j(X)$ is an element of $j(Y)$ and $j(Z)$ is the power set of $j(Y)$. One says that j is a *truth-preserving mapping*. In this sense, V remains "unchanged" by its own transformational dynamics, embodied in j.

(3) The cardinal κ is the "point" representing the first sprouting of activity of j. The map j does not move sets X that occur in V at a stage earlier than the κth stage; that is, $j(X) = X$

for all sets X that belong to a stage V_α, where $\alpha < \kappa$. The cardinal κ is the first cardinal moved by j.

(4) Interaction between j and its point κ generates a "blueprint" (a *Laver sequence*), which is a special sequence of sets that encodes all sets in the universe.

(5) Interaction between j and the blueprint generates every object in the universe.

(6) The embedding j is not far removed from the concrete sets living in V—this characteristic is guaranteed by the fact that, for every set X, the restriction of j to X is an actual set in the universe. This means that the dynamics represented by j are "present at every point."

Points (3)–(5) parallel the insight that, in the flow of wholeness within itself, expressed by the first letter A of Ṛk Veda, wholeness collapses to its own point; this collapse is embodied in the second letter K of Ṛk Veda. Moreover, from these dynamics emerges the blueprint for creation, the *Veda*, and from the *Veda* emerges[8] the universe, *Vishwa* (Figure 3).

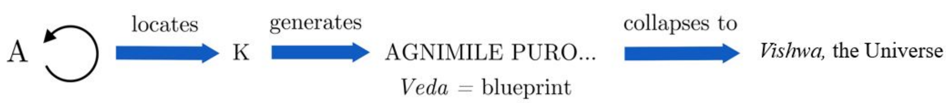


Figure 3: Collapse of A to K and Expansion to Veda and Vishwa

These points highlight the extent to which the Wholeness Axiom embodies principles and dynamics of wholeness, as described in Maharishi Vedic Science.

The Wholeness Axiom, when added to the list of ZFC axioms, also achieves another goal: It gives a complete account of all the most widely studied large cardinals. The strongest among these is known as *super-n-huge for every n*. One can show that the first cardinal κ moved by j, representing the first sprouting of activity in the unfoldment of the universe, is[9] super-n-huge for every n.

[8] The following passages give more precise expression to these dynamics.

> The sound क् (K) expresses "stop"—stop of the continuous flow of अ (A). क् (K), the continuous stop, is actually a commentary on the continuous flow of अ (A). अक् (Ak) indicates eternal stop on the ground of eternal continuum expressing *Ātmā* and its indescribable nature upheld by the unmanifest continuum of अक् (Ak)—the eternal unstructured *Ātmā* completely expressed by the expression अक् (Ak). [19, pp. 501-2]

Maharishi continues:

> The spontaneous expansion of अक् (Ak) into the Veda and Vedic Literature is actually the analysis of *Ātmā*; furthermore, the expression of Veda into *Vishwa* is actually the continuous process of expansion (evolution) of *Ātmā*. [19, pp. 503]

[9] In fact, not only is it true that κ is super-n-huge for every whole number n, but actually, κ is the κth cardinal that is super-n-huge for every n; this means that there are κ cardinals less than κ that are also super-n-huge for every n. Moreover, in the presence of the Wholeness Axiom, it

4. Indestructibility of Wholeness

The Wholeness Axiom has been successful at giving an account of large cardinals. But to qualify as an axiom that is truly acceptable as one of the fundamental axioms at the basis of all of mathematics, it must satisfy additional criteria. One of the most important of these is that it must survive after the universe has been transformed in various ways in *independence proofs*.

Typically, mathematicians are involved in the following two sorts of activities:

(1) Proving certain mathematical statements are true.
(2) Proving certain mathematical statements are false.

In the past 60 years, experts in foundations of mathematics have also become involved in a third activity:

(3) Proving that a mathematical statement can be neither proved nor disproved.

A statement that can be neither proved nor disproved from the axioms of set theory is said to be an *undecidable proposition*. A proof that a statement is undecidable is called an *independence proof*.

Independence proofs are produced using a special technology called the *technique of forcing*. Forcing is a way of producing a new universe V' in which some desired statement S holds. The new universe V' is very much like V in the sense that it also satisfies all the axioms of ZFC and it also is built up in stages, starting from the empty set. But V' is crafted by the forcing method to satisfy the statement S as well.

To establish that a particular statement T is undecidable, forcing experts design two separate forcing arguments: One forcing argument produces a universe in which T holds true; producing such a universe establishes that T is *consistent*. The second forcing argument produces a universe in which T is false; producing this second universe establishes that the *negation* of T is also consistent.

Here is a summary of the steps of a forcing argument.

(1) Start with a statement S that you hope to prove is undecidable—neither provable nor disprovable.
(2) Come up with a partially ordered set P (called a *notion of forcing*) that expresses your intention to make S true. Then, later, come up with another partially ordered set Q that expresses the intention to make S *false*.
(3) Expand the universe V, using P, to an "all possibilities" state V^P. Elements of V^P are no longer sets, but are potential sets (V^P is said to be a universe of *names*).
(4) Collapse V^P using an "ideal set" G, known as a *generic filter*. G is a special subset of P, but is not actually a set in the universe V. Techniques of logic, similar to the method[10] of producing the imaginary number i and the complex number field, starting from the real number line, allow us to

can be shown that "almost all" cardinals in the universe are super-n-huge for every n! This result illustrates the principle that when wholeness is missing from life, even its existence is called into doubt, but when wholeness has been awakened, one finds wholeness everywhere. Indeed, wholeness is all there is.

[10] See Appendix I for a discussion of the parallels between the construction of the field of complex numbers and the forcing methodology.

assume that this generic filter G can be found. The "collapse" of V^P is always done in the same way, and produces a new universe that is denoted $V[G]$ (Figure 4). The universe $V[G]$ is called the *forcing extension of* V *obtained by forcing with* P; it is the smallest universe that contains every set in V and also contains the set G. In $V[G]$, S ends up being true.

(5) Then do steps (3) and (4) again for the partially ordered set Q. In the forcing extension obtained by forcing with Q, S ends up being false.

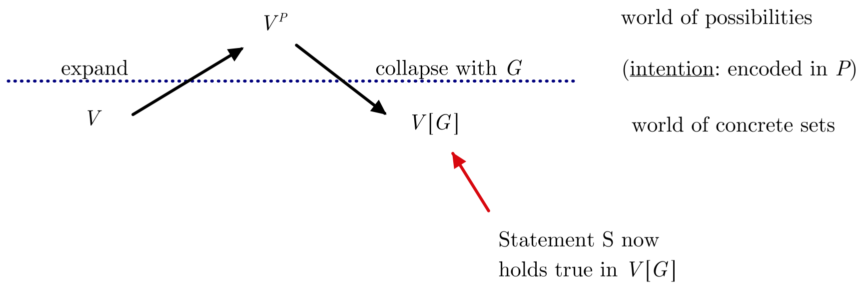


Figure 4: Forcing: Expand to V^P, Then Collapse to $V[G]$

The result of applying Steps (1)–(5) is a proof of the undecidability of S. One certainly cannot prove S to be true since we have obtained a universe in which S is false. And one cannot prove S is false since we have obtained a universe in which S is true. Therefore, we have shown S is neither provable nor disprovable.

Historically, the first example of an independence proof was concerned with the size (cardinality) of the real number line. Cantor showed that $\mathbb{N} < \mathbb{R}$. But which of the cardinals in Cantor's list specifies the exact size of $\mathbb{R}$? Since $\mathbb{N}$ has size ω_0, we know $\mathbb{R}$ has size greater than ω_0.

The *Continuum Hypothesis* (CH) is the statement that $\mathbb{R}$ has the smallest possible infinite size bigger than ω_0, namely ω_1. Cantor believed CH is true but could not prove it.

In 1900, the famous mathematician David Hilbert compiled a list of the 23 most important unsolved problems in mathematics, and number one on the list was to settle the Continuum Hypothesis.

Many years after Hilbert announced his list of problems, the Continuum Hypothesis was finally settled, but not in the expected way: It was shown that CH is *undecidable*. Researchers discovered partially ordered sets P and Q so that forcing with P makes the Continuum Hypthesis true[11], and forcing with Q makes the Continuum Hypothesis false.[12]

The question about forcing that concerns us is this: What happens to the Wholeness Axiom after forcing? If we take the Wholeness Axiom to be true, so that it holds in V, is the Wholeness Axiom still true in the new universe $V[G]$?

[11] This is done by forcing a new subset of ω_1; details can be found in [13].

[12] The logic for this forcing argument is given below, starting in the section "Tracing Through the Proof."

It is vital to ask this kind of question when one wishes to add a new axiom to the ZFC axioms that is truly fundamental and worthy of the name "axiom." It is *always* true that a forcing extension satisfies all the ZFC axioms; no axiom is destroyed by forcing. If some new axiom is suggested that then is destroyed by forcing, it simply is not robust enough to be taken seriously as an axiom.

For instance, consider the axiom I that asserts the following:

> I: There exists exactly one weakly inaccessible cardinal.

A weakly inaccessible cardinal is one of the "smaller" varieties of large cardinals.[13] One cannot prove I from ZFC but one could conceivably add I as a new axiom, so that our new axioms for mathematics would consist of ZFC together with I. However, I is not a good axiom because it is easy to perform a forcing argument for which the forcing extension contains *no* weakly inaccessible cardinal.[14]

The "indestructibility" result concerning the Wholeness Axiom that has been recently discovered is the following:

> **Theorem** [10]. If the Wholeness Axiom holds in V, then for any notion P of forcing, the forcing extension $V[G]$ obtained from forcing with P also satisfies the Wholeness Axiom. Wholeness is indestructible.

The next section outlines some of the details leading up to this new discovery and also outlines the proof.

5. Tracing Through the Proof

In this final section, we give a flavor of the proof of the main result in the relatively simple case in which we perform forcing to obtain a model in which CH is false; we will indicate why, if the Wholeness Axiom is true in the starting universe, it remains true in the forcing extension.

We begin with an outline of the forcing method for adding many reals to the universe; for this, we start with the simplest case, in which just one new real is added to the universe.

5.1. **The Forcing Technology and Adding a Single Real to the Universe.** The idea behind forcing is to expand the universe V to an "all possibilities" state, keeping in mind the *intention*—in this case, the intention is to add a new real to the universe. In forcing, the intention is represented by the choice of a partially ordered set. A partially ordered set is a set with an order[15] relation $\leq$. The set $\mathbb{N}$ of natural numbers is a simple example of a partially ordered set; in this case, the relation $\leq$ is simply the natural ordering of natural numbers: For instance, $2 \leq 5$, $7 \leq 23$, and

[13] Weakly inaccessible cardinals were discussed in the footnote on p. 4.

[14] This is done by adding an onto function from ω to the weakly inaccessible; see [13].

[15] Speaking more precisely, a relation $\leq$ on a set X is a *partial ordering* if it satisfies the following properties: For all $x, y, z \in X$, the following must hold:

(i) (Reflexive Property) $x \leq x$;
(ii) (Antisymmetric Property) if $x \leq y$ and $y \leq x$, then $x = y$;
(iii) (Transitive Property) if $x \leq y$ and $y \leq z$, then $x \leq z$.

The examples of partial orders mentioned in the main text satisfy these properties.

$18 \not\leq 5$. Another example is the subset relation $\subseteq$ defined on sets. Here, the subset relation ($\subseteq$) itself serves as the partial order relation: For any sets X and Y, we can declare that $X \leq Y$ if and only if $X \subseteq Y$. Note that in this case, we can come up with sets X and Y for which $X \not\leq Y$ and $Y \not\leq X$; for instance, let $X = \{1, 2\}$ and $Y = \{1, 3\}$. Because there are two elements of this partial order that cannot be compared, one says that the partial order is not *total*; note that, by contrast, the natural partial ordering of $\mathbb{N}$ *is* a total ordering.

The partially ordered set for adding a new real to the universe involves functions from finite subsets of $\mathbb{N}$ to the set $\{0, 1\}$. The forcing method in this case will actually add a new function $g : \mathbb{N} \to \{0, 1\}$ to the universe. Given any sets X and Y, a *function* f from X to Y (notation: $f : X \to Y$) is a rule[16] that associates to each element x of X an element y of Y; one writes $f(x) = y$. The *domain* of f is the set of inputs of f—in this case the domain is X. We write $\operatorname{dom} f = X$. The *range* of f is the set of outputs of f—in this case, the range of f is a subset of Y. We denote the range of f with the notation $\operatorname{ran} f$.

Any function g from $\mathbb{N}$ to $\{0, 1\}$ specifies a real number x by the following formula:[17]

$$x = \frac{g(1)}{2} + \frac{g(2)}{2^2} + \frac{g(1)}{2^3} + \cdots + \frac{g(n)}{2^n} + \cdots . \tag{5.1}$$

Therefore, if we can show how to add a new function $g : \mathbb{N} \to \{0, 1\}$ to the universe, we have at the same time added a new real as well.

The partially ordered set P we will use consists of all functions $p : A \to \{0, 1\}$, where A is a finite subset of $\mathbb{N}$; such functions are called *finite partial functions.* For instance, the following function $p : \{1, 3, 8\} \to \{0, 1\}$ is a finite partial function and therefore belongs to P:

$$p(1) \;=\; 1$$

[16]We can examine the features of a function using one of the examples given in the next few paragraphs. Let $p : \{1, 3, 8\} \to \{0, 1\}$ be the function defined as follows:

$$\begin{aligned} p(1) &= 1 \\ p(3) &= 0 \\ p(8) &= 1 \end{aligned}$$

Here, the elements of the domain of p are $1, 3$, and 8, so we write $\operatorname{dom} p = \{1, 3, 8\}$. The elements of the range of p are 0 and 1, so we write $\operatorname{ran} p = \{0, 1\}$. Note that p assigns just one value to each element of its domain; for instance, we could not have defined p so that $p(1) = 1$ and also $p(1) = 0$—a function must assign *only one* value to each element of its domain.

[17]This formula becomes more accessible when you consider the fact that the infinite series

$$\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots$$

sums to the number 1 (notice that $\frac{1}{2} + \frac{1}{4} = \frac{3}{4}$ and $\frac{1}{2} + \frac{1}{4} + \frac{1}{8} = \frac{7}{8}$, so the partial sums of the series tend to 1). Also notice that if we change all the numerators of this series to 0, we obtain the series

$$\frac{0}{2} + \frac{0}{4} + \frac{0}{8} + \cdots ,$$

which sums to 0. Therefore, it is not unreasonable to expect (and it can be proven) that *every* real number between 0 and 1 can be obtained as a sum of the form

$$\frac{i_1}{2} + \frac{i_2}{4} + \cdots + \frac{i_n}{2^n} + \cdots ,$$

for some sequence $i_1, i_2, i_3, \ldots$ of 0s and 1s.

$$\begin{array}{rcl} p(3) & = & 0 \\ p(8) & = & 1 \end{array}$$

The elements of P represent *approximations* to the function g that we are intending to build.

We need to define a relation $\leq$ for P: Given $q, p \in P$, we declare[18] $q \leq p$ if and only if the domain of p is a subset of the domain of q, and, for all $n \in \operatorname{dom} p$, $p(n) = q(n)$. For instance, if we define $q : \{1,2,3,8,10\} \to \{0,1\}$ by

$$\begin{array}{rcl} q(1) & = & 1 \\ q(2) & = & 1 \\ q(3) & = & 0 \\ q(8) & = & 1 \\ q(10) & = & 0 \end{array}$$

then, referring to our earlier example of p, we have $\operatorname{dom} p \subseteq \operatorname{dom} q$, and p and q agree on the domain of p. Therefore, we may say that $q \leq p$; one says that q *extends* p, and also that q is *stronger than* p.

Intuitively, to motivate this particular order relation, we can imagine obtaining the new function $g : \mathbb{N} \to \{0,1\}$ by piecing together some functions $p_1, p_2, \ldots$ all belonging to P, in the following way. We define $p_1, p_2, \ldots$ as follows:

$$\operatorname{dom} p_1 = \{1\}, \operatorname{dom} p_2 = \{1,2\}, \ldots, \operatorname{dom} p_n = \{1,2,\ldots,n\}, \ldots$$

and for each n, for each even number i in the domain of p_n, we define $p_n(i) = 0$, and for each odd number i in the domain of p_n we define $p_n(i) = 1$.

$$\begin{array}{rcl} p_1(1) & = & 1 \\ p_2(1) & = & 1 \text{ and } p_2(2) = 0 \\ p_3(1) & = & 1 \text{ and } p_3(2) = 0 \text{ and } p_3(3) = 1 \\ \cdot & = & \cdot \\ \cdot & = & \cdot \\ \cdot & = & \cdot \end{array}$$

We can see that for each n, p_{n+1} extends p_n. We have:

$$\cdots \leq p_{n+1} \leq p_n \leq \cdots \leq p_3 \leq p_2 \leq p_1.$$

In other words, as n gets larger, the domains become larger and all functions in the sequence agree on the common part of their domains. If we then form the "union" of these functions $f = p_1 \cup p_2 \cup \cdots \cup p_n \cup \cdots$, we may conclude that f is a function from $\mathbb{N}$ into $\{0,1\}$. Note that this particular way of obtaining f is just one of many ways one could go about forming a function $\mathbb{N} \to \{0,1\}$ by piecing together elements of P. We must mention, however, that, though the f we obtained in our example is indeed a function from $\mathbb{N}$ to $\{0,1\}$, it is still very far from being a "new" function added to the universe. This is because the procedure we used to define f is definable in the universe; it is just an ordinary construction that can be carried out in the universe. To actually obtain a *new* function from $\mathbb{N}$ to $\{0,1\}$ requires us to obtain a collection $G = \{p_1, p_2, p_3, \ldots\}$ of elements of P that is so jumbled up

[18] The reader may wish to verify that the partial ordering defined here satisfies the three properties of partial orders, mentioned in footnote 15.

that, as a set, it cannot belong to V. Such a subset G of P is called a *generic filter*, a term we will define shortly.

The example given above provides an opportunity to introduce some useful notation for forming unions of this kind. Suppose we define the set G to be $\{p_1, p_2, \ldots\}$. Instead of writing $f = p_1 \cup p_2 \cup \cdots \cup p_n \cup \cdots$ as we did above, we could write $f = \bigcup G$; the notation $\bigcup G$ indicates that we are forming the union of all the elements of G.

Since the concept of forming the union of a collection of functions is a key point in the forcing method we describe here, we illustrate with another example. Suppose we have two partial functions $u : \{3, 7, 8\} \to \{0, 1\}$ and $v : \{1, 7, 8, 20\} \to \{0, 1\}$, defined by

$$\begin{array}{rclrcl} u(3) & = & 1 & v(1) & = & 0 \\ u(7) & = & 0 & v(7) & = & 0 \\ u(8) & = & 1 & v(8) & = & 1 \\ & & & v(20) & = & 1. \end{array}$$

In this case, the relationship between u and v is not as convenient as in the earlier example—we do not have $u \leq v$, nor do we have $v \leq u$. Nevertheless, we can form[19] the union $w = u \cup v$ to produce a function $w : \{1, 3, 7, 8, 20\} \to \{0, 1\}$ whose domain is the union of the domains of u and v and whose value at each element i of its domain is computed by computing either $u(i)$ or $v(i)$. Although we do not have that $u \leq v$ or that $v \leq u$, we *can* form the union to obtain a function because u and v agree on the common part of their domains. Indeed, the only reason a union of this kind could fail to produce a function is if the functions that are being pieced together disagree at some value. Here, notice that u and v both contain 7 and 8 in their domains and their values at 7 agree ($u(7) = 0 = v(7)$) and their values at 8 agree ($u(8) = 1 = v(8)$). We end up with a new function w with a larger domain.

$$\begin{array}{rcl} w(1) & = & 0 \\ w(3) & = & 1 \\ w(7) & = & 0 \\ w(8) & = & 1 \\ w(20) & = & 1. \end{array}$$

In a similar way, we can form the union of infinitely many finite partial functions to obtain a function whose domain is all of $\mathbb{N}$. Whenever two functions agree on the common parts of their domains, we say that the functions are *compatible*. Forming the union of a collection of functions that are compatible with each other always produces another function (whose domain is the union of the domains of the functions in the collection).

[19] Even in this simple example, the idea that we are forming the *union* of two functions may seem unfamiliar. The idea makes more sense though when you represent a function as a set of ordered pairs: For any function h, we can identify h with the set of pairs $\{(x, y) \mid y = h(x)\}$. This set is sometimes called the *graph* of h. In this example, using this definition of "function," we have

$$\begin{array}{rcl} u & = & \{(3,1), (7,0), (8,1)\}; \\ v & = & \{(1,0), (7,0), (8,1), (20,1)\}; \\ w & = & \{(1,0), (3,1), (7,0), (8,1), (20,1)\} = \{(3,1), (7,0), (8,1)\} \cup \{(1,0), (7,0), (8,1), (20,1)\}. \end{array}$$

Here, then, it is clear that w is literally the union of u and v.

To produce a new real—a real that does not belong to the universe V—we will need to have a way to extract from P some subset G consisting of finite partial functions that are compatible with each other; and, morever, the union of the domains of functions in G must be precisely $\mathbb{N}$ (and not just a subset of $\mathbb{N}$). These requirements, on their own, are easy enough to fulfill—we have already given an example of how to accomplish this. But, in addition, we need to be sure that the function we get by forming the union of these partial functions is *new*—that it is not already present in the universe V. This latter requirement is quite different from the simple need to produce a function defined on all of $\mathbb{N}$. To meet these requirements, we will require G to have the properties of a *generic filter*. To understand genericity, we first need the concept of a *dense* subset of a partial order, so we define this first.

> **Definition** (Dense Set). Suppose $(P, \leq)$ is a partial order. Then a subset D of P is *dense* if, for each $p \in P$, there is $d \in D$ with $d \leq p$.

Let's consider some examples of dense sets in our partial order P of finite partial functions. We will use these dense sets later to check that the function g that we create yields a *new* real number.

Dense Set Example 1. Let $n \in \mathbb{N}$ and define the set D_n by

$$D_n = \{p \in P \mid n \in \operatorname{dom} p\}.$$

We verify that D_n is dense. Let $p \in P$. We obtain $d \in D_n$ with $d \leq p$. If $n \in \operatorname{dom} p$ already, then we simply let $d = p$. If $n \notin \operatorname{dom} p$, define a function d as follows. Let $\operatorname{dom} d = \operatorname{dom} p \cup \{n\}$. Define d on its domain by:

$$d(i) = \begin{cases} p(i) & \text{if } i \neq n \\ 0 & \text{if } i = n. \end{cases}$$

It is clear now that $\operatorname{dom} d \supseteq \operatorname{dom} p$ and that d and p agree on $\operatorname{dom} p$; it follows that $d \leq p$. Since $n \in \operatorname{dom} d$, it follows that $d \in D_n$. We have shown D_n is dense in P.

Dense Set Example 2. Let $i \in \{0, 1\}$ and define the set E_i by

$$E_i = \{p \in P \mid i \in \operatorname{ran} p\}.$$

We verify that E_i is dense. Let $p \in P$. We obtain $d \in E_i$ with $d \leq p$. If $i \in \operatorname{ran} p$ already, then we simply let $d = p$. If $i \notin \operatorname{ran} p$, define a function d as follows. Pick $n \in \mathbb{N}$ so that $n \notin \operatorname{dom} p$; this is possible since $\operatorname{dom} p$ is a finite subset of $\mathbb{N}$. Let $\operatorname{dom} d = \operatorname{dom} p \cup \{n\}$. Define d on its domain by:

$$d(k) = \begin{cases} p(k) & \text{if } k \neq n \\ i & \text{if } k = n. \end{cases}$$

It is clear now that $\operatorname{dom} d \supseteq \operatorname{dom} p$ and that d and p agree on $\operatorname{dom} p$; it follows that $d \leq p$. Since $i \in \operatorname{ran} d$, it follows that $d \in E_i$. We have shown E_i is dense in P.

Dense Set Example 3. Let $f : \mathbb{N} \to \{0, 1\}$ be a function—a function that belongs to the universe V. Define the set D_f by

$$D_f = \{p \in P \mid \text{for some } n \in \operatorname{dom} p,\ f(n) \neq p(n)\}.$$

We verify that D_f is dense. Let $p \in P$. We obtain $d \in D_f$ with $d \leq p$. If there is already $n \in \operatorname{dom} p$ with $p(n) \neq f(n)$, we let $d = p$. Otherwise, define a function d as follows. Pick $n \in \mathbb{N}$ so that $n \notin \operatorname{dom} p$. Let $\operatorname{dom} d = \operatorname{dom} p \cup \{n\}$. Let $i = f(n)$. Define d on its domain by:

$$d(k) = \begin{cases} p(k) & \text{if } k \neq n \\ 1 - i & \text{if } k = n. \end{cases}$$

It is clear now that $\operatorname{dom} d \supseteq \operatorname{dom} p$ and that d and p agree on $\operatorname{dom} p$; it follows that $d \leq p$. Since, by design, $f(n) \neq d(n)$, it follows that $d \in D_f$. We have shown D_f is dense in P.

Now we are ready to define the characteristics of the generic set G. Recall that in the technology of forcing, we first define a partial order P that captures our intended outcome—in this case, to add a new function $\mathbb{N} \to \{0, 1\}$—and then we use P to expand the universe V to an "all possibiliites" kind of universe V^P. Recall that the elements of V^P are to be considered as *potential sets*; they are formally called *names* (and the reason for this terminology will be explained later). The second step is to collapse V^P to a universe of real sets. This collapsing step is achieved by introducing a special subset G of P, called a generic filter.

Definition (Generic Filter). Suppose $\mathcal{P} = (P, \leq)$ is a partial order. A subset G of P is a *filter* if

(a) G is nonempty;
(b) whenever $q \in P$ and $p \in G$ and $p \leq q$, it follows that $q \in G$;
(c) whenever $p, q \in G$, there exists $r \in G$ with $r \leq p$ and $r \leq q$.

A filter G in P is *generic* if, for every dense subset D of P, we have $G \cap D \neq \emptyset$.

The most important part of the definition of filter is (c). In the present context, in which P is the set of all finite partial functions $\mathbb{N} \to \{0, 1\}$, part (c) of the definition ensures that we can form the union of the functions belonging to G to form a new function. In particular, part (c) tells us that the functions that belong to G are all compatible. To see this, suppose $p, q \in G$ and let $r \in G$ be such that $r \leq p$ and $r \leq q$. This means that r agrees with p on $\operatorname{dom} p$ and with q on $\operatorname{dom} q$. But this is possible only if p and q already agree on the common part of their domains; in other words, p and q must be compatible.

Part (b) of the definition tells us that G contains the "larger" elements of P. In forcing, we always assume that our partially ordered set has a largest element; this largest element is by convention denoted 1. Therefore, for every $p \in P$, we have $p \leq 1$. By part (b), every generic filter contains the maximum element 1 of P.

Part (c) guarantees that, if we let $g = \bigcup G$, then g will indeed be a function. However, parts (a)–(c) do not yet guarantee that the domain of g is $\mathbb{N}$ or that g is "new" in the sense that it does not belong to V. It is the *genericity* of G that guarantees these things (mentioned in the last part of the definition). We take a moment to do the verifications.

> Suppose P is the set of all finite partial functions, as described above. Suppose G is a generic filter in P. Then the following two claims hold:
>
> **Claim 1**. $\operatorname{dom} g = \mathbb{N}$.
>
> **Claim 2**. $g \notin V$.
>
> **Proof of Claim 1**. We must show that for each $n \in \mathbb{N}$, $n \in \operatorname{dom} g$. Let $n \in \mathbb{N}$. Let D_n be the set defined in Dense Set Example 1: $D_n = \{p \in P \mid n \in \operatorname{dom} p\}$. We have already shown D_n is dense. By genericity of G, $G \cap D_n \neq \emptyset$. Let $q \in G \cap D$. Since g is obtained by forming the union of all the functions in G, and since q is one of those functions, it follows that $\operatorname{dom} q \subseteq \operatorname{dom} g$. Since $n \in \operatorname{dom} q$, we are done. We have shown that each $n \in \mathbb{N}$ belongs to $\operatorname{dom} g$.
>
> **Proof of Claim 2**. Let $f : \mathbb{N} \to 2$ be a function in V. We must show that $g \neq f$. Let D_f be the set defined in Dense Set Example 3: $D_f = \{p \in P \mid \text{for some } n \in \operatorname{dom} p,\ f(n) \neq p(n)\}$. We have already shown that D_f is dense in P. By genericity of G, it follows that $G \cap D_f \neq \emptyset$. Let $p \in G \cap D_f$. Since g is obtained by forming the union of all the functions in G, and since p is one of those functions, it follows that p and g agree on $\operatorname{dom} p$. Since $p \in G$, there is $n \in \operatorname{dom} p$ such that $p(n) \neq f(n)$; it follows that $g(n) \neq f(n)$ and therefore that $g \neq f$. We have shown that g is different from every function $\mathbb{N} \to \{0, 1\}$ in the universe V.

Claims 1 and 2 together are the essence of the proof that, using P for forcing, we are adding a new function $\mathbb{N} \to \{0, 1\}$ to the universe V. We give a summary of the proof. We begin with the partial order P consisting of all finite partial functions and obtain the universe V^P of names. If we can find a $G \subseteq P$ that is a generic filter in P, then we use the forcing technology to collapse V^P to a new universe $V[G]$ that includes all sets in V and also contains the new set G. In $V[G]$ we now take the union of the elements of G: We let $g = \bigcup G$; since $V[G]$ is also a model of the ZFC axioms, forming the union of a set in $V[G]$ yields another set in $V[G]$, so $g \in V[G]$. Since G is a filter, g is a function; since G is generic, g is a function from $\mathbb{N}$ to $\{0, 1\}$ and $g \notin V$. We have added a new function $\mathbb{N} \to \{0, 1\}$, and therefore a new real number, to the universe.

Still, one may wonder whether such a generic filter G really does exist. And there is good reason to be skeptical because V is supposed to already contain all possible sets; so the claim that there could exist a set G that is *not* in the universe appears unjustified.

Set theorists have developed a number of ways of justifying existence of a generic filter G in forcing arguments. We spend a moment to discuss one approach[20] that is reasonably accessible, but once we have given this explanation, we will return to

[20] A different approach is discussed in Appendix II.

the view that we are adding a new set G to V in order to produce the universe $V[G]$; this will be the perspective for the rest of this article.

Since the forcing methodology is designed to establish consistency of a statement (like the statement S that asserts "$\mathbb{R}$ contains at least ω_2 real numbers"), our starting universe does not need to be V itself; it is possible to start with any universe that satisfies the axioms of ZFC. A remarkable discovery that was made over 100 years ago by the mathematician Leopold Löwenheim is that if there is a universe of sets at all—that is, if there is a *model* of the ZFC axioms—there must be such a universe whose size is just ω (the same size as the set of natural numbers); such models are called *countable models of* ZFC. The basic logic we wish to follow is that, if ZFC is consistent, then ZFC has a model, and by Löwenheim's result, it has a *countable* model. Then, by forcing, one obtains a model of $\mathrm{ZFC}+S$, where S is the statement whose consistency we are aiming to establish.

How then does it help to have a *countable* model of ZFC as a starting point? Let's follow the outline of the argument we gave earlier under this new assumption. Let M be a countable model of ZFC. This universe M does not "know" it is countable; there is no way to prove within M that there is a one-one correspondence between M and $\mathbb{N}$. But we, as dwellers in the real universe V, know about such a one-one correspondence; this is how we know that M is countable. Likewise, M believes that its version of the real number line $\mathbb{R}$ has size greater than ω; M must believe this because M is a model of ZFC and from ZFC we can prove the $\mathbb{R}$ has size greater than ω. But it seems strange that a universe that has size only ω could contain a set having size greater than ω. This apparent paradox[21] is resolved by the observation that M does not contain a one-one correspondence between $\mathbb{N}$ and $\mathbb{R}$, though, we, as dwellers in the real universe V, can obtain such a correspondence.

We carry out our forcing logic for adding a new real, using M as the starting universe. Define P in M as we did before in V: P is the set of all finite partial functions (as seen from the point of view of M). We wish to obtain a generic filter in P. For added generality, we will show a bit more. We will show that for any $q \in P$, we can obtain a generic filter G in P such that $q \in G$. We notice that since M has size ω, the number of dense subsets of P that lie in M is at most ω (one can show that the number of dense subsets is *exactly* ω). We can therefore enumerate those dense sets: We let $D_1, D_2, \ldots, D_n, \ldots$ denote the list of all sets that are, in M, dense subsets of P. We obtain a sequence $p_0, p_1, p_2, \ldots$ of elements of P in the following way. Let $p_0 = q$. Let $p_1 \in D_1$ be such that $p_1 \leq p_0$; since D_1 is dense, we can find such an element p_1. Let $p_2 \in D_2$ be such that $p_2 \leq p_1$; this is possible because D_2 is dense. We may continue in this way to arrive at a sequence $p_0 \leq p_1 \leq p_2 \leq \cdots$ with $p_n \in D_n$ for each n. We may now define G to be the "upward closure" of this sequence $p_0, p_1, p_2, \ldots$. More formally, we define G as follows:

$$G = \{r \in P \mid \text{for some } n \in \mathbb{N},\, p_n \leq r\}.$$

It is a straightforward exercise to show that G is indeed a filter. And since we have explicitly placed each of $p_1, p_2, \ldots$ into G, it follows that G meets every dense subset of P that belongs to M. And finally, since $p_0 \in G$ as well, we have guaranteed that our starting function $q = p_0$ also is in G.

[21] This paradox is known in the literature as *Skolem's Paradox.*

When we expand our starting universe M to a universe of names, we denote this new universe M^P; the universe of names in this case is defined in M rather than in V. And when we collapse M^P, we obtain a new model $M[G]$ of set theory (which, as it happens, is also countable). As before, G is a member of $M[G]$ (but not of M) and so, in $M[G]$, we can obtain the union g: that is, in $M[G]$ we may define $g = \bigcup G$. The same reasoning as before shows that g is a function from $\mathbb{N}$ to $\{0,1\}$ and that g is *new* in the sense that $g \notin M$.

5.2. **Using Forcing to Prove That CH Is Consistently False.** In order to prove CH is consistently false, one can do another forcing argument, similar to the one given in the previous paragraphs, only now we explicitly add ω_2 new reals to the universe instead of just one. More precisely, in the forcing extension $V[G]$ we will have a collection $\{g_x \mid x \in X\}$ of functions, where X has size ω_2 (in V), and, for each $x \in X$, $g_x : \mathbb{N} \to \{0,1\}$.

The partial order Q that accomplishes this for us is defined as follows: Let X be a set in V that has size ω_2. Then let

$$Q = \{p \mid p \text{ is a function, } \operatorname{dom} p \subseteq X \times \mathbb{N}, \operatorname{dom} p \text{ is finite, and } \operatorname{ran} p \subseteq \{0,1\}\}.$$

For all[22] $p, q \in Q$, we declare $p \leq q$ if and only if $\operatorname{dom} p \supseteq \operatorname{dom} q$ and p and q agree on $\operatorname{dom} q$.

Now suppose G is a generic filter for Q and, once again, let $g = \bigcup G$. Since G (being a subset of Q) consists of finite partial functions having domain a finite subset of $X \times \mathbb{N}$, and since the functions in G are compatible, forming the union of the elements of G produces a function whose domain is the union of all domains of elements of G—in particular, $\operatorname{dom} g \subseteq X \times \mathbb{N}$. As in the case of adding one real, we need to use genericity of G to verify that $\operatorname{dom} g = X \times \mathbb{N}$, and we will verify this point below.

Working in $V[G]$, we define, for each $x \in X$, a function $g_x : \mathbb{N} \to \{0,1\}$, defined by

$$g_x(n) = g((x,n)).$$

Using genericity of G again, we will show that all of these functions are distinct. Since distinct functions $\mathbb{N} \to \{0,1\}$ produce distinct real numbers using the formula in equation (5.1), our work provides us with a collection of functions from $\mathbb{N}$ to $\{0,1\}$ in $V[G]$ that is indexed by the set X. Genericity of G will allow us to prove

[22] The set $X \times \mathbb{N}$ is defined to be the set of all ordered pairs (x,n) for which $n \in \mathbb{N}$ and $x \in X$. In symbols, this set is defined as follows:

$$X \times \mathbb{N} = \{(x,n) \mid n \in \mathbb{N} \text{ and } x \in X\}.$$

It may be helpful here to consider a typical element of Q. A finite subset A of $X \times \mathbb{N}$ could be written as $A = \{(x_1,n_1),(x_2,n_2),\ldots,(x_k,n_k)\}$. An element p of Q with domain A would assign either 0 or 1 to each of the pairs in A. We could have, for example,

$$\begin{array}{rcl} p((x_1,n_1)) & = & 1 \\ p((x_2,n_2)) & = & 0 \\ \cdots & & \cdots \\ p((x_k,n_k)) & = & 1. \end{array}$$

that, for each $x \in X, g_x \notin V$. Since the size of X is ω_2, it appears that we have shown there are at least ω_2 real numbers in $V[G]$; in particular, that the Continuum Hypothesis is false in $V[G]$.

However, for this conclusion to be valid, there is one subtle additional point that needs to be verified: Although it is true in our starting universe V that X has size ω_2, in the expansion from V to $V[G]$ something may have happened to the size of X. It is conceivable that, for some set Y in V for which the size of Y is less than the size of X, our forcing may have inadvertently added a new function $u : Y \to X$ with $\operatorname{ran} u = X$. In that case, although our list of functions would still be indexed by elements of X in $V[G]$, from the point of view of $V[G]$, X no longer has size ω_2; X would actually be of smaller size, and so would not be big enough to violate CH. One can show, however, that no such u is added by this forcing; in particular, the ω_2 as seen inside V will be the same as the ω_2 as seen in $V[G]$; a proof of this fact can be found in [13].

To complete the proof that in $V[G]$ we have a collection $\{g_x \mid x \in X\}$ of distinct functions $\mathbb{N} \to \{0,1\}$, we need to prove the following three claims.

Claim 1. In $V[G]$, for each $x \in X$, $\operatorname{dom} g_x = \mathbb{N}$.

Claim 2. In $V[G]$, for all $x, y \in X$ with $x \neq y$, there is $n = n_{x,y} \in \mathbb{N}$ such that $g_x(n) \neq g_y(n)$. In other words, the functions in the set $\{g_x \mid x \in X\}$ are all distinct.

Claim 3. For all $x \in X, g_x \notin V$.

Proof of Claim 1. We begin by defining the dense sets that will allow us to prove the claim. For each $x \in X$ and $n \in \mathbb{N}$, let $D_{x,n} = \{p \in Q \mid (x,n) \in \operatorname{dom} p\}$. We observe that $D_{x,n}$ is dense in Q: Given $q \in Q$, we find $d \in D_{x,n}$ for which $d \leq q$. If $(x,n) \in \operatorname{dom} q$, we let $d = q$. If $(x,n) \notin \operatorname{dom} q$, we define d so that $\operatorname{dom} d = \operatorname{dom} q \cup \{(x,n)\}$ and define d by

$$d((y,m)) = \begin{cases} q((y,m)) & \text{if } (y,m) \neq (x,n) \\ 1 & \text{if } (y,m) = (x,n). \end{cases}$$

In both cases, $d \leq q$ and $(x,n) \in \operatorname{dom} d$. We have shown that $D_{x,n}$ is dense in Q.

Let $x \in X$; we show the $\operatorname{dom} g_x = \mathbb{N}$. Let $n \in \mathbb{N}$; we show $n \in \operatorname{dom} g_x$. Let $p \in G \cap D_{x,n}$. Then $(x,n) \in \operatorname{dom} p \subseteq \operatorname{dom} g$. By the definition of g_x, it follows that $n \in \operatorname{dom} g_x$.

Proof of Claim 2. To see that the functions g_x are all distinct, we obtain the necessary dense sets. Let x and y be distinct elements of X. Define $D_{x,y}$ by

$$D_{x,y} = \{p \in Q \mid \text{for some } n \in \mathbb{N}, p((x,n)) \neq p((y,n))\}.$$

We verify that $D_{x,y}$ is dense. Let $p \in Q$. Write elements of $\operatorname{dom} p$ as follows: $\operatorname{dom} p = \{(x_1, n_1), (x_2, n_2), \ldots, (x_k, n_k)\}$. Let $n \in \mathbb{N}$ be greater than each of $n_1, n_2, \ldots, n_k$. We obtain an element d of Q that is less than or equal to p. Let $\operatorname{dom} d = \operatorname{dom} p \cup \{(x, n), (y, n)\}$. Define d on its domain as follows:

$$d((u,r)) = \begin{cases} p((u,r)) & \text{if } (u,r) \in \operatorname{dom} p \\ 0 & \text{if } (u,r) = (x,n) \\ 1 & \text{if } (u,r) = (y,n). \end{cases}$$

Clearly $d \le p$. Also, $d((x,n)) \ne d((y,n))$, so $d \in D_{x,y}$. We have shown $D_{x,y}$ is dense.

To complete the proof, we show that for any $x, y \in X$ with $x \ne y$, there is $n \in \mathbb{N}$ such that $g_x(n) \ne g_y(n)$. Given x and y, let $p \in D_{x,y} \cap G$. Since $p \in D_{x,y}$, we can find $n \in \mathbb{N}$ with $p((x,n)) \ne p((y,n))$. Since $p \in G$, it follows that g agrees with p on the domain of p. Therefore,

$$g_x(n) = g((x,n)) = p((x,n)) \ne p((y,n)) = g(y,n) = g_y(n).$$

We have shown $g_x \ne g_y$.

Proof of Claim 3. Let $f : \mathbb{N} \to \{0,1\}$ be a function in V and let $x \in X$. We show that in $V[G]$, $g_x \ne f$. We begin with the necessary dense set. Let $D^f_x = \{p \in Q \mid \text{for some } n \in \mathbb{N}, p((x,n)) \ne f(n)\}$. We show D^f_x is dense. Let $p \in Q$. We obtain $d \le p$ as follows. If there is $n \in \mathbb{N}$ so that $p((x,n)) \ne f(n)$, we let $d = p$. Otherwise, write $\operatorname{dom} p = \{(x_1, n_1), (x_2, n_2), \ldots, (x_k, n_k)\}$. Let $n \in \mathbb{N}$ be greater than each of $n_1, n_2, \ldots, n_k$. We define d. Let $\operatorname{dom} d = \operatorname{dom} p \cup \{(x,n)\}$ and define d on its domain as follows:

$$d((u,r)) = \begin{cases} p((u,r)) & \text{if } (u,r) \in \operatorname{dom} p \\ 1 - f(n) & \text{if } (u,r) = (x,n). \end{cases}$$

Clearly $d \le p$. Also, $d((x,n)) \ne f(n)$, so $d \in D^f_x$. We have shown D^f_x is dense.

To complete the proof, we show that for every $f : \mathbb{N} \to 2$ that belongs to V and each $x \in X$, we have, in $V[G]$, that $g_x \ne f$. Given f and x, let $p \in D^f_x \cap G$. Let $n \in \mathbb{N}$ be such that $p((x,n)) \ne f(n)$. Then since $p \in G$, g agrees with p on the domain of p. Therefore, $g_x(n) = g(x,n) \ne f(n)$. We have shown that $g_x \ne f$, as required.

5.3. **More on the Universe V^P of Names and the Forcing Relation.** In order to understand why forcing does not destroy the Wholeness Axiom, we need to probe a bit more deeply into the role of the universe V^P of names in the forcing technology and introduce the forcing relation, which allows us to prove things about the forcing extension $V[G]$ by studying the universe V^P of P-names, working entirely within V.

We will develop points about the forcing technology with reference to the partial order P of all finite partial functions from $\mathbb{N}$ to $\{0,1\}$; the principles we identify here generalize to all partial orders used for forcing.

We begin with a convention that we will observe for the rest of the article. In forcing arguments, the partial order that is used will always have a largest element, which is usually denoted 1. In the case of the partial order P of finite partial functions, the largest element[23] is the empty function: the function e whose domain is the empty set; this is the case because, for any partial function $p : A \to \{0,1\}$, we have $A \supseteq \emptyset$ and, vacuously, p and e agree on the domain of e. It follows that $p \leq e$. For the rest of this discussion, we will let 1 denote this largest element e of P.

Let us recall that the first step in the forcing methodology, once a partial order has been selected, is to expand the universe V to a universe of names. Using the partial order P, the universe of names is denoted V^P. The elements of V^P are called, more formally, *P-names* and are typically denoted with Greek letters like σ (*sigma*) and τ (*tau*). The universe V^P of P-names, like the universe V itself, is built up in stages: $V^P = V_0^P \bigcup V_1^P \bigcup V_2^P \bigcup \cdots$. We describe the first few stages in the build-up of V^P.

A P-name τ in a particular stage[24] V_{n+1}^P is a set of ordered pairs of the form (σ, p), where σ is a name that belongs to the previous stage V_n^P. More precisely,

$$\begin{aligned} V_0^P &= \emptyset \\ V_{n+1}^P &= \{A \mid A \text{ is a set of pairs } (\sigma,p) \text{ where } \sigma \in V_n^P \text{ and } p \in P\}. \end{aligned}$$

One can check[25] that $V_1^P = \{\emptyset\}$, but subsequent stages get bigger very quickly. Already, V_2^P has size at least ω_1, consisting of all possible sets of the form $\{(\emptyset, p) \mid p \in B\}$ for $B \subseteq P$. An example of a P-name τ belonging to stage V_{n+1}^P would be $\tau = \{(\sigma,p),(\mu,q),(\delta,r)\}$, where σ, μ, and δ all belong to V_n^P.

The forcing extension $V[G]$ that is obtained by collapsing V^P using a generic filter G is obtained by *collapsing* or *evaluating* each P-name using G. The evaluation of a P-name τ is denoted τ_G; the object τ_G is an actual set in the new universe $V[G]$ (while τ itself is only a name for a set).[26] The universe $V[G]$ is defined to be the collection of all such evaluations of names. More precisely, we have:

$$V[G] = \{\tau_G \mid \tau \text{ is a } P\text{-name}\}.$$

To compute τ_G from τ, we make use of the evaluations σ_G of P-names that occur in τ, which necessarily occur at earlier stages in V^P. Here is a precise definition:

$$\tau_G = \{\sigma_G \mid \text{for some } p \in G, (\sigma,p) \in \tau\}.$$

[23]If p and q are elements of any partial order and $p \leq q$, one says that p is *smaller than* q and that q is *larger than* p. Recall however that in the case of the partial order P, $p \leq q$ implies that $\operatorname{dom} p \supseteq \operatorname{dom} q$, so, although p has a larger domain, it is *smaller* than q according to the relation $\leq$.

[24]We are indexing the stages here with the whole numbers $0, 1, 2, \ldots$, but in reality there are many stages beyond those that can be indexed in this way; indices that lie beyond the whole numbers are called *infinite ordinals*. This topic is beyond the scope of the present article but is covered in [7].

[25]This is easier to see if we observe that $V_{n+1}^P = \mathcal{P}(V_n^P \times P)$, so that

$$V_1^P = \mathcal{P}(\emptyset \times P) = \mathcal{P}(\emptyset) = \{\emptyset\}.$$

[26]Formally, P-names are also represented as sets inside V, but it is more useful to think of them as being entities living in an expanded world of *potential* sets.

Returning to our earlier example, suppose $\tau = \{(\sigma, p), (\mu, q), (\delta, r)\}$ and $\tau \in V^P_{n+1}$. The P-names σ, μ, and δ all belong to V^P_n, so we may assume we have already evaluated each of these with G to obtain actual sets σ_G, μ_G, and δ_G. Then the set τ_G will consist of some or all of the sets $\sigma_G, \mu_G, \delta_G$. To determine which of these are chosen to belong to τ_G, the second components p, q, r of the pairs $(\sigma, p), (\mu, q), (\delta, r)$ belonging to τ are examined; those that belong to the generic filter G are chosen. For instance, assume that $p \in G$ and $r \in G$ but $q \notin G$. Then $\tau_G = \{\sigma_G, \delta_G\}$. This example shows that the partial order component p of a pair like (σ, p) belonging to τ tells us the "likelihood" that the evaluation of σ will end up being an element of τ_G. Elements p of the partial order that are larger in the ordering and closer to 1 are more likely to belong to a generic filter, and, as we have just seen, if the element p in (σ, p) does end up in G, we conclude that $\sigma_G \in \tau_G$ in the forcing extension $V[G]$.

There are some simple P-names that are always defined the same way in any forcing argument. These simple names provide a way of representing sets in the starting universe V as P-names, which will then always be transformed back to their original value as a set in $V[G]$. For any set x in V, we define a P-name $\check{x}$ (pronounced "x check") as follows:

$$\check{x} = \{(\check{y}, 1) \mid y \in x\}.$$

This is a recursive definition; $\check{x}$ is defined under the assumption that names at earlier stages have already been defined.

We illustrate the definition with an example. Let us note that $\check{\emptyset} = \emptyset$ and if $x = \{\emptyset\}$, then $\check{x} = \{(\check{\emptyset}, 1)\}$. Then if we evaluate these with a generic filter G, we obtain $\check{\emptyset}_G = \emptyset$ and, since we necessarily have $1 \in G$ for any generic filter G,

$$\check{x}_G = \{\sigma_G \mid \text{for some } p \in G, (\sigma, p) \in \check{x}\} = \{\check{\emptyset}_G\} = \{\emptyset\}.$$

In each case, the starting set $x \in V$ is first transformed to a P-name $\check{x}$ and then the evaluation $\check{x}_G$ turns out to be[27] precisely x. For this reason, we may conclude that, for any forcing extension $V[G]$, we have $V \subseteq V[G]$.

An important part of the intuition about the structure of the universe V^P of P-names is that the partial order P represents an expanded idea of *truth values*. The usual truth values are *false* and *true*, which we could think of as corresponding to the two-element set $\{0, 1\}$. But in the forcing methodology, when we expand from V to V^P, we are thinking of the various elements of P as representing "partial truth values." One consequence is that, in V^P, it is possible for a P-name to represent different sets depending on which generic filter G is used to evaluate it.

We give an example of a P-name that could be evaluated to either of the following two sets, depending on the choice of G: $\{\emptyset\}$ or $\{\{\emptyset\}\}$. Let $x = \{\emptyset\}$. Consider the following two elements p and q of P:

$$\begin{aligned} \operatorname{dom} p &= \{1\} \text{ and } p(1) = 0 \\ \operatorname{dom} q &= \{1\} \text{ and } q(1) = 1. \end{aligned}$$

Let us build a "hybrid" name σ as follows:

$$\sigma = \{(\check{\emptyset}, p), (\check{x}, q)\}.$$

[27] In set theory, the number 0 is defined to be the empty set $\emptyset$ and the number 1 is defined to be the set $\{0\}$. We have just shown that $(\check{0})_G = 0$ and $(\check{1})_G = 1$.

We show that in some forcing extensions, $\sigma_G = \emptyset$, while in others $\sigma_G = x$.

Let $G \subseteq P$ be generic. Let $D = \{s \in P \mid \text{either } s \leq p \text{ or } s \leq q\}$. We show that D is a dense set: Let $r \in P$. We obtain $d \in D$ with $d \leq r$ as follows. If $1 \notin \operatorname{dom} r$, then obtain $d \leq r$ by defining $\operatorname{dom} d = \operatorname{dom} r \cup \{1\}$ and letting $d(1) = 0$. Then $d \leq p$, and so $d \in D$. If $1 \in \operatorname{dom} r$, then if $r(1) = 0$, it follows $r \leq p$, while if $r(1) = 1$, we have that $r \leq q$; either way, we let $d = r$. Then once again $d \in D$. We have shown D is dense.

Since G is generic, there is $r \in G \cap D$, and there are two possibilities: (a) $r \leq p$ or (b) $r \leq q$. In case (a), because $r \in G$ and $r \leq p$, it follows that $p \in G$. When we compute σ_G, we obtain (because $(\check{\emptyset}, 1) \in \sigma$ and $1 \in G$):

$$\sigma_G = \{\delta_G \mid \text{for some } r \in G, (\delta, r) \in \sigma\} = \{\check{\emptyset}_G\} = \{\emptyset\}.$$

In case (b), because $r \in G$ and $r \leq q$, it follows that $q \in G$. When we compute σ_G, we obtain (because $(\check{x}, 1) \in \sigma$ and $1 \in G$):

$$\sigma_G = \{\delta_G \mid \text{for some } r \in G, (\delta, r) \in \sigma\} = \{\check{x}_G\} = \{x\} = \{\{\emptyset\}\}.$$

Therefore, σ names both sets $\{\emptyset\}$ and $\{\{\emptyset\}\}$ at the same time; which of these ends up being the real value of σ_G depends on how V^P is collapsed[28]—that is, it depends on the choice of G. This tells us that the multiplicity of truth values that we find represented by the multiple elements of the partial order P is collapsed to the standard *two* truth values *true* and *false* by evaluating with G.

> **Exercise**. Give an example of a P-name that could be evaluated to any of the following three sets, depending on the choice of G: $\emptyset, \{\emptyset\}$, or $\{\{\emptyset\}\}$. Let $x = \{\emptyset\}$ and let $y = \{\{\emptyset\}\}$. *Hint.* Consider the following three elements p, q, and r of P:
>
> $$\begin{aligned} \operatorname{dom} p &= \{1\} \text{ and } p(1) = 0 \\ \operatorname{dom} q &= \{1, 2\} \text{ and } q(1) = 1 \text{ and } q(2) = 0 \\ \operatorname{dom} r &= \{1, 2\} \text{ and } r(1) = 1 \text{ and } r(2) = 1. \end{aligned}$$

The example in the previous paragraph and the exercise above highlight the fact that a partial order component p of a pair (σ, p) in a name τ can be viewed as a "partial truth value" that tells us the likelihood that the matched P-name σ actually will belong to τ_G after the collapse.

These examples suggest another aspect of forcing that we have not discussed so far: the *forcing relation*. For any formula ϕ of set theory,[29] we replace the variables in the formula with P-names; the resulting expression is a *sentence in the forcing language*. An example of a sentence in the forcing language would be $\sigma \in \tau$. Here, we have started with the formula $x \in y$ and replaced x with the P-name σ and y

[28]There is an obvious parallel here to quantum mechanics: The wave function for a particle is in a superposition of many states; the collapse of the wave function determines a particular state of the particle. By analogy, in this example, one can think of σ as being a "superposition" of the two sets $\{\emptyset\}$ and $\{\{\emptyset\}\}$. Though this is only an analogy, it provides a way of understanding the way in which elements of V^P are to be considered "potential sets."

[29]Formulas of set theory are mathematical statements involving variables $x, y, z, \ldots$ and the membership relation $\in$, such as $x \in y$ and $y \in z$. More complicated relationships are built up using $\in$. For instance, another formula is $\forall z\,(z \in x \text{ implies } z \in y)$; this formula says "for all sets z, if z belongs to x, then z belongs to y." In other words, the formula asserts "x is a subset of y."

with the P-name τ. We can ask whether ϕ holds true in $V[G]$ after evaluating the names that occur in ϕ; that is, is it true that $\sigma_G \in \tau_G$ inside $V[G]$? Of course, as we have seen, the answer depends on the structure of the names σ and τ and the choice of G.

Now, given any $p \in P$ and any sentence ϕ in the forcing language for P, we can ask whether *p forces ϕ* (notation: $p \Vdash \phi$). Here is a definition.

> **Definition.** Give a sentence ϕ in the forcing language for P and given $p \in P$, we say *p forces ϕ*, and write $p \Vdash \phi$, if, for every generic filter G in P that contains p as an element, it follows that ϕ holds in $V[G]$.

Thus, in the example above (preceding the Exercise), we may say that $p \Vdash \check{\emptyset} \in \tau$ while $q \Vdash \check{x} \in \tau$.

Each instance of the forcing relation $p \Vdash \phi$ tells us that p gives enough information for us to conclude that ϕ will hold in each forcing extension $V[G]$, provided that $p \in G$. And expressions of the form $1 \Vdash \phi$ tell us that ϕ is true in every extension $V[G]$, since it is always true that $1 \in G$.

There is one other important canonical P-name that should be mentioned: There is a P-name that is always realized in a forcing extension $V[G]$ as the set G itself. We denote this name Γ (*gamma*, upper case); it is defined as follows.

$$\Gamma = \{(\check{p}, p) \mid p \in P\}.$$

Let G be any generic filter in P; we evaluate Γ with G:

$$\begin{aligned}
\Gamma_G &= \{\sigma_G \mid \text{for some } q \in G, (\sigma, q) \in \Gamma\} \\
&= \{(\check{p})_G \mid \text{for some } q \in G, (\check{p}, q) \in \Gamma\} \\
&= \{(\check{p})_G \mid (\check{p}, p) \in \Gamma\} \\
&= \{p \mid p \in G\} \\
&= G.
\end{aligned}$$

We have shown that Γ is a name for G; the P-name Γ is realized as the set G in the forcing extension $V[G]$ no matter which generic filter G is used.[30]

As an easy application of these observations, we can show that, for any $p \in P$, we have $p \Vdash \check{p} \in \Gamma$. This says, intuitively speaking, that "p forces it to be the case that p is the *true* truth value." To prove this, suppose $p \in G$. Then in $V[G]$ it follows that $\check{p}_G \in \Gamma_G$, that is, $p \in G$.

This observation sheds new light on the forcing argument given earlier in which we showed how it is possible to add a new function $g : \mathbb{N} \to \{0, 1\}$ by forcing with the set P of finite partial functions into $\{0, 1\}$. Let γ (*gamma*, lower case) be a P-name for the union of a generic filter G; this means that γ is defined so that,

[30]We note here that, although in some treatments of forcing, existence of the generic set G is problematic—for instance, if our starting universe is V itself, the best one can do is to show that existence of G is *consistent* (luckily, consistency of existence of G is still good enough for any forcing argument)—the canonical *name* Γ for G *always* exists. This fact lends credence to the viewpoint that the right way to look at the forcing methodology is the one given in Appendix II; namely, that the universe of names is *never* collapsed (so that existence of G never becomes an issue) but is used on its own as the way of establishing the desired consistency results.

for any generic filter G, γ_G is realized in $V[G]$ as the set $\bigcup G$, which we have been calling g in our earlier arguments.[31] For any $p \in P$, we have

$$p \Vdash \text{“}\gamma \text{ agrees with } \check{p} \text{ on the domain of } \check{p}\text{”}. \tag{5.2}$$

This is simply a restatement of the fact that if, in $V[G]$, we let $g = \bigcup G$, then for any $p \in G$, g must agree with p on its domain. But using the forcing relation in this way allows us to see that, though the starting universe V does not know about the new real g, it does know (by way of the forcing relation) that a certain name (γ) is forced to be a new real, even though V does not know how that name could be realized as an actual set.[32] For this reason, the partial order P truly embodies the “intention” to add a new function $g : \mathbb{N} \to \{0, 1\}$: Each element of P serves as a potential piece of g and, even from the perspective of V, it is clear that the presence of any generic filter G would cause the the collective contributions of each of its elements to “crystallize” and form a new function $\mathbb{N} \to \{0, 1\}$.

This observation about elements of P and the new function $g : \mathbb{N} \to \{0, 1\}$ is a consequence of a more general forcing fact, which we will now state.[33] It says that statements that are true in $V[G]$ are always *forced* to be true by some $p \in G$.

> **Forcing Theorem**. *If $\phi(\tau_1, \tau_2, \ldots, \tau_n)$ is any sentence in the forcing language for P and if G is any generic filter in P, then $\phi((\tau_1)_G, (\tau_2)_G, \ldots, (\tau_n)_G)$ holds in $V[G]$ if and only if, for some $p \in G$, $p \Vdash \phi(\tau_1, \tau_2, \ldots, \tau_n)$.*

Proof of Indestructibility of the Wholeness Axiom. We recall that the acronym WA stands for the Wholeness Axiom. Saying that forcing does not “destroy” WA means that, if WA holds true in our starting universe V, then, no matter which partial order we use to do a forcing argument, it will follow that, in the forcing extension $V[G]$, WA will continue to hold true. We will describe the proof of this fact for the case in which the starting partial order is the set of finite partial functions into $\{0, 1\}$ that we used above to add a new real to the universe.

---

[31] That such a name exists is derivable from the following fact, which the ambitious reader may wish to verify.

> **Fact.** Given a P-name τ, the following P-name π has the property that for any generic G in P, $\pi_G = \bigcup \tau_G$.
>
> $$\pi = \{(\delta, p) \mid \exists(\sigma, q) \in \tau\, \exists r\, ((\delta, r) \in \sigma \text{ and } p \le r \text{ and } p \le q)\}.$$

[32] The forcing relation can be defined without reliance on the generic filter G, though we have not shown here how this can be done. One says that the forcing relation $\Vdash$ is *definable in V*. Intuitively speaking, this means that all the assertions one can make with $\Vdash$ are known within V, whereas statements involving the generic filter G are not comprehensible to V. Therefore, in our earlier discussion, where we observed that whenever $p \in G$, the new function g must agree with p on its domain, the observation was taking place within the extension $V[G]$, since this is where G and g live. With the help of the forcing relation, we can make the same observations about G and g from the perspective of the starting universe V, as in equation (5.2), about arbitrarily close approximations to g.

[33] Having developed more of the forcing technology, we could at this point describe an alternative approach to forcing that does not make use of a generic filter at all. This approach is developed in Appendix II.

Therefore, we begin the argument by assuming at the outset that WA holds true in our starting universe V. This means that there is a function $j : V \to V$ that is an elementary embedding; recall that this means that all relationships that are true in V are preserved by j. We remind the reader of the two examples mentioned earlier of the strong preservation guaranteed by j: First, assume X and Y are sets in V and we have that $X \in Y$. Then elementarity of j tells us that $j(X) \in j(Y)$; one says that j *preserves the membership relation.* For the second example, assume Y and Z are sets in V, and Z is the set of all subsets of Y. Then elementarity of j ensures that $j(Z)$ is the set of all subsets of $j(Y)$. Written more succinctly, if $Z = \mathcal{P}(Y)$, then $j(Z) = \mathcal{P}(j(Y))$. One says that j *preserves the power set operation.*

The most general way of stating that j is an elementary embedding is to say that, for any formula $\phi(x_1, x_2, \ldots, x_n)$ that talks about sets (we just considered two such formulas: $x_1 \in x_2$ and $x_3 = \mathcal{P}(x_2)$) and, for any sets $X_1, X_2, \ldots, X_n$ in V, if the formula $\phi(X_1, X_2, \ldots, X_n)$ holds true in V, then the formula $\phi(j(X_1), j(X_2), \ldots, j(X_n))$ also holds in V.

Given such a $j : V \to V$, we want to show that, if we carry out a forcing construction to obtain a forcing extension $V[G]$, then there must be another elementary embedding $k : V[G] \to V[G]$. This will (nearly) establish that WA continues to hold in the new universe. We indicate how it is possible to obtain such an embedding k in the special case in which the partial order used for forcing is the set P of all finite partial functions from $\mathbb{N}$ into $\{0, 1\}$, used earlier to add a new real to the universe.

Let us recall that the Wholeness Axiom tells us not only that j is an elementary embedding, but also that there is a special infinite cardinal number κ that is the first cardinal to be moved by j. This means that, for all cardinal numbers λ that are less than κ, we have $j(\lambda) = \lambda$; in other words, j does not move any of the cardinals that are smaller than κ. Using this fact, one can also prove that any set A that appears in the universe at any stage V_α, where $\alpha < \kappa$, is also fixed by j. In other words,[34]

$$\text{For all } A \in V_\kappa, \text{ we have that } j(A) = A. \tag{5.3}$$

We will make use of this fact in our upcoming argument.

Let us now repeat the forcing construction that we did earlier, using the set P of all finite partial functions as the partial order, to produce the extension $V[G]$. Our objective is to define an elementary embedding $k : V[G] \to V[G]$. Recall that each element of $V[G]$ is the realization τ_G of a P-name τ. Although we think of P-names as being only "potential sets," they are technically defined in terms of sets. Formally, every P-name belongs[35] to V.

[34] Note that if a set A belongs to V_κ, it must also belong to an earlier stage V_α for some $\alpha < \kappa$.

[35] One can rightly ask at this point, "So, *are* P-names sets, or are they *not* sets?" An analogy will help to clarify this point. Imagine a message that is written using a secret code. The same letters of the alphabet are used in the coded message, but the coded message is unreadable; it becomes readable after it is decoded. One can ask, "Is the coded message made up of letters of the alphabet or is it not?" The answer is that the message is certainly made of the letters of the alphabet—the same letters that are used in normal language—but the "words" formed by the letters do not make sense in their coded form. The words of the coded message become meaningful only after decoding. Likewise, the P-names are, technically, sets in the universe, but they are as if "encoded" sets—they make sense as sets after we decode them using a generic filter G. A P-name

Because P-names are, technically speaking, sets in V, we may apply the embedding j to them, just as we can apply j to any set in the universe. Since j preserves all properties and relationships, it follows that, for any P-name τ, the value $j(\tau)$ is a $j(P)$-name; that is, it is a name belonging to $V^{j(P)}$, obtained by using elements of the partial order $j(P)$ instead of P. However, considering the fact that P is a relatively small infinite set, consisting of functions defined on finite subsets of $\mathbb{N}$, mapped into $\{0,1\}$, it is not hard to show that P belongs to V_κ. Therefore, by(5.3), we must have $j(P) = P$. We conclude that, in fact, j maps P-names to other P-names (since a $j(P)$-name is now understood to be simply a P-name).

To define k on $V[G]$, we must find a way to specify where any given element τ_G of $V[G]$ is mapped by k. We intend for k to map elements of $V[G]$ to other elements of $V[G]$. Here is the technique to compute $k(\tau_G)$:

$$k(\tau_G) = j(\tau)_G. \tag{5.4}$$

Equation (5.4) is saying that, to define $k(\tau_G)$, we first apply j to the P-name τ; the value $j(\tau)$ is another P-name, and so it may be evaluated using G, to obtain $j(\tau)_G$. Therefore, defining k on $V[G]$ amounts to applying j to P-names and evaluating the results.

A key step in checking whether k is indeed an elementary embedding is to verify that the definition of k does not depend on the choice of names. Notice that there could be many different P-names whose evaluation produces the same set as τ_G. Suppose σ is one such P-name, so that we have $\sigma_G = \tau_G$. When we apply the rule given in (5.4) to compute $k(\sigma_G)$, certainly, we get $j(\sigma)_G$, but since σ_G and τ_G are the same set, k should map σ_G and τ_G to the same set. In other words, we expect the following to hold true:

$$k(\sigma_G) = k(\tau_G). \tag{5.5}$$

To establish equation (5.5), we are going to show the following:

$$\text{Whenever } p \in P \text{ and } p \Vdash \sigma = \tau, \text{ we also have } p \Vdash j(\sigma) = j(\tau). \tag{5.6}$$

We can almost obtain equation (5.6) simply by making use of elementarity of j: We know that if $p \Vdash \sigma = \tau$, then when we apply j, we must get $j(p) \Vdash j(\sigma) = j(\tau)$. One final step leads to the result: Just as P itself is so small that it belongs to V_κ, so likewise must each element of P belong to V_κ. Therefore, by (5.3) again, $j(p) = p$, and we immediately obtain equation (5.6).

Let us now verify that k has the needed property, namely, that $k(\sigma_G) = k(\tau_G)$ whenever $\sigma_G = \tau_G$. Since we have that $\sigma_G = \tau_G$ in $V[G]$, the Forcing Theorem tells us that this fact must have been forced; that is, for some $p \in G$, we have $p \Vdash \sigma = \tau$. But now equation (5.6) allows us to conclude that $p \Vdash j(\sigma) = j(\tau)$. Therefore, $j(\sigma)_G = j(\tau)_G$, and we have:

$$k(\sigma_G) = j(\sigma)_G = j(\tau)_G = k(\tau_G),$$

which establishes equation (5.5).

Establishing that our definition of k in equation (5.4) is legitimate, as we have just now done, also establishes the first step in the proof that k is an elementary

τ is like an encoded message; the evaluation τ_G of τ is like a decoded message. A P-name τ is technically a set, but its "meaning" as a set is obscured, whereas τ_G is a normal, familiar set.

embedding. To prove elementarity of k, we must show that $k : V[G] \to V[G]$ preserves the truth of *any formula*. What we have just shown is that k preserves the truth of very simple formulas of the form $\sigma_G = \tau_G$. More formally, we have shown

If the formula $\sigma_G = \tau_G$ holds in $V[G]$,
then the formula $k(\sigma_G) = k(\tau_G)$ holds in $V[G]$.

Similar reasoning shows that k preserves the truth of formulas of the form $\sigma_G \in \tau_G$. Repeating somewhat our work above, we may argue as follows to prove that this type of formula is also preserved by k. We first conclude, using elementarity of j and the fact that all elements of P belong to V_κ, that

(5.7) Whenever $p \Vdash \sigma \in \tau$, it follows that $p \Vdash j(\sigma) \in j(\tau)$.

Therefore, suppose that in $V[G]$ we have that $\sigma_G \in \tau_G$. This fact must have been forced by some $p \in G$; that is, it must be true that $p \Vdash \sigma \in \tau$. But then by equation (5.7), we have that $p \Vdash j(\sigma) \in j(\tau)$, and since $p \in G$, we conclude that

$$k(\sigma_G) = j(\sigma)_G \in j(\tau)_G = k(\tau_G).$$

To complete the proof that k is an elementary embedding, we must consider all possible formulas ϕ built up from these two basic types of formulas (one type is $\sigma_G = \tau_G$ and the other is $\sigma_G \in \tau_G$) using connectives *and*, *or*, *not*, and *implies*, as well as the quantifiers *there exists* and *for all*. To show that k preserves all such formulas, one argues by induction on the length of the formulas. For instance, one of the many formulas that need to be checked, for P-names σ, τ, and δ, is the following sentence ϕ, assuming that ϕ holds in $V[G]$:

$$\phi : \ \sigma_G = \tau_G \ \textit{and} \ \tau_G \in \delta_G.$$

The inductive reasoning here would allow us to assume that k preserves each component of the *and*-statement, namely, the two statements $\sigma_G = \tau_G$ and $\tau_G \in \delta_G$ (and in this example we have already explicitly shown that k preserves these), and we would then prove that the entire sentence ϕ is preserved by k.

We give a flavor of the induction argument that establishes that k preserves all formulas that hold in $V[G]$. We adopt the convention from mathematical logic to represent the connectives using the following symbols: $\wedge$ (and), $\vee$ (or), $\to$ (implies), $\exists$ (there exists), and $\forall$ (for all). Given formulas ϕ and ψ, and assuming k preserves each of them individually, one shows k preserves the formulas $\phi \wedge \psi$, $\phi \vee \psi$, $\phi \to \psi$, and so forth. For instance, suppose both of the sentences $\phi((\sigma_1)_G, \ldots, (\sigma_n)_G)$ and $\psi((\tau_1)_G, \ldots, (\tau_n)_G)$ hold in $V[G]$, and both are preserved by k. We show in this example that j preserves the truth of

$$\phi((\sigma_1)_G, \ldots, (\sigma_n)_G) \ \wedge \ \psi((\tau_1)_G, \ldots, (\tau_n)_G).$$

By the Forcing Theorem, there are p and q in G that force the truth of each of ϕ and ψ individually:

$$\begin{aligned} p &\Vdash \phi(\sigma_1, \ldots, \sigma_n) \\ q &\Vdash \psi(\tau_1, \ldots, \tau_n). \end{aligned}$$

Using the fact that G is a filter, we may conclude that there is $r \in G$ so that $r \leq p$ and $r \leq q$; it follows[36] that $r \Vdash \phi(\sigma_1, \ldots, \sigma_n) \wedge \psi(\tau_1, \ldots, \tau_n)$. Applying elementarity of j, we have $r \Vdash \phi(j(\sigma_1), \ldots, j(\sigma_n)) \wedge \psi(j(\tau_1), \ldots, j(\tau_n))$. By the definition of the forcing relation $\Vdash$, the sentence $\phi(j(\sigma_1)_G, \ldots, j(\sigma_n)_G) \wedge \psi(j(\tau_1)_G, \ldots, j(\tau_n)_G)$ holds in $V[G]$. Using the definition of k, we conclude that the following sentence holds in $V[G]$, as required:

$$\phi(k((\sigma_1)_G), \ldots, k((\sigma_n)_G) \ \wedge \ \psi(k((\tau_1)_G), \ldots, k((\tau_n)_G).$$

We have outlined the proof that $k : V[G] \to V[G]$ is an elementary embedding, but we have not quite finished proving that WA holds in $V[G]$. What remains to be proved is the last clause in the statement of WA: We must show that, for any set X in $V[G]$,

(5.8) The restriction $k \upharpoonright X : X \to k(X)$ is a set in the universe $V[G]$.

The fact that k is an elementary embedding is not enough to establish this additional property.

This final step in the proof can be simplified significantly. Rather than showing $k \upharpoonright X$ is a set for *every* set X in $V[G]$, it is enough to show that $k \upharpoonright Z_n$ is a set for each set in a special collection[37] $\{Z_0, Z_1, \ldots, Z_n, \ldots\}$. We state this fact as a Proposition, which we will prove below:

> **Proposition**. Assuming WA holds in V, there is a collection $\{Z_0, Z_1, Z_2, \ldots\}$ of sets in $V[G|$ so that the following statements are equivalent:
> (i) The map $k \upharpoonright X$ belongs to $V[G]$ whenever X belongs to $V[G]$.
> (ii) For each n, the map $k \upharpoonright Z_n$ belongs to $V[G]$.

In order to specify these special sets $Z_0, Z_1, \ldots$, we first need to introduce two simplifying notations.

> **Notation.**
> (A) *The critical sequence* $\kappa_0, \kappa_1, \kappa_2, \ldots$. The crtical sequence begins with κ (we let $\kappa_0 = \kappa$) and successive terms are obtained

[36] One step of reasoning has not been shown here: one needs to verify that for any formulas ϕ and ψ in the forcing language, and any $r \in P$, the following holds:

$$\big((r \Vdash \phi) \text{ and } (r \Vdash \psi)\big) \text{ if and only if } r \Vdash (\phi \wedge \psi).$$

This is an easy verification using the definition of the forcing relation $\Vdash$.

[37] Notice that the elements $Z_0, Z_1, Z_2, \ldots$ of this collection are indexed by the counting numbers; the set of counting numbers is a set of smallest possible infinite size. Therefore, checking $k \upharpoonright Z_n$ belongs to $V[G]$ for each Z_n in the collection is a much easier task than checking $k \upharpoonright X$ is a set for every set X in the universe.

by applying j. We have:

$$\begin{array}{rcl} \kappa_0 & = & \kappa \\ \kappa_1 & = & j(\kappa) \\ \kappa_2 & = & j(j(\kappa)) \\ . & & . \\ . & & . \\ . & & . \\ \kappa_{n+1} & = & j(\kappa_n) \\ . & & . \\ . & & . \\ . & & . \end{array}$$

(B) *The stages of* $V[G]$: $V[G]_0, V[G]_1, V[G]_2, \ldots$. Since $V[G]$ is a universe of sets—a model of ZFC—it is built up stage by stage in the same way as V. For each α, we denote the αth stage in the build-up of $V[G]$ by $V[G]_\alpha$. Here are the first few stages:[38]

$$\begin{array}{rcl} V[G]_0 & = & \emptyset \\ V[G]_1 & = & \mathcal{P}(\emptyset) = \{\emptyset\} \\ V[G]_2 & = & \mathcal{P}(V[G]_1) \\ . & & . \\ . & & . \\ . & & . \\ V[G]_{n+1} & = & \mathcal{P}(V[G]_n) \\ . & & . \\ . & & . \\ . & & . \end{array}$$

Regarding (A), we observe here that κ_n is not only the result of applying the nth iterate j^n of j to κ; the cardinal κ_n is also the output when the nth iterate k^n of k is applied to κ. In other words, we have $\kappa_n = k^n(\kappa)$. The reason this is so is that k and j agree on *all* sets in V: For any $x \in V$, we have $j(x) = k(x)$. The proof is

[38] The alert reader may notice that the stages of $V[G]$ appear to be defined in exactly the same way as the stages of V; this observation might lead one to wrongly conclude that V and $V[G]$ are the same universe. In reality, new sets, which do not belong to V, begin to appear in $V[G]$ at a stage that comes after those indexed by the whole numbers $0, 1, 2, \ldots$. The number that comes after all the whole numbers is denoted ω. Our forcing adds a new subset A of $\mathbb{N}$ (namely, the set $A = \{n \in \mathbb{N} \mid g(n) = 1\}$). One shows that $A \notin V$ but $A \subseteq V[G]_\omega$, and so $A \in V[G]_{\omega+1} = \mathcal{P}(V[G]_\omega)$ is a new set that is not found in V. We also point out that the indices of these stages are the same as those for the stages in the build-up of V itself; they extend well beyond the whole numbers, through the expanse of infinite ordinals.

given in this sequence of equalities. Given $x \in V$ and recalling that $x = (\check{x})_G$, we have:

$$\begin{aligned} k(x) &= k(\check{x}_G) \\ &= j(\check{x})_G \\ &= j(\{(\check{y}, 1) \mid y \in x\})_G \\ &= \{(\check{y}, 1) \mid y \in j(x)\}_G \\ &= (\check{j(x)})_G \\ &= j(x). \end{aligned}$$

Therefore, since $\kappa \in V$, we have $j(\kappa) = k(\kappa)$; since $j(\kappa) \in V$, we also have $j^2(\kappa) = j(j(\kappa)) = j(k(\kappa)) = k(k(\kappa)) = k^2(\kappa)$, and so on. One says that j and k agree on the critical sequence of j.

Using (B), we can now specify the sets $Z_0, Z_1, \ldots$ that were mentioned in the Proposition above: For each n, we will let $Z_n = V[G]_{\kappa_n}$. We are now in a position to explain why these particular sets allow us to obtain the result in the Proposition. Because WA holds in V, it is possible to show [2] that the critical sequence $\kappa_0, \kappa_1, \kappa_2, \ldots$ is *unbounded* in the universe; from this fact it follows that, for each set X in V, there is n such that $X \in V_{\kappa_n}$. Remarkably, the critical sequence continues to be unbounded after forcing:[39] For each $X \in V[G]$, there is n such that $X \in V[G]_{\kappa_n}$.

Now we can prove the Proposition quite easily. Suppose (ii) of the Proposition holds true: For each n, we assume $k \upharpoonright Z_n$ belongs to $V[G]$; that is, $k \upharpoonright V[G]_{\kappa_n}$ belongs to $V[G]$. To prove (i), let X be any set in $V[G]$. Let n be such that $X \in V[G]_{\kappa_n}$. Since $k \upharpoonright V[G]_{\kappa_n}$ is an ordinary function f lying in $V[G]$, restricting this function further to X is also an ordinary function in $V[G]$: $k \upharpoonright X = f \upharpoonright X$.

Having established the Proposition, in order to complete the proof of indestructibility of WA, we must now show that, for each n, $k \upharpoonright V[G]_{\kappa_n}$ belongs to $V[G]$. We already have a very clear idea about how $k \upharpoonright V[G]_{\kappa_n}$ is defined:

(5.9) $$\text{For each } \tau_G \in V[G]_{\kappa_n},\ k(\tau_G) = j(\sigma)_G.$$

This is just a restatement of how k was originally defined. The reason there is anything at all to be proved here is that k, acting on elements of $V[G]_{\kappa_n}$, might possibly have an *unbounded range*; it is conceivable, for example, that for each natural number $r > n$, there is $y \in V[G]_{\kappa_n}$ such that $k(y)$ has rank at least κ_r. In that case, the range of $k \upharpoonright V[G]_{\kappa_n}$, and therefore the map $k \upharpoonright V[G]_{\kappa_n}$ itself, would be a *proper class*[40] and not a set in $V[G]$.

[39] Here is the logic: Suppose τ_G is a set in $V[G]$. Then τ is a P-name and therefore belongs to V. Since WA holds in V, there is n such that $\tau \in V_{\kappa_n}$ and so $\tau \in V[G]_{\kappa_n}$. We need a definition: The *rank* of a set $Y \in V[G]$, denoted rank(Y), is the least α for which $Y \subseteq V[G]_n$. A fairly easy fact to prove about P-names is that the rank of a P-name has rank at least as big as the set that is named. In particular, rank$(\tau_G) \leq$ rank(τ). It follows that $\tau_G \in V[G]_{\kappa_n}$.

[40] In any universe V of sets, the sets are the objects that lie in one of the stages in the build-up of V: For any X, there is some α so that $X \in V_\alpha$. One can also talk about proper classes relative to V. A proper class is too big to be a set. An example is V itself (V cannot belong to any of its own stages V_α—it necessarily contains elements that lie outside of any such V_α). Another example is the collection $\{\{x\} \mid x \in V\}$; this is also a proper class. One way to prove that a collection C is a proper class is to show that for every α, there is an element $y \in C$ such that rank$(y) \geq \alpha$.

The fact is, though, the map $k \upharpoonright V[G]_{\kappa_n}$ does *not* have unbounded range. The reason for this follows from a technical fact concerning P-names. Using the fact that each κ_n must be a large cardinal, one can show [10] that every set y in $V[G]_{\kappa_n}$ has a P-name τ that belongs to V_{κ_n}; that is, $\tau \in V_{\kappa_n}$ and $\tau_G = y \in V[G]_{\kappa_n}$.

Now suppose $y \in V[G]_{\kappa_n}$. Let τ be a name for y that lies in V_{κ_n}. Then $k(y) = k(\tau_G) = j(\tau)_G$. By elementarity, and because $P \in V_\kappa \subseteq V_{\kappa_n}$, $j(\tau)$ is a P-name that belongs to $j(V_{\kappa_n}) = V_{j(\kappa_n)} = V_{\kappa_{n+1}}$. Since the rank of a name is always at least as big as the set that it names, it follows that $j(\tau)_G \in V[G]_{\kappa_{n+1}}$, and so $k(y) = k(\tau_G) \in V[G]_{\kappa_{n+1}}$. We have therefore shown that the range of $k \upharpoonright V[G]_{\kappa_n}$ is a subcollection of $V[G]_{\kappa_{n+1}}$; it follows that the map $k \upharpoonright V[G]_{\kappa_n}$ itself, viewed[41] as a collection of ordered pairs, is a subcollection of $V[G]_{\kappa_{n+1}}$. We also have seen that we have a simple formula—equation (5.9)—that defines the behavior of $k \upharpoonright V[G]_{\kappa_n}$ in terms of sets in $V[G]$. A direct consequence of the Axiom of Separation (an axiom of ZFC) is the fact that any definable subcollection of a set is itself a set; therefore, we have established that $k \upharpoonright V[G]_{\kappa_n}$, being a definable subcollection of $V[G]_{\kappa_{n+1}}$, is a set in $V[G]$.

This completes our outline of the proof that WA is preserved in forcing extensions. Let us summarize what has been done. Under the assumption that WA holds in the universe V, our objective is to show that WA holds in any forcing extension $V[G]$ obtained by forcing with the partial order P for adding a real to the universe. The proof requires us to do two things:

(1) Using the fact that there is an elementary embedding $j : V \to V$, show that there is an elementary embedding $k : V[G] \to V[G]$.

(2) Using the fact that the embedding $j : V \to V$ has the property that $j \upharpoonright X$ is a set in V for any set $X \in V$, show that the embedding k, obtained in (1), has the same property: For any set $X \in V[G]$, $k \upharpoonright X$ is also a set in $V[G]$.

For (1), we defined k by $k(\sigma_G) = j(\sigma)_G$. This definition is possible because j maps P-names to other P-names, so, in particular, $j(\sigma)$ is a P-name that can be evaluated using G. We showed that k is well-defined—if σ_G and τ_G denote the same set in $V[G]$, then $k(\sigma_G) = k(\tau_G)$. We then outlined the logic that shows k preserves all formulas of the forcing language. We proved this fact explicitly for simple formulas $\sigma_G = \tau_G$ and $\sigma_G \in \tau_G$ and outlined the inductive argument that establishes it for any more complex formula, built up using the connectives $\wedge, \vee, \neg$ and $\to$ and the quantifiers $\exists$ and $\forall$. Having established this more general result, we concluded that k is an elementary embedding.

For (2), we first noted that it is enough to show that, for each $n \in \mathbb{N}$, the map $k \upharpoonright V[G]_{\kappa_n}$ is a set in $V[G]$—having shown this, it is straightforward to show $k \upharpoonright X$ is a set for any set $X \in V[G]$. We then cited a technical fact that for each n, every set y belonging to $V[G]_{\kappa_n}$ has a name τ in V_{κ_n}. It follows that $j(\tau)$ belongs to $V_{\kappa_{n+1}}$ and that, therefore, $k(y) = k(\tau_G) = (j(\tau))_G$ belongs to $V[G]_{\kappa_{n+1}}$. The reasoning

This technique is being used here to explain why a function on $V[G]_{\kappa_n}$ having unbounded range cannot be a set in $V[G]$.

[41] Any function f may be viewed as a collection of ordered pairs:

$$f = \{(x, y) \mid y = f(x) \text{ and } x \in \operatorname{dom} f\}.$$

shows that $k \upharpoonright V[G]_{\kappa_n}$ is a definable subcollection of $V[G]_{\kappa_{n+1}}$ and is therefore a set in $V[G]$.

We have completed our outline of the proof that WA cannot be destroyed by forcing. In actual fact, there are some parts of the proof that we have not attempted to describe. For instance, our starting partial order may be much bigger than the P we used for our discussion here; it might be too big to fit inside V_κ, and so the reasoning given here would not be applicable. This difficulty is handled in [10]. Also, the version of the Wholeness Axiom we have discussed in this article is slightly weaker than the full-strength version that one finds in the original paper [1]. The proof of indestructibility for this stronger version of WA requires a considerably deeper analysis, which is carried out in [10].

6. Conclusion

We have seen how the mathematical need to give an account of large cardinals led to a closer examination of what might be missing in the current list of ZFC axioms. A key observation is that the ZFC axioms do not discuss the characteristics of the wholeness of the universe. The Wholeness Axiom is proposed as a way to fill this gap; it describes, in a mathematical way, the key characteristics of the wholeness of the mathematical universe. It accomplishes this aim by incorporating essential qualities and dynamics of wholeness as described in Maharishi Vedic Science. Moreover, it has been shown that, if ZFC is supplemented with the Wholeness Axiom, all the most widely studied large cardinals become easily derivable as properties of the first cardinal moved by the Wholeness Axiom embedding j [6, Theorem 7.1].

In this article, we have summarized the results of [10] in which the following important question is addressed: Is the Wholeness Axiom a *good* axiom? Does it survive the tranformational dynamics of the forcing technology? We have outlined in this article the proof from [10] that this is indeed the case: If the Wholeness Axiom holds true in the universe V, it must continue to hold true in every forcing extension $V[G]$ of V.

This result points to the fact that the Wholeness Axiom is a good axiom from two very different perspectives. First, as a mathematical result, the work here shows that the Wholeness Axiom passes a test that any new foundational axiom must pass: It survives the transformational dynamics involved in applying the technology of forcing. But the Wholeness Axiom proves itself to be a good axiom in a second sense as well. This axiom was originally crafted not only as a tool to solve the purely mathematical problem of accounting for large cardinals, but also with the intention to incorporate key aspects of the nature and dynamics of wholeness, as described in Maharishi Vedic Science, so that the foundation of mathematics could be structured on the basis of profound wisdom concerning the nature of wholeness. This article shows that the Wholeness Axiom has also been successful in this second respect. The work here shows that the Wholeness Axiom embodies enough of the principles of wholeness from Maharishi Vedic Science to guarantee its *indestructibility*. It is not at all obvious from the technical definition of the axiom that it should have this property; the fact that it does provides evidence that the effort to incorporate the ancient wisdom concerning wholeness into the axiom has been largely successful.

Using the language of Maharishi Vedic Science, this result that WA is indestructible by forcing tells us that when WA holds in the universe V, we find that V exhibits the characteristic[42] of *invincibility*; it maintains its knowledge of its own wholeness, in the form of the axiom WA, even in the face of the transformational dynamics imposed by forcing constructions. We spend a moment here to see how the mechanics of maintaining invincibility, as described in Maharishi Vedic Science, find expression in the present context of set theory.

The fundamental principle for establishing invincibility in an individual or nation is to create a condition of strong coherence. In the individual, strong coherence means that all parts of life work in step so that progress in life moves forward in a frictionless way; in short, life is lived in accord with Natural Law. Coherence in the individual is achieved by regular contact with pure consciousness, the source of all the diversity of the individual, through the process of transcending. At the level of the nation, sufficient coherence can be achieved when a large enough percentage of individuals in the nation have achieved a sufficient level of coherence within themselves.[43]

Many analogies from the sciences have been used to illustrate the mechanics of establishing invincibility. The most notable of these is the *Meissner Effect*[44] in physics. In an ordinary conductor, like lead, at ordinary temperatures, the electrons within the conductor are incoherently oriented. In the presence of an external magnetic field, the behavior of electrons within the conductor becomes even more chaotic; the magnetic field is able to penetrate the conductor to create a kind of disturbance. On the other hand, if one starts instead with a *superconductor*, whose internal electrons are in a highly coherent state, the presence of an external magnetic field has a very different effect. In that case, the electrons nearly instantaneously produce a magnetic field of their own that exactly cancels out the penetrating magnetic field everywhere inside the superconductor. The coherent state of the electrons cannot be disturbed by the external magnetic field. This internal coherence of electrons makes it impossible for the external disturbing influence of a magnetic field to undermine the integrity of the coherent state of the superconductor. This ability of a superconductor to repel a disturbing magnetic field is called the Meissner Effect.

We consider the extent to which the mechanics of invincibility find expression in the context of set theory, in the presence of the Wholeness Axiom. Assuming WA holds in V, the "coherence" we find in V is embodied in the elementary embedding j: The embedding makes lively a fundamental self-referral dynamic—from V to itself—that is completely absent when WA is absent. This self-referral dynamic results in a strong interconnectedness—a kind of "coherence"—among the parts of the

[42]See [19, pp. 150–301] for an extensive discussion of invincibility and its expression in the fields of physics and chemistry.

[43]There is considerable research by now that verifies the impact of coherence on society when even 1% of the population practices Transcendental Meditation (TM), and an even more dramatic impact of coherence when even just $\sqrt{(1\% \text{ of the population})}$ practice both TM and the TM-Sidhi program, including Yogic Flying. See [19, pp. 457–460] for a summary of this research, with references.

[44]See for example [19, p. 213].

universe.[45] When forcing is used on such a universe V (using a partial order that belongs[46] to V_κ), how exactly is this coherence maintained? Does it look like the Meissner Effect? Actually, coherence is maintained in a way that is quite different from the way it occurs in the Meissner Effect. In the Meissner Effect, a disturbing influence is *repelled*. But in the context of WA, the disturbing effect of expanding V by forcing is not repelled, but is rather *embraced*. The Wholeness Axiom, which asserts there is an elementary embedding $j : V \to V$, is resurrected in the forcing extension $V[G]$ by extending j to embrace all the new sets that the presence of G has brought into existence. The coherence that was found in V is expanded to encompass all elements of $V[G]$. This expansion is achieved by appreciating these new sets as aspects of the "Self"—each new set is the realization of a name that is known within V; a new embedding $k : V[G] \to V[G]$ is obtained by relying on the self-interacting dynamics already present within V; $k(\tau_G)$ is defined to be the evaluation of $j(\tau)$ by G. The very influence that has created the disturbance, namely G, is used to define a new, more expanded level of coherence. One can say in the language of Maharishi Vedic Science that lifting j to k is a way of "stopping the birth of an enemy."[47]

We have seen in this article how introducing the Wholeness Axiom as a foundational axiom for set theory enriches the foundation of mathematics by giving expression to the nature of the wholeness of the mathematical universe. Taking this step has made it possible to provide a nearly complete solution to the problem of large cardinals. We have also seen that this new axiom embodies enough of the principles of wholeness as described in Maharishi Vedic Science to give expression to the indestructible nature of wholeness; to make lively within the universe of mathematics its inherent invincible nature.

[45] One example of this interconnectedness is the following remarkable fact: If WA holds in V, then almost every stage V_α of the universe has complete knowledge of V. More precisely, for almost all α and for any statement ϕ in the language of set theory, ϕ is true in the universe if and only if, from the perspective of the stage V_α, ϕ is seen to be true. This means that all "secrets of the universe" are known everywhere; full knowledge of the totality is present in each part.

[46] This article has focused on partial orders that belong to the stage V_κ, but indestructibility of WA is not limited to forcing based on such partial orders. In this footnote, we mention some points about how coherence is maintained when the partial order does not belong to V_κ. Suppose the forcing at hand requires the use of a large partial order Q. The first step in handling this difficulty is to notice that Q must belong to V_{κ_n} for some n. The strategy is then to replace j by a certain kind of iterate of j. There is an operation called *application*, denoted by a dot $\cdot$, that allows one to apply j to itself, to produce new embeddings $j \cdot j$, $j \cdot (j \cdot j)$, ... (See [7, p. 181ff.] for a discussion of this operation.) One writes $j^{[2]} = j \cdot j$, $j^{[3]} = j \cdot (j \cdot j)$, and, in general, $j^{[n+1]} = j \cdot j^{[n]}$. Each map $j^{[n]}$ is another WA-embedding from V to itself. Intuitively speaking, each $j^{[n]}$ can be seen as a "higher octave" of the original embedding j. Moreover, it can be shown that the least cardinal moved by $j^{[n]}$ is κ_n. This means that $j^{[n]}(Q) = Q$, and so all the techniques of proof described earlier in this article can be used to show that $j^{[n]}$ can be lifted to a WA-embedding $k : V[G] \to V[G]$, as before. Therefore, even in the presence of this kind of bigger forcing, WA is preserved. And here also we see that the way that coherence is maintained under the influence of forcing is by *embracing* the influence. In this case, a higher octave of j is lifted to an embedding $k : V[G] \to V[G]$ so that the sets introduced by using G are related to each other in a coherent way. This is accomplished using the same technique as before: For each τ_G in $V[G]$, $k(\tau_G) = (j^{[n]}(\sigma))_G$.

[47] See for example [19, p. 69].

In human life, the winds of change can bring about events and circumstances that are unexpected and surprising. It is the indestructibility of wholeness that makes it possible to be at home with any kind of change that may arise; that makes it possible to live an invincible life. It seems to the author very fortunate that we can now find lively within the domain of modern mathematics this indestructible character of wholeness.

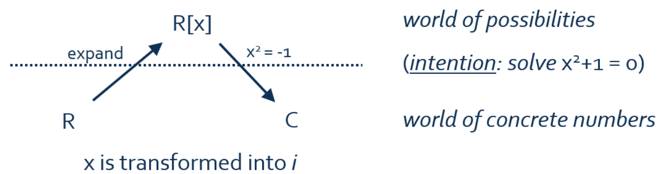
7. Appendix I: Parallels Between Forcing Constructions and the Construction of the Complex Number Field

In this Appendix, we review the construction of the field $\mathbb{C}$ of complex numbers from the real line $\mathbb{R}$ and highlight parallels between this construction and the construction of a forcing extension of the universe.

Historically, the fact that there is no real number x for which $x^2+1=0$ turned out to be quite inconvenient. For instance, in some cases, intermediate computations in applying the formula developed by Tartaglia and G. Cardano for obtaining roots of a third degree polynomial require computing the square root of negative numbers; this formula was developed long before the rigorous development of complex numbers.

In this appendix, we discuss a version of the mechanism that was used to add a new (imaginary) number i to the real number line $\mathbb{R}$ to obtain a new number field $\mathbb{C}$, known as the field of *complex numbers*. The number i is defined so that it satisfies $x^2+1=0$; in other words, $i^2=-1$.

To construct $\mathbb{C}$, we begin with the set $\mathbb{R}$ of real numbers; we wish to expand $\mathbb{R}$ to some new field $\mathbb{R}[i]$—the smallest field that includes $\mathbb{R}$ as a subset but that also contains the new "ideal" element i. As we carry out the construction of the complex numbers, we identify parallels to the construction used in the forcing methodology.



The first step in the case of forcing is to expand V to a world V^P of potential sets; V^P is no longer a universe of sets, but serves as a universe of *names* of sets in a new alternative universe. In analogous fashion, the first step in building a number field that includes both $\mathbb{R}$ and i is to expand $\mathbb{R}$ to a much bigger structure $\mathbb{R}[x]$, the ring of *polynomials over* $\mathbb{R}$ (a polynomial over $\mathbb{R}$ is an expression of the form $a_nx^n + a_{n-1}x^{n-1} + \cdots + a_2x^2 + a_1x + a_0$). The symbol x represents an "indeterminate"; it can be thought of as an analogue to the *name* Γ of the generic filter G. The symbol x has the potential of becoming an actual element of a field, but $\mathbb{R}[x]$ itself is not a field, just as Γ has the potential of becoming an actual element of a new universe $V[G]$. The structure $\mathbb{R}[x]$ may be thought of as consisting of "potential" field elements or of "names" for elements of a new, as yet unconstructed field.

Returning to forcing, when V^P is collapsed using a generic G, there are actually two steps involved. The first step is to identify elements of V^P that will, in the final collapse, name the same set. This step produces a collection V^P/G of equivalence classes, where two elements of V^P are said to be equivalent if they are "forced" to be equal[48] by some element of P that belongs to G.

Analogously, the first step of collapse in creating the complex numbers also involves identifying elements of $\mathbb{R}[x]$ that will ultimately be equal as complex numbers. The equivalence relation in this case is obtained by thinking of the expression x^2+1 as representing 0 (this is intuitively natural since we wish to be able to solve the equation $x^2+1=0$ in the final collapse). One achieves this by declaring two polynomials to be equivalent if their difference is divisible by the polynomial x^2+1. The collection of all equivalence classes in this case is denoted $\mathbb{R}[x]/\langle x^2+1\rangle$. Note that this equivalence relation causes x^2+1 to be identified with the trivial polynomial 0 (since the difference $(x^2+1)-0$ is divisible by x^2+1).

The second and final step of collapse in forcing involves a transformation of the equivalence classes in V^P/G to actual sets; this is achieved by the Mostowski collapsing function that transforms (in a recursive way) every equivalence class $[\tau]$ to a set τ_G according to the rule: $\tau_G = \{\sigma_G \mid [\sigma] \mathrm{~E~} [\tau]\}$, where E is the membership relation for the model V^P/G. In this way, each "name" τ in V^P is ultimately mapped to a set τ_G in the final universe $V[G]$. In particular, the equivalence class $[\Gamma]$ is realized as the actual set $G = \Gamma_G$ in $V[G]$.

Analogously, the final step of collapse that leads to the final number field $\mathbb{C}$ of complex numbers is obtained in the following way: We map the equivalence class $[x]$ containing the indeterminate x to the imaginary number i and then, in general, map $[p(x)]$ to $p(i)$. In this way, each "name" (polynomial) $p(x)$ in $\mathbb{R}[x]$ is mapped to a complex number $p(i)$ in $\mathbb{C}$.

One can reasonably ask how it is possible to map $[x]$ to i when we do not yet know that i even exists. Before we invoke this mapping, we can think of $\mathbb{C}$ as consisting of all *formal expressions* $a+ib$, where $a, b \in \mathbb{R}$ and i satisfies the equation $x^2+1=0$. The operations of addition and multiplication may be defined on $\mathbb{C}$ in the obvious way (for instance, $(a+ib)+(c+id) = (a+c)+i(b+d)$). One can show that $\mathbb{C}$, with these operations and with zero element $0+i\cdot 0$ and unit element $1+0\cdot i$, forms a field. However, even with these steps, one can ask whether such a field "exists," since all elements of $\mathbb{C}$ are simply formal expressions based on the imaginary number i. The construction of $\mathbb{R}[x]$, however, produces a field of actual mathematical objects (polynomials) and the quotient field $\mathbb{R}[x]/\langle x^2+1\rangle$ is also a real mathematical object. (One can also ask in what sense a polynomial is "real" since the sense in which the indeterminate x exists is not clear. However, one should consider a polynomial $a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$ as convenient notation for the finite sequence $(a_0, a_1, a_2, \ldots, a_n)$; the indeterminate x is not needed. Thus, the familiar polynomial x^2+1 is in reality a convenient notation for the finite sequence $(1, 0, 1)$.) The mapping $[p(x)] \mapsto p(i)$ (which is an isomorphism) shows that $\mathbb{C}$ is indeed quite real, and the notation for elements of $\mathbb{C}$ that we have described here is a more convenient way of describing the elements of the field $\mathbb{R}[x]/\langle x^2+1\rangle$.

In an analogous way, we can ask whether the forcing extension $V[G]$ is "real" since it is not clear that G itself exists at all. As is discussed in Appendix II (p. 39), the generic filter G may be understood to be, like i, a convenient notation for a

[48]See the section "More on the Universe V^P of Names and the Forcing Relation," p. 21, for a discussion of the forcing relation.

real mathematical object that has a somewhat complicated definition. As described in that appendix, the "reality" of a forcing extension $V[G]$ lies in the universe V^P of names, and $V[G]$ is a convenient way of providing concrete expression to the elements of that universe.

In forcing, the "intention" that leads to a forcing extension $V[G]$ and that satisfies our intended property is captured by the partial order P. In producing $\mathbb{C}$, the "intention" is captured in the definition of the equivalence relation, which forces the polynomial $x^2 + 1$ to be viewed as 0.

8. Appendix II: Development of Forcing Without Collapsing the Universe of Names

In this Appendix, we describe an alternative approach to obtaining a generic filter G in order to collapse V^P to $V[G]$. In this alternative approach, we work entirely in the world V^P of names and never need to take the step of collapsing to obtain a new universe $V[G]$.

To take this step, it is first necessary to embed the partial order P into a somewhat richer kind of partial order B called a *complete Boolean algebra*. A complete Boolean algebra is also a partial order with largest element 1; however, a complete Boolean algebra also has a smallest element 0 and operations $\vee$ ("join"), $\wedge$ ("meet"), and * ("complement"), which are naturally related to the connectives used in logic: $\vee$ ("or"), $\wedge$ ("and"), and $\neg$ ("not"). (For the interested reader, we explain more about these operations. Suppose $b, c \in B$, where B is a complete Boolean algebra. Then $b \vee c$ is the *least upper bound* of b and c; $b \wedge c$ is the *greatest lower bound* of b and c; and b^* is the *complement* of b; that is, b^* is the unique element d of B such that $b \wedge d = 0$ and $b \vee c = 1$.) If P is embedded in a complete Boolean algebra B in this way, we write $B = \mathrm{ro}(P)$ (B is technically known as the *regular open algebra of* P; see [13]). It can be shown that forcing with P produces the same forcing extension as forcing with B treated as an ordinary partial order (with the bottom element 0 removed from B).

To develop the forcing machinery based on complete Boolean algebras, one begins by building up B-names in essentially the same way as P-names were defined earlier in this article. The collection of all B-names is denoted V^B. One defines the canonical name $\check{x}$ for each x in V in essentially the same way as described earlier, and so we view V as a subclass of V^B. Because B is a complete Boolean algebra, V^B can be viewed as a kind of model of set theory, a kind of universe of sets (by contrast, we cannot view V^P as a model of set theory because an ordinary partial order P lacks the additional structure that is provided by a complete Boolean algebra). Just as any model of set theory is able to determine the truth or falsity of any sentence in the language of sets (for instance, the sentence $\emptyset \in \emptyset$ is false in every model of set theory), so likewise can a truth value be assigned to every sentence of the forcing language, working in V^B. However, in the context of V^B, the "truth values" are no longer simply *true* and *false*; in this context, each element of B is considered to be a truth value. Each $b \in B$ represents a kind of *degree of truth*; if b is closer to the largest element 1 of B, then b is "almost true," whereas if b is close to the smallest element 0, then b is "almost false."

Statements in the forcing language are assigned truth values (that is, elements of B) by an evaluation map $[\![\,\cdot\,]\!]_B$. For each statement ϕ in the forcing language, the expression $[\![\phi]\!]_B$ denotes an element b of B, signifying the "truth" of ϕ. If b is close to the top element 1 of B, then we think of ϕ as being "almost certainly" true, while if b is close to the bottom element 0 of B, then ϕ is "almost certainly" false. The forcing relation $\Vdash$ can now be defined in V as follows: For any sentence χ of the forcing language and for any $b \in B$, the meaning of $b \Vdash \chi$ is $b \leq [\![\chi]\!]$.

As an example, if ϕ is the following formal sentence in the forcing language for B

$$\check{\emptyset} \in \check{\emptyset},$$

then ϕ will have value 0 (in other words, $[\![\phi]\!]_B = 0$) no matter which complete Boolean algebra B we begin with. Equivalently, one has $[\![\neg\phi]\!]_B = 1$. Using the forcing relation, one could write this second expression in the equivalent form: $1 \Vdash \neg\phi$. (Note that, in order to evaluate even this simple sentence, it is necessary to assign meaning to the membership symbol $\in$; one also needs to assign meaning to the usual equality symbol $=$. These definitions are rather complicated, but can be found in texts like [13].)

To see how the Boolean operations $\wedge, \vee$, and $*$ come into play, one can show, for example, that for any two sentences ϕ and ψ in the forcing language,

$$[\![\phi \wedge \psi]\!]_B = [\![\phi]\!]_B \wedge [\![\psi]\!]_B.$$

We see here how the logical "and" ($\wedge$) connective directly translates into the Boolean operation by the same name ($\wedge$).

One can show that every axiom of ZFC has Boolean value 1 in V^B (for each axiom ϕ of ZFC, we have $[\![\phi]\!]_B = 1$). In this sense, V^B can be viewed as a kind of model of set theory, though in this case there is a multiplicity of truth values. For forcing arguments, if, for some sentence ψ of the forcing language and some complete Boolean algebra B, we discover that $[\![\psi]\!]_B > 0$, then we conclude that ψ is *consistent* (if it were not consistent, then there would be a proof from ZFC of $\neg\psi$ and we could show that, in fact, $[\![\neg\psi]\!]_B = 1$, since all ZFC axioms have Boolean value 1). And, if we can find another complete Boolean algebra C for which $[\![\neg\psi]\!]_C > 0$, then we likewise conclude that $\neg\psi$ is consistent (that is, it is consistent for ψ to be false). In this way, one shows, using Boolean-valued models, that ψ is undecidable.

The steps described in the previous paragraph are precisely how one shows that the Continuum Hypothesis (CH) is undecidable using Boolean-valued models: In one model V^B, one shows $[\![\mathrm{CH}]\!]_B = 1$, and in another model V^C, we have $[\![\neg\mathrm{CH}]\!]_C = 1$. Here, $B = \mathrm{ro}(R)$, where R is the partial order for adding a new subset of ω_1, and $C = \mathrm{ro}(Q)$, where Q consists of finite partial functions from $X \times \mathbb{N} \to \{0, 1\}$, where $|X| = \omega_2$, as described earlier in this article; recall that forcing with Q (and hence also with C) adds ω_2 new reals to the universe.

Using the method of Boolean-valued models, no concrete (two-valued) "new" universe is ever created; there are no forcing extensions $V[G]$; there is no generic filter G. The undecidability of propositions is established by studying only the world of names that is induced by the partial order (embedded in a complete Boolean algebra). This approach to forcing suggests a philosophy that these universes of names are in some sense more "real" than the concrete models produced by collapsing with a generic filter. This perspective is reminiscent of a point of view commonly held by quantum field theorists concerning the true nature of concrete particles in nature, expressed in a recent article by quantum field theorist Art Hobson [12]:

> Quantum foundations are still unsettled, with mixed effects on science and society. By now it should be possible to obtain consensus on at least one issue: Are the fundamental constituents fields or particles? As this article shows, experiment and theory imply that unbounded fields, not bounded particles, are fundamental.... Particles are epiphenomena arising from fields (p. 211).

Though, as reasonable human beings, we may tend to believe we are interacting with concrete objects in a physical world, the underlying reality is in fact the dynamics of unbounded fields. And this point of view matches the ancient wisdom on the ultimate nature of the physical world. As Maharishi explains [19]:

> Here Unity (Samhita) appears to be diversity (Rishi, Devata, Chhandas). This is the absolute, eternal principle of *Vivarta*, where something appears as something else. The very structure of pure knowledge (Samhita) has the principle of *Vivarta* (Rishi, Devata, Chhandas) within it. (p. 589)

and also:

> The principle of *Vivarta* makes the unmanifest quality of self-referral consciousness appear as the Veda and Vedic Literature, and makes the Veda and Vedic Literature appear as Vishwa. (pp. 377, 589)

The fact that undecidability of propositions can be discovered by exploring just the "world of possibilities" represented by the Boolean-valued universe V^B brings the activity of mathematics closer to Maharishi's Vedic Mathematics, as the mathematics of all possibilities [19]:

> The obvious conclusion is that as Vedic Mathematics is the mathematics of the Self, as Vedic Mathematics is the mathematics of one's own consciousness and as consciousness is a field of all possibilities, Vedic Mathematics is the mathematics of all possibilities—all possible computations, all possible derivations, all possible calculations wait at the door of Vedic Mathematics. (p. 385)

This Boolean-valued approach is also suggestive of an alternative approach to gaining knowledge. The approach in which new models $V[G]$ are created involves creating something external to V in order to know the truth (or undecidability) of a proposition, while the Boolean-valued model approach does not require one to step outside of V at all. This alternative approach is reminiscent of the deeper wisdom that everything knowable can be known entirely from within one's own nature. Laozi, in the *Tao Te Ching*, expresses this eternal truth lyrically in the following way [11]:

> Without going outside, you may know the whole world.
> Without looking through the window, you may see the ways of heaven.
> The farther you go, the less you know.
> Thus the sage knows without traveling.

He sees without looking.
He works without doing. (Verse 47)

References

[1] P. Corazza. *The wholeness axiom and Laver sequences.* Annals of Pure and Applied Logic. (2000) 105 pp. 157–260.

[2] P. Corazza. *The spectrum of elementary embeddings $j : V \to V$.* Annals of Pure and Applied Logic. (2006) 139 pp. 327–399.

[3] P. Corazza. *The wholeness axiom.* In *Consciousness-Based Education: A Foundation for Teaching and Learning in the Academic Disciplines, Volume 5: Consciousness-Based Education and Mathematics, Part I.* P. Corazza, A. Dow, C. Pearson, D. Llewellyn (eds.). Maharishi University of Management Press, Fairfield, IA, 2011, pp. 225–264.

[4] P. Corazza. *Vedic wholeness and the mathematical universe: Maharishi Vedic Science as a tool for research in the foundations of mathematics.* In *Consciousness-Based Education: A Foundation for Teaching and Learning in the Academic Disciplines, Volume 5: Consciousness-Based Education and Mathematics, Part II.* P. Corazza, A. Dow, C. Pearson, D. Llewellyn (eds.). Maharishi University of Management Press, Fairfield, IA, 2011, pp. 109–229.

[5] P. Corazza. *The axiom of infinity and transformations $j : V \to V$.* The Bulletin of Symbolic Logic. (2010) 16 pp. 37–84.

[6] P. Corazza. *Addressing the problem of large cardinals with Vedic wisdom.* International Journal of Mathematics and Consciousness. (2015) 1(1) pp. 67–89.

[7] P. Corazza. *Magical origin of the natural numbers.* International Journal of Mathematics and Consciousness. (2016) 2(1) pp. 1–207.

[8] P. Corazza. *Tao of the Infinite. To appear (2000 pages).*

[9] P. Corazza. *The Axiom of Infinity, QFT, large cardinals. To appear.* 130 pages.

[10] P. Corazza. *Indestructibility of wholeness.* Fundamenta Mathematicae. (2021) 252 pp. 147–170.

[11] G. Feng and J. English. *Tao Te Ching, 25th Anniversary Edition.* Vintage, 1997. Available at: `http://www.duhtao.com/translations.html`.

[12] A. Hobson. *There are no particles, there are only fields.* American Journal of Physics. (2013) 81(3) pp. 211-223.

[13] T. Jech. *Set Theory: The Third Millenium Edition, Revised and Expanded.* Springer-Verlag, 2003.

[14] Maharishi Mahesh Yogi. *Maharishi Mahesh Yogi on the Bhagavad-Gita, a New Translation and Commentary with Sanskrit Text.* Penguin Books, 1967.

[15] Maharishi Mahesh Yogi. *Maharishi Vedic University, Inauguration.* Age of Enlightenment Press, 1985.

[16] Maharishi Mahesh Yogi. *The science of being and art of living.* Plume, New York, 2001.

[17] Maharishi Mahesh Yogi. *Life supported by natural law: Lectures by His Holiness Maharishi Mahesh Yogi.* Maharishi International University Press, Fairfield, IA, 1988.

[18] Maharishi Mahesh Yogi. *Maharishi Vedic University: Introduction.* Maharishi Vedic University Press, Vlodrop, Holland, 1994.

[19] Maharishi Mahesh Yogi. *Maharishi's absolute theory of defence.* Maharishi Vedic University Press, Vlodrop, Holland, 1996.

Mathematics Department, Maharishi International University, Fairfield, IA

NATURAL LANGUAGE PROCESSING WITH SEMANTICS AND LOGIC: A NECESSITY STRONGLY SUPPORTED BY MAHARISHI VEDIC SCIENCE

EMDAD KHAN

Abstract. Natural Language Processing (NLP) is a rapidly growing field, with great progress having been made over the past sixty years. However, due to the high level of complexity of NLP in general, it remains an open problem. NLP complexity is mainly related to semantics, which involves abstraction, representation, real meaning, and computational complexity. In this paper, we argue that while existing approaches are effective in solving some specific problems of NLP, they do not address key NLP problems in a practical and natural way.

As humans, we can express our thoughts, create an abstract of a document, make inferences from an article, debate various topics, and answer questions using our knowledge and analysis. All these involve semantics, logic, intelligence, and our thinking capabilities. Computers cannot accomplish these things in many circumstances. It can be extremely difficult or even impossible for computers to work with knowledge, semantics, logic, intelligence, and thinking in the same way that the human mind can. Today's computers are capable for number crunching but not so well suited for the human capabilities listed above.

We show that a semantic engine which computes the meanings of words, sentences, and paragraphs—especially using a brain-like approach and brain-inspired algorithms—is critical to solving the key NLP problems of semantics: abstraction, representation, real meaning, and computational complexity.

We also show that logic and machine learning are integral parts in conjunction with semantics to help accomplish natural human interaction with any computing system. Emphasis on lifelong machine learning allows integrative learning, an essential component that permits NLP systems to learn from previous experiences by integrating past learning with the current matter at hand, similar to how we learn as humans.

Finally, we present a solution using a semantic engine, machine learning, logic, and lifelong machine learning. We call it Intelligent Natural Language Computing System or LMLS_NL_SEM_LOGIC which is a natural language-driven lifelong machine learning system that solves all key NLP problems listed above, although not yet to the level that humans can.

Intelligence and thoughts are directly related to consciousness. Natural language is key for the creation of expression of thoughts and intelligence. Thus, natural language is strongly related to pure consciousness. Logic, semantics, and learning work together very closely to create expressions from thoughts and intelligence (or to understand and convert expressions to thoughts/intelligence) using natural language. Thus, our semantics, logic, machine learning and lifelong machine learning-based NLP approach is well aligned with Maharishi Vedic Science as we shall discuss, and hence has the potential to work in a natural way.

Received by the editors August 12, 2021.

1. Introduction

Natural language is the normal way of communication for humans. We use natural language to communicate via speaking or writing. On the other hand, to communicate with computing machines we mainly use keyboards, keypads, touch pads, mice, and the like as well as graphical user interfaces (GUI). Microphones and speakers are used in limited ways since computing machines do not yet understand natural language at a level that allows us to communicate with such machines. Research is making great progress in this direction. For example, today we can speak with Siri from Apple, Alexa from Amazon, or Google Voice from Google. These systems understand some basic words and sentences. However, conversations cannot be continued for more than a few minutes. The main reason for this is the limited understanding of the semantics of human natural language by today's computing devices. Other key reasons are the limited capabilities of learning, logic, and cognitive computing which means deriving new facts from existing facts, applying knowledge in a more intelligent way, and helping the inference process.

It would be ideal if we could interact with a computer naturally as follows:

(1) Say "Show me all the pictures from last Saturday's birthday party," and get all the requested pictures from Facebook.
(2) Say "I would like to buy the book *Artificial Intelligence* by Stuart Russel, 3rd edition; use my credit card on file and ship it to my home address," and receive the requested book on time.
(3) Say "How do I sell my produce?" and get useful and specific answers that help a farmer's sales effort.
(4) Ask a specific question and get the correct answer.
(5) Get a summary of an article.
(6) Get a useful prediction from business intelligence (BI) or analytics software.

In fact, we see clear indications that the future Internet will be something that can provide very specific, precise, and direct information (like the examples listed above) in a very easy way so that anyone, even an illiterate person, can naturally talk to, listen to, or view the Internet easily using any computing device. We call this the Intelligent Internet (IINT) [59].

The need to more effectively communicate with computing devices has become very important with the rapid growth of the Internet, which has become an important and essential part of everybody's life. It is a key driver for almost everything, including basic necessities such as food, water, shelter, and healthcare. The Internet and associated new devices such as smart phones, smart watches, and the Internet of Things have fueled the rapid growth of data, both structured (for example, data in a database) and unstructured (for example, texts, audio, and video).

In fact, data in our world has been exploding. Analyzing, processing, searching, storing, and understanding large data sets, so-called *Big Data*, has become a critical issue that provides both challenges and opportunities. The increasing volume and details of information captured by enterprises, the rise of multimedia, social media, and the Internet of Things will fuel exponential growth in data for the foreseeable future [12]. Big Data includes both structured data and unstructured data. Unstructured data dominates the Internet (80% unstructured and 20% structured). Since NLP is mainly unstructured data, the need to address NLP problems is growing rapidly.

Broadly speaking, we have two key NLP problems to solve using semantics, logic, machine learning and a cognitive computing driven approach:

- Core Computation, which involves understanding, analyzing, processing, searching, inferring, summarizing, recommending, generating natural language, using data including Big Data.
- User Interface, which enables humans to interact and converse with computing devices in a natural way, using voice or text. This is the main component of conversational AI.

Solving these two broad NLP related problems is difficult from multiple angles. As already mentioned, today's computers are adequate for number-crunching tasks but are not well suited for the typical human queries listed above (refer to Section 2 below). At a high level, the major issues are how to

(1) Convert key human features—expression, semantics, abstraction, summarization, queries, and inferences into a form that numerical computers will understand and process successfully.
(2) Convert key ingredients for mathematics, statistics and other number driven fields into key human features mentioned above.

As mentioned, numbers are not well suited to represent a sentence which has several words. The meaning, one major part of semantics, of a sentence is not just the sequence of words represented in numbers. Rather we need to relate and integrate the meaning of each word to compute the meaning of a sentence in a logical way. All these are very hard to do using the computers that we have today. To summarize, when drawing an inference or answering a question, in addition to processing meanings, we need to use knowledge and intelligence.

Besides the difficulties in computing with today's computers, we also do not have algorithms for most of the key natural language features for the tasks listed above.

Thus, we need to figure out:

- How to use today's computers by determining good representation of semantics, knowledge, and logic.
- How to do the key tasks listed above in a very effective way.
- How to think about different types of computing machines.

Most number-based computers will probably not be able to solve all NLP-related problems. Thus, we would need different types of computers. Many researchers have been working on such computers [58].

1.1. **OUTLINE.** We have discussed all key issues related to NLP problems, identified key areas for our focus (semantics, knowledge representation, semantic engine, logic, machine learning for unstructured data, and lifelong machine learning to solve complex NLP problems), and provided a solution to such problems. We have also determined that existing computing machines may not be able to solve all complex NLP problems due to the numerical way computing machines address NLP issues, which are dominated by semantics.

We have explained how natural language is related to consciousness. Logic, semantics, and learning work together very closely to create expressions from intelligent thoughts and understand and convert expressions to thoughts/intelligence using natural language. Therefore, our semantics, logic, machine learning, and lifelong machine learning based-NLP approach is well-aligned with Maharishi Vedic Science, having great potential to work in a natural way.

Our solution mainly addresses key NLP problems of core computation, all complex NLP related computations, and user interface to enable natural language based communication with any computing machine. Key components of our solution are:

(1) Defining and representing semantics, an open problem in computer science. We have used a new way of representing semantics (see Section 4).
(2) Semantic Engine: a core engine needed to deal with all complex NLP tasks. Develop a semantic engine called SEBLA (Semantic Engine using Brain-Like and Brain-Inspired Algorithms; see Section 4).
(3) Machine Learning for unstructured data. Not using number-based training, as used in regular ML, but rather using logic and semantics, as we usually learn from lectures or reading materials, rather than use a large data set for training and learning (see Section 5).
(4) Computing using semantics, logic, machine learning, and lifelong machine learning. This is called "Intelligent Natural Language Computing System," or LMLS_NL_SEM_LOGIC (A Natural Language Driven Lifelong Machine Learning System) (see Section 6).

These working together support all major aspects of natural language issues discussed, including cumulative and integrative learning. LMLS_NL_SEM_LOGIC addresses both core computation and user interface issues as discussed above.

Section 2 briefly describes all major issues that we need to address in dealing with NLP in a computing machine. Section 3 describes what has been achieved so far. Section 4 discusses Semantic Engine, needed to understand the meaning of a word, sentence, or paragraph. Section 5 discusses Machine Learning Paradigm for unstructured data (for example, text data). Section 6 describes lifelong machine learning and our proposed solution, LMLS_NL_SEM_LOGIC, that solves all key NLP problems mentioned above, but not yet to the level humans can. Section 7 discusses strong synergy with Maharishi Vedic Science and Section 8 presents conclusions and future directions. A literature review is provided in Appendix A.

2. Major Steps in Natural Language Processing

When we read a sentence, we understand the sentence and we understand the meaning of the whole sentence. We do so with ease, using the meaning of each word to derive or understand the meaning of the sentence. Now let's think about how we can tell a computer to do the same. Here comes the difficult part: things that we take for granted to do easily as humans are very difficult for a computer to do.

For example, we automatically understand the words in a sentence, the meaning (semantics) of each word, how they fit within the grammatical structure, where one sentence ends and another sentence begins, and so on. Unfortunately, we need to spell everything out in a proper way so that we can program a computer to do the same. First, in this process we would need to break the sentence into words, then determine the types of words (parts of speech), then determine whether a word is an entity, then parse the sentence—for example, subject, predicate, object, define meaning (semantics) of each word, and finally provide a mechanism to combine meanings of all the words in a manner that would yield the same meaning that we understand as a human when we read the sentence.

We also need to tell the computer about the root of a word. A root word is a word or word part that forms the basis of new words through the addition of prefixes and suffixes. Understanding the meanings of common roots can help a computer work out the meanings of new words as it encounters them. Many of the words we use in our daily language come from a root word. Finding a root word by taking the suffix or prefix (affixes) out is called stemming, and the root word is called the stem. Once you pull off any prefixes or suffixes, the root is usually what remains. For example,"egotist" has a root word of "ego", plus the suffix -ist. "Acting" has the root word "act", while "-ing" is merely the suffix. For example, search engines or bots use stemming to find all related words from the stem to get better search results (or better understanding for the bot).

However, sometimes just taking out the prefix or suffix may not provide the correct root word. In such cases, lemmatization, which takes into consideration the morphological analysis of the words to find the root, is used. The stem may not be an actual natural language word, whereas the lemma is an actual language word. For example, for the word "believes," stemming will remove "es" resulting "believ", rendering a stem which does not have a meaning. But a lemmatization will provide "believe," as it ensures that the root word is a correct word of English vocabulary. Here, the root is called the lemma.

This is a reasonable description to explain how difficult it is for a computer to process a sentence. We, as humans, do not need to do all this based on our knowledge of natural language. We process words, sentences, and paragraphs—naturally. We do not even identify parts of speech, determine names and entities, identify subject or predicate objects separately. In our brain computation model these happen naturally. But to make computers understand and process a sentence, we need to spell out everything as described above, including the meaning and context for each word. [Refer to Appendix B, which discusses the key NLP steps and processes (except semantics) in more detail. Semantics issues are briefly discussed below.]

Expressing the meaning of a word, one key part of semantics, needs further explanation as it is the most difficult part that we have to convey to a computer. We learn meanings of words from our childhood using various areas in our brain which involves seeing, hearing, talking, logic, thinking, and more. Since we are not really aware of how exactly these things happen in our brain and mind, we do not really know yet how to convey these processes to a computer—in other words, we cannot program a computer well for these complex processes.

Let's contrast these with computations using numbers. Number-based representation can easily do numerical computations, such as addition, subtraction, multiplication, and solving differential equations, quite well. But how can we represent the meaning of a word with a number? This is the crux in dealing with NLP. For numerical cases, we can define the meanings clearly so that the program can compute correctly, except when we need to do something with infinity, completeness, decidability, consistency, and soundness.

This semantics issue becomes a much larger issue when we try to program a computer to derive the meaning of a paragraph by using the meanings of its sentences, as well as using the meanings of paragraphs to derive a summary of a document. The same happens when we try to program a computer to draw an inference after reading a few documents, answer a complex question, or effectively continue a dialogue with a human for a long time. Humans also do cumulative learning (add new learning to what was learned before) and integrative learning (logically integrating various bits of information, knowledge, and intelligence) effortlessly. To program a computer to do so is much more difficult.

Thus, to effectively convey to a computer how to deal with natural language, we need to have very good representation of semantics and knowledge, core computing capabilities, logic, learning, cognition, and thinking.

Sections 3–6 discuss these key issues in more detail, along with a state of the art solution.

3. What Has Been Achieved So Far

The following key steps of NLP processing (as discussed in Section 2), can be done reasonably well today [Refer to Appendix B for details]:

(1) Breaking a text into sentences (sentence segmentation)
(2) Breaking a sentence, text, or string into words or tokens (Word Tokenization)
(3) Predicting parts of speech for each token or word
(4) Text lemmatization and stemming
(5) Identifying stop words
(6) Dependency parsing
(7) Named Entity Recognition (NER)

However, not all of these can be done at a highly satisfactory level. NER and parsing can be complex. Syntactic parsing often provides wrong parsing (wrong parse tree). Semantics-driven syntactic parsing can improve it to some extent but since semantics cannot be well handled yet (as mentioned in Sections 1 and 2), complex parsing needs improvement. As mentioned, semantics, logic, knowledge representation, representation of semantics, and cognitive computing are keys to deal with the most common NLP tasks, such as information extraction, information retrieval, universe of discourse, answering questions, summarization, and drawing inference. These are the key focus areas of NLP researchers today. The contributions made so far in these areas are dominated by predicate logic, frames, and semantic networks. Semantic networks, which can be described as declarative graphic representations that can be used either to represent knowledge or to support automated systems for reasoning about knowledge, are limited. They mainly work for small and some medium applications [see Appendix A for a literature review of semantics and knowledge representation].

Another problem related to semantics is context. Recently some advanced work has been done, especially by Google (BERT—Bidirectional Encoder Representations from Transformers) that tackles the issue of context effectively. Some people consider context as semantics, which in fact is not the case—context can help compare words or sentences and find similarities between those but cannot answer what the meaning of a sentence is.

Section 4 discusses some details of these problems along with a solution. Machine learning and lifelong machine learning (as discussed in Section 5 and Section 6) are also very relevant to solve semantics, logic, cognitive computing, and relevant problems.

Even more complex issues related to NLP—like thinking, intelligence, and consciousness—are at a very preliminary level from a computer science standpoint. It is also important to note that everything we invent is a reflection of our thinking, and thus whatever we have invented so far has overlap with the human way of thinking and processing things. However, there is still a large gap. For example, today's NLP approaches are based on the algorithms, tools, and techniques that are available today that are still very different from what we as humans do.

When we read a sentence we do not parse it as we do today in NLP to find out its components. As humans we can easily extract the context of a sentence, paragraph, or document after we read it. This is mainly because of the way we use semantics and logic. This is strongly related to the fact that the whole is greater than the sum of the parts. Our brain can easily integrate and compute the whole from the parts, and also parts from the whole, in a natural way which is effective, complete, and done with ease.

It's reasonable to assume that in the future, when we will be using more human-like computing, some of the ways we do NLP tasks will be different and parsing may not be needed for most tasks. Section 4 provides such an example using SEBLA (Semantic Engine using Brain-Like and Brain-Inspired Algorithms).

4. Need for a Semantic Engine

As discussed in Sections 1 and 2, the advancement of NLP is limited by semantics and knowledge representation. NLP and natural language understanding remain complex, open problems. NLU complexity is mainly related to semantics: abstraction, representation, real meaning, and computational complexity. We argue that while existing approaches are good for solving some specific problems, they do not seem to address key natural language problems in a practical and natural way.

Logic is another important component that works together with semantics and knowledge representation as mentioned in Sections 1 and 2. Section 5 is mainly based on logic. These are all critical for a better semantic engine. Clearly, we need an advanced semantic engine that can significantly help understand user questions and requests by summarizing and drawing inferences. Helping conversations by understanding what a person says and coming up with a dialogue that reflects how two or more humans converse requires knowledge representation and a knowledge base. Providing necessary information and continuing dialogue also requires automatic generation of dialogue, which is called natural language generation. Natural language generation is an important part of conversational AI, in order for AI to have natural conversation with a computer. This demands a good knowledge base and associated knowledge representation. It also needs good logic and machine leaning to help address all key NLP problems and enable humans to interact with computing devices in a natural and more effective way.

Moreover, the semantic engine will help process huge sources of information from very large data, about 2 exa bytes (where exa means 10^{18}), that are generated every day from emails, social media, photos, the Internet of things, and the like. Such a Semantic Engine will have "natural semantics" that can help derive new knowledge, and thus will help cognitive computing. It will also support lifelong machine learning, as described in Section 6. Major principles of natural law and ideas from the science of consciousness will help realize an ideal semantic engine. Section 7 discusses strong synergy of the semantic engine with Maharishi Vedic Science.

As we mentioned in the literature review (Appendix A), the best existing methods for a semantic engine are limited by "mechanical semantics" and its scalability, as well as how to represent knowledge. This affects almost all applications of NLP, including information retrieval, Q&A, summarization, language translation, and conversational systems.

It is through semantics that we really understand and communicate. Semantics is the key element for natural language computing and learning via natural language. This is why we have defined some semantics in mathematics, such as semantics of number theory, which supports all the axioms of number theory. However, defining semantics for mathematics or for a computer is more difficult. For example, a Turing Machine, a mathematical model of a hypothetical computing machine that can use a predefined set of rules to determine a result from a set of input variables, ultimately doing mathematical computation, is quite a challenge. Defining semantics in predicate logic is only practical for small applications. We would need to define semantics for all combinations, thus facing combinatorial explosion. In mathematical logic, predicate logic is the generic term for symbolic formal systems such as using first-order logic or second-order logic to formulate propositions. It is the lack of semantics in mathematics that causes difficulty in dealing with issues like infinity, undecidability, incompleteness, and inconsistency.

Maharishi Vedic Science exists within the infinite field of consciousness, and our natural language also exists within this same universal field. We can communicate easily with each other using natural language, but not with machines. We do not have an efficient way to represent the semantics of our natural language and pass it to machines. Thus, the challenge is to ensure that we can effectively define natural language and semantics for mathematics and computers so that:

(1) humans can more effectively communicate with any computing machine
(2) Computing machines can do human-like computing, for example, natural language computing.

4.1. **SEBLA.** To address the issues discussed above with existing semantic engines we proposed in ([5], [6], [7]) an alternate approach using a semantic representation for each word, deriving semantics for each sentence using the semantics of the words in an automated way. SEBLA calculates relevance using semantics and natural language understanding. This approach tries to overcome all the limitations mentioned with existing approaches. As humans, we automatically understand semantics when we read content. No special tag or representation is needed to add onto the content to derive semantics. Our approach uses the same idea. Thus, there is no need to derive or develop ontology, which is the set of concepts and categories in a subject area or domain that shows their properties and the relations between them.

The main theme of the approach in SEBLA is to use each word as an object with all the important features, most importantly the semantics. In our natural-language-based communication, we understand the meaning of every word, even when it stands alone, without any context. Sometimes a word may have multiple meanings which get resolved within the context of a sentence. The next main theme is to use the semantics of each word to develop the meaning of a sentence, as we do in our natural language understanding as humans. Similarly, the semantics of sentences are used to derive the semantics or meaning of a paragraph. The third main theme is to use natural semantics as opposed to existing "mechanical semantics" of predicate logic or ontology.

A SEBLA-based NLU system is able to:

(1) Paraphrase an input text.
(2) Translate the text into another language.
(3) Answer questions about the content of the text.
(4) Draw inferences from the text.

As an example, consider the following sentence: "Maharani serves vegetarian food." Semantics represented by existing methods, for example predicate logic, is serves (Maharani, vegetarian food) and restaurant (Maharani).

Now, if we ask "Are vegetarian dishes served at Maharani?" the system will not be able to answer correctly unless we also define a semantics for "vegetarian dish," or define that "food" is the same as "dish." This means almost everything would need to be clearly defined, which is what is best described by "mechanical semantics." But with SEBLA-based natural language understanding, the answer for the above question will be "yes", without adding any special semantics for "vegetarian dish", as the semantics of "vegetarian food" and "vegetarian dish" are the same in SEBLA.

The nature of "mechanical semantics" becomes more prominent when we use more complex predicates, as when we use universal and existential quantifiers or add constructs to represent time. For example, let us consider how to represent "time" in the following three sentences:

(1) I arrived in New York.
(2) I am arriving in New York.
(3) I will arrive in New York.

Using an existential quantifier in first-order logic, we can write:

"There exists an event e such that
$\text{Arriving}(e) \wedge \text{Arriver}(e, \text{Speaker}) \wedge \text{Destination}(e, \text{New York})$."

However, this is not complete, as it does not represent time. To represent time, we can do something like the following:

"There exists e, i, n, t such that Arriving(e, Speaker) $\wedge$ Destination(e, NewYork) $\wedge$ IntervalOf(e, i) $\wedge$ EndofPoint(i, e) $\wedge$ Precedes(e, Now)" .... (a)

"There exists e, i, n, t such that Arriving(e, Speaker) $\wedge$ Destination(e, NewYork) $\wedge$ IntervalOf(e, i) $\wedge$ MemberOf (i, Now)" .... (b)

"There exists e, i, n, t such that Arriving(e, Speaker) $\wedge$ Destination(e, NewYork) $\wedge$ IntervalOf(e, i) $\wedge$ EndofPoint(e, n) $\wedge$ Precedes(Now, e)" .... (c)

The point is that by defining additional terms like i, n, and t, we can give additional semantics to define *IntervalOf*, *EndofPoint* of the interval, *precedes*, *now*, and *try* to represent the original three natural sentences. Basically, we just have added more "mechanical semantics" since such a scheme does not use the natural semantics contained in the original three natural sentences shown above.

Clearly, the processing of such statements to derive the semantics of multiple sentences with similar structure is more difficult to understand, to compute, to verify, and to generalize. Such schemes are sufficient for small-size applications but very limited for any good-sized practical application.

We do not do so in our natural language communication. Natural semantics is very different [5]. The semantics representation in SEBLA is much simpler, as shown below with some examples. As mentioned above, SEBLA uses each word as an object with all important features that define the function of the word.

The key question we have addressed ([37], [38]) is how to represent the semantics for each word and how to associate appropriate world knowledge with each word. By using the representation and semantic feature of each word along with the world knowledge associated with each word, the meaning of a sentence is derived by applying the grammar of the language and appropriate rules to combine words. Key features of the words and appropriate rules to combine them are learned and refined using large text corpora, existing machine learning algorithms, existing machine learning, and machine learning for semantics. The inference engine is the computing engine, or intelligent agent, that determines the meaning of a sentence by using the word semantics and appropriate rules to combine the words in a sentence.

The key features of a word are the features that define it, essentially a combination of features and functions. For example, the key features of a ball are:

- Something that is round.
- Something that rolls.
- Something that bounces when it hits a wall.

The color of the ball is a secondary feature, as one can identify whether an object is a ball or not without using its color. So a ball is represented in the word feature table as:

{Move, roll, round, bounce back, play, ...}

Similarly, the function word for "go" is {go, move, not static, ...} and the function word for "school" is {school, study, student, teacher, learn}. We can add other related words which are usually implied, for example, for "school", "a place to" {study}, "a place where" {students} go, and so on. In general, a short list of function words suffices and makes it simpler. Note that the word itself is included in its function word. This can be done in the world knowledge, so we may not include the word itself in its function word. Now consider the sentence "I go to school". For semantic retrieval, we will use "I go school" as simplifying the verb phrase from "go to school" to "go school". This is referred to as a *declarative* sentence type.

I {person, he, she, living object, ...} {eat, go, fly, all verbs} go {move, walk, run, ...}. Then, we take only the subject words for "I" and verb words for "go", which yields the following semantics:

I {person} go {move, walk, run, . . . } ....(1)
So the semantics of the sentence "I go to school" is
I {person} go {move, walk, run, . . . } school {study, student, teacher, learn} .. (2)

Note that the main words are there to visualize the sentence better. The real semantics are represented by the words between the curly braces.

Consider another sentence, "I open door", the semantics of which is
I {person} open {open, unlock, push, pull, . . . } door {a thing that blocks, closes, opens, moves, . . . } .... (3)

We can now ask a question such as "What am I doing?" The semantics of this sentence is
I {person} doing {unlock, push, pull, open, . . . , all verbs as included in world knowledge} what {question} ....(4)
A match operation between statements (3) and (4) will yield:
I {person} doing-open {unlock, push, pull, . . . } what-door {question} .... (5)

The match between "doing" and "open" is true as "doing" includes all verbs in the world knowledge. For other words without an auxiliary verb that imply open, synonyms such as "ajar" would be included in the function word "open." In this case, we do not need to use the world knowledge, but it won't hurt even if it is in the world knowledge. Equation (5) can then be processed to yield the answer "I am opening the door" after some grammatical refinement.

To better explain this, let's use an invalid sentence such as "Door walks", which is not valid, as the function words for "door" and "walk" are not there. Besides, "door" is not a living thing and hence it will not be supported by world knowledge. So, the semantics of this would be null or a question mark "?".

A similar approach is used for more complex sentences. Usually, a long sentence is broken into smaller sub-sentences and action words are determined. Action words are verbs that determine the key actions to be followed by computing its relation with other words and the logic involved. Our SEBLA-based approach for declarative sentences will work in a similar way for other types of sentences, including imperative, yes/no, wh-structure, and wh-nonsubject-structure, where *wh* indicates questions that begin with what, when, where, who, whom, which, whose, and why along with how.

Clearly, the representation of semantics is much simpler. More importantly, it contains all the main words that describe the functions in a manner that reflects how we compute, express, communicate, and learn using our natural language.

From the language standpoint, the word "I" (noun phrase, NP) is the subject. The word "go" is the verb which is part of the verb phrase (VP) "go school" where "school" is a noun. Next we apply the function words to calculate the semantics of the first two words, "I go". In doing so, we first take the function words for "I" and "go".

Another important concept used in SEBLA is running semantics, which means that, in computing the semantics of a sentence, we look into the meaning of each word and derive semantics in a running way as more words come until the sentence is completed. At this point, we have the complete semantics of the sentence. Logic is used to determine the running semantics. This helps the real time processing of contents, like the processing in the human brain. Logic is also used in computing general semantics of the whole sentence, as logic is needed along with word semantics. Such logic can be learned by MLANLP (Machine Learning Algorithms for Unstructured Data) as discussed in Section 5.

The semantics of a paragraph is also computed in a similar way [37], using the semantics of sentences in a paragraph. Similarly, the semantics of a document is computed using the semantics of paragraphs. The time, tense, and similar information can be represented in a simple way using the approach shown for SEBLA. It also avoids the combinatorial explosion that is unavoidable with predicate logic and similar approaches. Clearly, SEBLA-based Natural Language Computing (NLC) can solve many problems in computing and machine learning, including question answering, summarization, and drawing inference.

In the next section we discuss MLANLP which uses SEBLA and we show some more details of logic, along with some example applications. However, SEBLA does not solve all the problems that humans can. There is still a large gap with human capabilities. However, borrowing more ideas from nature, as outlined in Maharishi Vedic Science literature, we can keep on improving SEBLA. See Section 7 for more details.

5. Machine Learning Paradigm for Unstructured Data

In the previous section, we have described the semantic engine SEBLA, which can deal effectively with the semantics issue. However, to efficiently handle all semantics related issues in NLP we also need machine learning, so that semantics can be refined over time via learning how to deal with new cases and other possible changes.

Today's machine learning algorithms mainly address the learning of numerical data. These algorithms do not address learning of text data where semantics between words, sentences, and paragraphs are very crucial. All machine learning algorithms that exist today, such as neural networks, support vector machines, probabilistic learning, and genetic algorithms, can be classified into four major categories—Supervised, Unsupervised, Reinforcement, and Evolutionary Learning. Such algorithms use numerical data properties for learning key features, helping mainly classification, regression, and clustering or matching. Such properties use common numerical features like diameter, weight, or shape for coin recognition; or eyes, nose, mouth, eyebrows for face recognition. Feature selection is very important in machine learning to ensure better results with good generalization. Nevertheless, generalization can be inadequate for many such applications, especially when using complex data, large data, or both.

On the other hand, learning the meanings of words, sentences, paragraphs, and documents is the key for many natural language applications. These include information retrieval, question answering, summarization, reliable machine translation, and drawing inference. All these problems use text, meaning non-numerical unstructured data. Classification, regression, and clustering—the major capabilities of existing numerical-data-driven learning algorithms—are not adequate for such semantics-driven applications. As mentioned, the same is true for existing methods to define and learn semantics, which encompasses predicate logic, ontology, frame, and semantic networks. Such algorithms produce *mechanical* semantics. Semantics needs to be defined in a crisper way. Another limitation of mechanical semantics is that new semantics cannot be easily computed or derived from existing ones. Hence, their use has been very limited—mainly for small and medium size applications [50].

We need new machine learning algorithms to address the growing need to handle text dominated data which contributes 80% of data on the Internet, while numerical data is attributed to the remaining 20%. In fact, we need a paradigm shift in machine learning approach for such semantics-driven problems. Such a new paradigm should be able to use and learn semantics of words, compute the semantics of sentences using the semantics of the words, compute the semantics of a paragraph using the semantics of sentences, and compute the semantics of a document using the semantics of its paragraphs. Machine learning algorithms that efficiently learn natural language semantics support this new paradigm. In addition, MLANLP (Machine Learning Algorithm for Unstructured Data) can also derive new semantics and new knowledge from existing ones.

Proper actions need to be learned or computed based on the meaning of a statement or query. Machine learning for unstructured data would need to learn logic and determine appropriate actions. Ideally, the new paradigm would focus on semantics and logic driven learning, as opposed to existing numerical data-driven learning that usually minimizes error, with respect to some objective function. MLANLP is designed to have all these new desired properties. [60] presents such a machine learning algorithm to efficiently learn natural language semantics, called MLANLP , that uses a semantics-driven and logic-driven learning paradigm, using explanation and logic rather than training using a large data set. MLANLP uses SEBLA and its concepts for learning.

MLANLP is similar to the way humans learn. We learn very easily when someone explains or teaches using logic. A few examples may be used to enhance the teaching process, but our learning is primarily based on explanation and logic. We do not use numerous data to learn something unless it is for regression. In contrast, numerical-data-driven machine learning systems learn from many examples of numerical data and do not use explanation or logic to learn. This is the main reason that such systems can only do limited key functions, such as regression, classification, clustering, and matching—with limited generalization.

Another key point in the paradigm shift is that the semantics-driven learning process uses computing and learning at the same time in many cases. This makes good sense, since semantics and semantics-driven computation, learning, and understanding are very closely related. In fact, for most cases, computing is learning. Learning can be considered separate from computing to some extent, as when we derive new semantics and knowledge from existing semantics and existing knowledge.

The generalization capability of semantics-driven and logic-driven learning paradigms is much better, as learning is dominated by semantics and logic. Since the semantics of words are the basic building blocks of learning and computation, there is also no need for a large natural language corpora to learn. In contrast, such large natural language corpora is very much needed when we use probability-based N-grams. However, natural language corpora can still be used to further refine the semantics as appropriate.

MLANLP uses the above-mentioned new paradigms and thus uses a human-learning approach. The learning and computing capabilities of MLANLP enable it to determine the actions to be performed based on the input sentence(s). This means that the input is used to determine how to determine the logic to use. In other words, a programmer needs to write a short high-level program and the MLANLP determines and performs the lower-level tasks automatically using the semantics of the input information. The following examples show how MLANLP deals with a common request or a question:

Example. "Show me the pictures of last Saturday's birthday party from my Facebook account."

Then MLANLP will perform the following:

(1) Go to the Facebook website and log on (assuming that login/password information was already in the system).
(2) Go to the appropriate link for pictures.
(3) Calculate the date for last Saturday considering today's date from the system.
(4) Search the photo link page for birthday party pictures with the specified date.
(5) Identify the pictures that are more relevant using the title, subtitle, or caption of the pictures.
(6) Present the requested pictures in a format usable by the user.

Before showing more complex examples, let's discuss further the semantics and logic of SEBLA and MLANLP.

More Complex Sentences. A SEBLA-based approach also works with more complex sentences. Consider the sentence
"I am trying [VP (Verb Phrase)] to find a flight that goes from Pittsburgh to Denver after 2:00 pm [VP]" .... (6) Note that the sentence starts with Noun ["I"], then VP ["am trying"] and then another VP ["to find a flight that goes from Pittsburgh to Denver after 2:00 pm"]. The word feature in the basic sentence and the question will automatically match, so we do not necessarily break the sentence as mentioned above. This is what traditional semantic role labeling and parsing try to do, so that there is a clear structure of the sentence that makes it easier to formulate an answer to a question.

It is thus similar to

$$\mathrm{S} \Rightarrow \mathrm{NP}\ (\mathrm{VP}\ \mathrm{VP})$$

that is, the constituent of the Verb Phrase (VP) is another VP or we can think of it as

$$\mathrm{S} \Rightarrow \mathrm{NP}\ \mathrm{VP}.$$

But this verb phrase has two verb phrases, namely "am trying" and "to find a flight that goes from Pittsburgh to Denver after 2:00 pm".

Here, the basic idea is to use the sentence starting at the top level and classify it as having a Noun Phrase (NP) and Verb Phrase (VP). Then, deal with the complexity of the VP using a way similar to that described above. So the first level semantics is

I {person} trying {doing something, working on, . . . }

to find {looking, trying to look, ...} .... (7)
as the main verb of the verb phrase "find a flight that goes from Pittsburgh to Denver after 2:00 pm" is "find" or "to find".

Now we can focus on "a flight that goes from Pittsburgh to Denver after 2:00 pm" in a similar way. This reduces to "flight goes" as the rest of the sentence "from Pittsburgh to Denver after 2:00 pm" is from one city to another after "time" 2:00 pm. The semantics for the words before the cities is
I {person} trying {doing something, working on, trying, ...} to find {looking, trying to look, ...} flight {a plane going from one city to another ...} .... (8)

Paragraph Level. Similar algorithms can be used in calculating semantics for multiple sentences and paragraphs. However, some modifications are needed for the following reasons:

(1) Within a sentence, words are used in a way constrained by grammar, but between sentences, there is no such grammar.
(2) Usually, a group of sentences carries a theme within a context and there are relations between sentences.

Thus, to calculate the semantics between sentences, we use word semantics as before but with some modifications. This is also true for a single long sentence segmented by "comma", "semicolon", "but", "as", and the like. We also need to take into account discourse, that is, coherence or co-reference to words in previous sentences. There are some good existing solutions mainly for a small domain problem. But in general, computational discourse in natural language is an unsolved problem. However, with our SEBLA based scheme, the computational discourse problem can be solved to a good extent for large domains. In calculating semantics in a long sentence, the previous, next, and other words can further influence or refine the semantics. For convenience, we have included this aspect in calculating semantics of multiple sentences as described before.

Deriving New Semantics and New Knowledge. To derive new semantics and new knowledge, we use a similar approach with some enhancements, mainly causal relationships and logic. Consider the following three sentences:

I was tired. I fell asleep. I worked and made good progress after I woke up. (9)

Here, we need to find causal relationships by using the semantics between sentences. The semantics of "tired" and "asleep" are related. Then from world knowledge, we know that "tired" may cause "asleep." Thus, the causal relationship between the first two sentences is established. The third sentence, "After I woke up" is opposite of "asleep" and hence a relation is established. And, "made some good progress" is not related to "tired" in the first sentence. The basic information in world knowledge will help compute the causal relationship between the first sentence and the third sentence. Thus, the SEBLA based semantics approach will compute the new derived fact "I made good progress" because my "tiredness" was gone due to "sleep".

This is an important feature for question answering, drawing inference, and summarization where sentences can be shortened—for all Natural Language Computing and cognitive computing that need new facts or new knowledge.

6. Lifelong Machine Learning

A fully capable learning system would need to have most of the learning capabilities of a human—self-learning, creating knowledge, learning from experience, determining what is to be learned and the like; we shall call this Lifelong Machine Learning (LML). With impressive growth of use of Machine Learning (ML) and Artificial Intelligence (AI) in many applications, the need for LML is becoming more apparent. By creating knowledge and learning from previous knowledge or experience continuously across tasks and across domains, LML can help AI-ML growth further. This is related to deriving new semantics and new knowledge from existing knowledge under Cognitive Computing as discussed in Section 5.

Existing Machine Learning (ML) algorithms are dominated by isolated learning (for example, in Supervised Learning, a specific data set for a specific task in a domain is used to train an ML for regression or classification). After learning (via training) is completed, the trained ML system is used to make predictions—no more learning is done. With online learning, the system can learn on a continuous basis and make better predictions. However, the learned information is not converted into knowledge and hence cannot be easily transferred for another task or application.

In other words, existing ML algorithms mainly do instance learning and do not support cumulative learning with knowledge creation. The generalization capabilities of such systems are closely related to data, task, and domain used to train, and hence are limited in scope. Transfer Learning can help to a good extent though for some applications. But such systems do not create knowledge and hence cannot learn from previous knowledge or experience across tasks and across domains.

In [43] we presented an LML approach using logic, semantics, and Natural Language, especially with Semantic Engine using Brain-Like Approach (SEBLA). Natural Language is an effective way to represent knowledge. It is an effective way to accumulate knowledge, learn from experience, and derive new knowledge and intelligence, just as we humans do. We also use a neural network and fuzzy logic combination (NeuFuz) to convert existing data-driven ML systems knowledge into fuzzy logic rules and natural/semi-natural language sentences to easily integrate with natural-language-based knowledge.

Here we focus on how SEBLA, logic, and MLANLP, discussed above, can be integrated with LML system to provide an integrated system that can help solve all NLP problems under one framework while also integrating existing machine learning knowledge to the LML system. Such an LML system is called LMLS_NL_SEM_LOGIC (Lifelong Machine Learning System driven by natural language, semantics and logic), as shown in Figure 1.

User inputs, such as query or request, that use text, voice, and other methods, are processed by SEBLA, MLANLP, and Logic, and stored in the natural-language-driven knowledge base, and then further processed by Q&A, Summarization, Inference, and other engines to create appropriate outputs for the user. If some necessary information is not available in the knowledge base, SEBLA will get the relevant information from the Internet (or similar sources), process it as appropriate, and then add to the knowledge base. Existing ML Systems are integrated via NeuFuz (combination of neural network and fuzzy logic) where ML knowledge of trained neural network is converted to fuzzy logic rules which are then processed by SEBLA, MLANLP, and logic in the same way as other user inputs.

Let's use an example application to explain the concept. Consider minutes from two meetings about going on a vacation.

Meeting 1: May 20; Attendees: Bob, Ron, Shelley, and Sandy. Topics: Duration of the vacation, countries to visit, making a cost estimate [Start date June 21]

Minutes: two weeks vacation, would like to visit China and Australia, approximately $7,000 per person.

Some key conversations:

(1) "We must see the Great Wall and visit Beijing and Shanghai."
(2) Bob: "Sounds good, but I prefer to stay longer in Shanghai."
(3) "Let's talk about key places to visit in Australia once we are in China."

Meeting 2: June 29; In Shanghai. Attendees: Bob, Ron, Shelley, and Sandy. [Possibly more meetings between May 20 and June 29] Topics: Plan change; How to accommodate conflicting interests

Minutes: "Sandy has an emergency and would need to return soon." "Bob would like to skip Australia and is thinking about going to Indonesia."

Some conversations:

(1) Bob: "I don't mind going to Indonesia by myself."
(2) Shelley: "I would prefer to go to Australia."
(3) Ron: "I am not sure whether to stay longer in China or go to Australia with Shelley. Maybe I have seen enough in China."

Inference: Where all travelers could be on June 30.

The semantics of each sentence in each meeting are computed using SEBLA (see Section 3). The action words of each sentence are used to determine the intent and actions for each sentence using the MLANLP (Section 3). Total actions, content for a query, summary, or inference from a few paragraphs or documents are computed using various blocks as appropriate (Figure 1). So, the knowledge base for Meeting 1 and Meeting 2 are accumulated easily and placed in the knowledge base. The world knowledge, common sense computation, and inference are also included in the knowledge base. For example, two weeks vacation for two countries means one week for each country, as nothing specifically is mentioned otherwise. The knowledge base can also create or derive new knowledge using MLANLP and logic.

On June 30, Sandy will be in San Francisco, Shelly will be in Australia, Bob will be in Indonesia, and Ron will most likely be in Australia. The inference for the location of Ron is the most difficult one. But the conversation 3 in Meeting 2, implies with some confidence that Ron has spent enough time in China. If more information is given, a better inference could be made. This is just a simple example to explain the concept. Of course, for complex cases, the system may fail sometimes in giving correct answers. But in general, it will still give something reasonable. The knowledge base will usually have hierarchy and various sub-domains—like History, Economics, Science—so that it is manageable and easy to search, while avoiding combinatorial explosion when integrated with the existing ML system (Section 5).

It is very important to integrate the existing ML system with the LML system so that we have a more complete, capable, and efficient LML system, like

LMLS_NL_SEM_LOGIC. A novel way to do so is to use a Neural-Fuzzy system where a Neural Network is trained and then its knowledge (the weights) is automatically converted to Fuzzy Logic Rules and Membership Functions. Since Fuzzy Logic Rules are similar to Natural Language, such rules can be easily integrated with LMLS_NL_SEM_LOGIC discussed in Section 6. NeuFuz [50, 51, 52] is a system where Artificial Neural Net (ANN) algorithms are used to generate fuzzy logic rules and membership functions. The combination of learned fuzzy rules, membership functions, and a fuzzy design technique based on new fuzzy inferencing and defuzzification methods significantly improves performance, accuracy, and reliability and reduces design time. NeuFuz also minimizes system cost by optimizing the number of rules and membership functions. Figure 1 shows how NeuFuz is integrated with LMLS_NL_SEM_LOGIC.

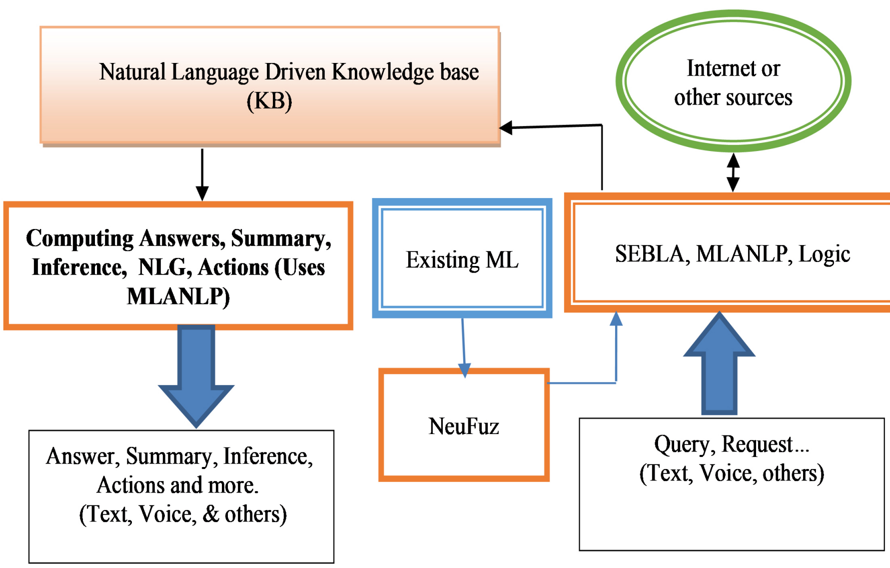


FIGURE 1. A Natural Language Driven Lifelong Machine Learning (LML) Architecture (LMLS_NL_SEM_LOGIC) that uses NeuFuz (Neural Nets and Fuzzy Logic Combination) to integrate existing ML. This is well suited for advanced analytics, cognitive computing, and intelligent agent.

The neural net is properly architected so that it maps well to fuzzy logic rules and membership functions (Figure 2). The first layer neurons in Figure 2 include the fuzzification process, whose task is to match the values of the input variables against the labels used in the fuzzy control rule. The first layer neurons and the weights between Layer 1 and Layer 2 are also used to define the input membership functions. In fact, it is difficult to do both fuzzification and learning membership functions just by one layer of neurons.

Figure 3 shows a multiple layer implementation for fuzzification and membership function generation. Both linear (L) and non-linear (NL) neurons are used. With

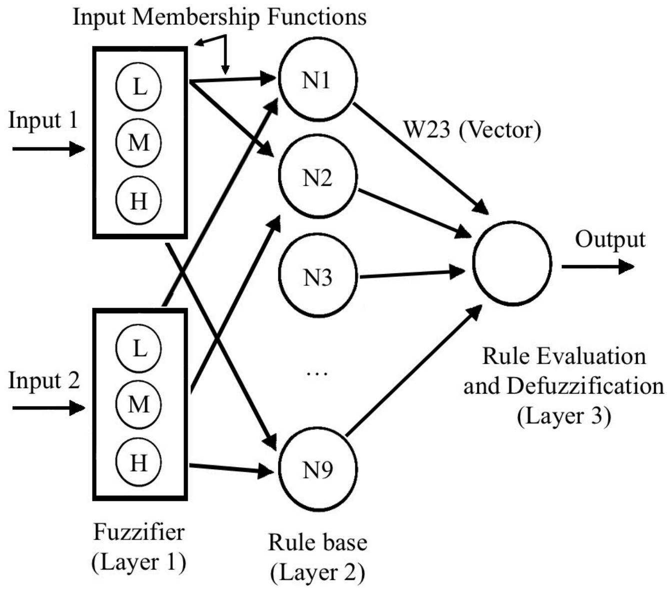


FIGURE 2. Neural network and corresponding fuzzy logic representation in simplified form. The net is first trained with system input-output data. Learning takes place by appropriately changing the weights between the layers. After learning is completed, the final weights represent the fuzzy rules and membership functions. The learned neural net, as shown above, can generate output very close to the desired outputs. Equivalent fuzzy design can be obtained by using generated fuzzy rules and membership functions.

an input level of x, the output of Layer 1 neuron is $gl.x$ where gl is the gain of neuron in Layer 1. The input of Layer 2 neuron is $gl.x.W1$. Continuing this way, we have the input of the fourth layer neuron, z to be

$$\begin{aligned} (6.1)\quad z &= (g1.x.W1.W2.g2 + b).w3\ldots & (9)\\ (6.2)\quad &= (a.x + b).c \end{aligned}$$

where $a = gl.g2.WI.W2$, $c = W3$, $g2 =$ gain of the Layer 2 neuron. The output of Layer 4 is the membership function which is the same as the output of Layer 1 in Figure 2. Thus, NeuFuz uses a six layer neural net [50]. Neurons in Layers 1–4 correspond to the membership functions. Neurons in Layer 5 (Layer 2 in Figure 2) correspond to fuzzy logic rules. A sample Fuzzy Logic Rule is (equation 6.3):

(6.3) "If Input 1 is Low and Input 2 is Low THEN the output is $W23$." (10)

where $W23$ is the weight between Layers 2 and 3 in Figure 2. The neuron in Layer 3 does the rule evaluation and defuzzification.

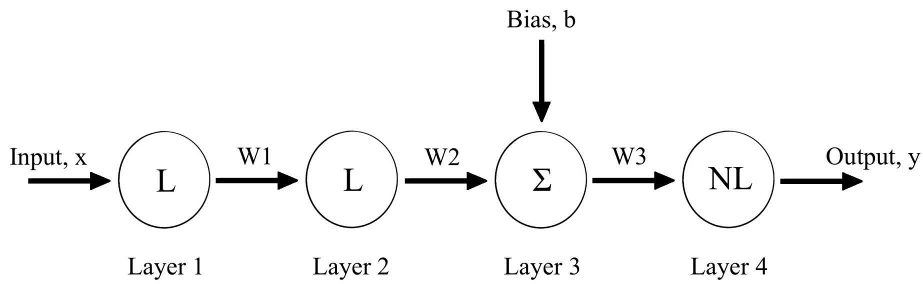


FIGURE 3. Neural network architecture designed to learn fuzzy membership functions.

The Neural Fuzzy approach typically uses nonlinear membership functions, as shown in Figures 2 and 3. The advantage of using a nonlinear membership function is that the system knowledge can be distributed evenly between the rule base and the membership function base. This results in a reduced rule base, helps prevent combinatorial explosion, and saves memory and overall cost.

Most importantly, a Neural Fuzzy System's learning and generalization capabilities allow generated rules and membership functions to provide more reliable and accurate solutions than with alternative methods. In conventional approaches, one writes rules and draws membership functions, then adjusts them to improve the accuracy using trial-and-error methods. However, with the proper combination of fuzzy logic and neural networks (such as NeuFuz), it is possible to completely map 100% of the neural net knowledge to fuzzy logic. This enables users to generate fuzzy logic solutions that meet a pre-specified accuracy of outputs. This is possible because the neural net is able to learn to a pre-specified accuracy, especially for the training set (the accuracy for the test set can be controlled to be very close to the accuracy of the training set by properly manipulating the learning parameters), and learned knowledge can be fully mapped to fuzzy logic. Full mapping of the neural net to the fuzzy logic is possible when the fuzzy logic algorithms are all based on the neural net architecture. Such an elegant feature is not possible in conventional fuzzy logic, in that one cannot write fuzzy rules and membership functions to meet a pre-specified accuracy. NeuFuz capabilities and performance can further be improved using Recurrent Fuzzy Logic [53].

7. Strong Synergy With Maharishi Vedic Science

The concepts and approaches presented related to semantic, logic, ML, and LML have strong synergy with Maharishi Vedic Science. We begin our discussion with the unified field, an important topic of quantum physics which is comparable to the unified field of the natural law that has been known throughout the ages by the ancient Vedic tradition.

Recent progress in quantum physics has developed a theory that unifies all the fundamental force fields and fine particles known to physics into one single field known as unified quantum field with compelling evidence that it is the field of consciousness [55]. This has profound implications for all areas of science and technology. The unified field is understood by quantum physics to be the ultimate origin of all the laws of nature studied in all fields of science and applied in all areas of technology [27]. However, the unified field of natural law has been known throughout the ages by the ancient Vedic tradition, the oldest tradition of human knowledge. Maharishi Mahesh Yogi [55] revived the ancient Vedic tradition and its expression in the language of the modern scientific age, called Maharishi Vedic Science.

7.1. **The unified field.** The unified field is the ultimate source and basis of all aspects of the universe. Modern physics has isolated a few fundamental force and matter fields whose interactions are responsible for all aspects of the function of natural law in the universe. The four fundamental force fields are electromagnetism, weak and strong nuclear forces, and gravitation. The fundamental matter fields are quarks, neutrinos, and charged leptons (including electrons). Unified quantum field theories describe a completely unified field, whose self-interacting dynamics give rise to all the fundamental force and matter fields of natural law [27].

Since the unified field is the ultimate source and basis of all aspects of the universe, it must also be the source of human life and human consciousness. The exact relationship between consciousness and the unified field is central to an understanding of unified field-based computing including natural law (Natural Language) and Natural Language Processing (NLP).

An analysis of one of the mathematical equations of unified field theory by theoretical physicist John Hagelin has located properties that are the same as those of consciousness: self-interaction, self-referral, dynamism, orderliness, and intelligence [54]. The quality of self-interaction makes the unified field dynamic in its nature, and this internal dynamism serves as the basis of the emergence of specific force and matter fields in the structure of the unified field. Since all natural laws emerge from the unified field, the order and precision exhibited by the expressed levels of natural law must have their origin in the perfect orderliness inherent in the unified field. As the ultimate source of nature's functioning, the unified field can also be considered as the most concentrated state of intelligence in nature, underlying all natural phenomena and giving a direction to all activities through the various channels of natural law [55].

There is a growing body of theoretical and experimental evidence that the unified field of natural law is actually a field of consciousness, a unified state of "pure consciousness." Individual human consciousness and all the subjective values of life—ego, intellect, mind, and senses—arise as excitations or "waves" of this unified state of consciousness, which resides in its pure unbounded form at the source of thought deep within the human mind.

This identification of the unified field with pure consciousness was first proposed by Maharishi Mahesh Yogi, based on the ancient Vedic tradition, which describes a "unified" state of pure consciousness as the ultimate source of all aspects of natural law. According to Maharishi Vedic Science, the universe emerges from this all-pervading, unbounded field of pure consciousness (or pure intelligence), which becomes aware of itself due to its own nature as consciousness, thereby creating a self-interacting dynamical relationship of knower, process of knowing, and known within itself. This self-referral interaction of pure consciousness with itself is responsible for (as if) breaking the unity and creating three from within the structure of its own oneness. Further interactions of the three (knower, process of knowing, and known) with each other and with the unity of pure consciousness then give rise sequentially to increasingly diverse levels of nature, resulting in all the localized values of natural law, including physical matter and life. In Maharishi Vedic Science, the unity of pure consciousness is termed Samhitā and the three values created from that are Ṛishi (knower), Devatā (process of knowing), and Chhandas (known) (Maharishi Mahesh Yogi, 1985). The unified field is the ocean of infinite silence.

Thus, both modern theoretical physics and Maharishi Vedic Science describe a "unified field" whose self-referral dynamics give rise to all forms and phenomena in nature. Quantum physics uses the language of modern mathematics, gradually developed through experimental and theoretical investigation of finer levels of physical matter, to describe the unified field. Vedic Science is derived from the direct experience of the unified field of consciousness by the ancient Vedic seers, who were able to directly cognize the mechanics of the creation of diversity from unity, and give expression to this knowledge in the form of the primordial sounds of natural law as it begins to manifest within the unified field through a process of self-interaction. These primordial sounds are known as Veda, which means "pure knowledge."

Tony Nader ([1]) describes the unified field as follows: "Maharishi's Vedic Science identifies the unified field as an unbounded field of consciousness—an eternal, silent ocean of intelligence that underlies all forms and phenomena. This field of pure consciousness is the unified element in Nature on the ground of which the infinite variety of creation is continuously emerging, growing, and dissolving."

Dr. Nader also shows that "consciousness is all there is" in this universe [1]. In [2], Nader illuminates with the following: "Consciousness is understood as primary. That is, the essence and foundation of life is pure consciousness, and all disciplines of knowledge—indeed, the entire cosmos—are expressions of pure consciousness, much as the waves on an ocean are expressions of the ocean itself."

The latest research shows that our mind has some physiology or subtle body or thought body which is made of new dark matter (also called Hidden Sector, super cold [$-270°$ C] micro-charged particles) that controls the motion of galaxies [56]. It is important to note that our mind is governed by quantum mechanics, which includes quantum entanglement.

7.2. **The Cosmic Algorithm of Natural Law.** Through the knowledge of the unified field provided by Maharishi Vedic Science, it is now possible to begin to study the "algorithms" of natural law directly within the unified field. According to Maharishi Vedic Science, there are guiding structures of intelligence in the unified field that control all the expressed activity of the laws of nature. These guiding structures can usefully be considered as "abstract algorithms" of natural law with properties similar to conventional electronic computer algorithms [27].

The three essential values of control, data, and operations (as typically used in a software program) can be located in their perfect and totally integrated state within the unified field of natural law, which is the source of every abstract algorithm.

According to Maharishi Vedic Science, the fundamental values of the unified field are Samhitā, which means togetherness or unity, and the three values of Ṛishi, Devatā, and Chhandas, which arise when Samhitā as knower (Ṛishi) becomes aware of itself as known (Chhandas) and thereby creates a process of knowing (Devatā) within the unmanifest structure of the unified field. The internal dynamics of this three-in-one structure is the basis for all functioning of natural law in every aspect of the universe.

The English words "knower," "process of knowing," and "known" do not, however, capture the full meaning of Ṛishi, Devatā, and Chhandas. According to Maharishi Vedic Science (Maharishi Mahesh Yogi, 1985), Devatā embodies the fundamental impulses of intelligence that govern all activity emerging from the unified field of natural law. Devatā is the essence of all "operations" in the unified field. Chhandas, as the known, is the more concrete objective value within the unified field, the "data" objects of the unified field. Ṛishi, the knower, is the unmanifest intelligence guiding Devatā and Chhandas, the pure intelligence value at the basis of every aspect of natural law. Thus, Ṛishi is the "controlling" aspect of the unified field. In its role as "knower," Ṛishi is the starting point for all steps of evolution and thus controls the direction of the sequential flow of natural law.

According to Maharishi, Samhitā may be characterized as a three-in-one structure that is continually pulsating within itself from one to three to one at infinite frequency, creating the primordial activity at the basis of natural law. Samhitā becomes Ṛishi, Devatā, and Chhandas, and each of the three becomes Samhitā. Ṛishi, Devatā, and Chhandas are also constantly being transformed into each other through their roles as knower, process of knowing, and known. All of these mutually interacting relationships create new dynamical relationships and continue the process of differentiation and manifestation of unity into diversity. The first phase of this manifestation is the emergence of the Veda, the primordial sounds of the natural law that exist within the structure of the unified field itself, and are therefore nonchanging and eternal. These primordial sounds of Veda, which govern every phase of natural law, can be called the "abstract algorithm" of natural law, structured within the unified field that governs all activities in the universe.

7.3. **The Eternal Structure of Veda.** Maharishi Vedic Science provides a detailed description of the structure and internal self-referral dynamics of the Veda. The most essential core of the Veda, giving rise to all other aspects of the Veda, is called Ṛk Veda. Ṛk Veda is traditionally divided into ten parts called *Maṇdala*, which may be considered analogous to the modules that constitute ordinary computer software packages [27]. Maharishi [56] has shown that each of the ten *Maṇdala* is primarily responsible for the manifestation of some specific aspect of natural law.

The first *Maṇdala* contains the fullness of knowledge of natural law, considered to be the seed of the entire Ṛk Veda and the entirety of natural law in the universe. The second through sixth *Maṇdala* contain the laws responsible for governing the five subtle elements or Tanmatras, forming the basis for all the physical aspects of the universe: Prithivi (earth) Tanmatra, Jala (water) Tanmatra, Agni (fire) Tanmatra, Vāyu (air) Tanmatra, and Ākāsha (space) Tanmatra. Hagelin [54] has equated these Tanmatras with the five fundamental categories of quantum fields, or "spin types," responsible for the entire material universe. These are the spin–2 graviton (responsible for the force of gravity), the spin–3/2 gravitino, spin–1 force fields, spin–1/2 matter fields, and spin–0 Higgs fields. The seventh through tenth *Maṇḍala* are concerned with the subjective aspects of life: mind, intellect, ego, and Self, respectively.

This description from Maharishi Vedic Science closely parallels the mathematical formulation of unified quantum field theories, especially superstring theory, in which all the fundamental force and matter fields of natural law arise from vibrations of a self-referral loop called the superstring. In superstring theory, the five quantum-mechanical spin types correspond to massless modes of vibration of the superstring at the basis of natural law. Hagelin [54] has suggested that the four subjective aspects of natural law governed by the seventh through tenth *Maṇḍala* may correspond to massive superstring vibrational modes within the unified field.

The ten *Maṇḍala* of Ṛk Veda exist eternally within the structure of the unified field and contain in seed form all the laws of nature responsible for governing all processes in the universe. To continue the analysis of the Veda as the cosmic software of natural law, these ten *Maṇḍala* at the core of the Veda may be considered as the ten fundamental "modules" in the cosmic software package of the unified field of natural law.

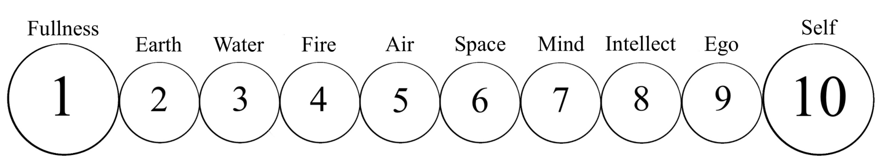


FIGURE 4. Ṛk Veda, the cosmic software of natural law, is traditionally a circular structure divided into ten parts called *Maṇḍala*, which may be considered as the modules that constitute ordinary computer software packages. The first *Maṇḍala* contains the fullness of knowledge of natural law. The second through sixth *Maṇḍala* contain the laws responsible for governing the five subtle elements or *Tanmatra*. Hagelin [55] has equated these *Tanmatra* with the five fundamental categories of the quantum fields. The seventh through ninth *Maṇḍala* govern the subjective aspects of life.

Figure 4 is a schematic diagram developed by Maharishi (1979) to illustrate the structure of the ten *Maṇḍala*. Since this structure exists within the unified field beyond the boundaries of spacetime geometry, it is eternal and nonchanging: this eternal structure contains the fundamental knowledge at the basis of the functioning of natural law at all times. In Figure 4 each *Maṇḍala* is drawn as a circle (the word *Maṇḍala* means "circle"), and this circular structure is partially responsible for the eternal and invincible nature of natural law. Although this property is not shown in Figure 4, each of the ten *Maṇḍala* arises from and is connected to one central "point," which contains the total potential of natural law. Maharishi (1979) explains that this total structure of the ten *Maṇḍala* of Ṛk Veda is contained within each point of the universe. Thus, the total potential of natural law is available at every point of the universe.

It is important to understand, however, that the Veda is not an intellectual formulation of knowledge "about" the unified field. According to Maharishi, the Veda is the structure of natural law within the unified field as directly cognized by the ancient Vedic seers during their direct experience of the unified field [61], p. 195.

Examining the detailed structure of each *Maṇḍala* will reveal more about the structure of the programming language of nature used to construct modules. Each of the *Maṇḍala* is divided into parts called *Sūkta*, ranging in number from 70 to 192 in each *Maṇḍala* (the average is about 100 *Sūkta* per *Maṇḍala*). These *Sūkta* are analogous to subroutines or procedures in conventional computer programming languages, each having its own control (Ṛishi), operations (Devatā), and data (Chhandas).

Each *Sūkta* consists of 5 to 50 verses with the average being about 10. The entire Ṛk Veda consists of approximately 10,000 verses. Each verse has an average of ten words and may be considered as a basic "instruction" in the programming language of the natural law. The length of a computer program is usually defined as a number of "lines of code," the total number of instructions in the program. The kernel of the operating system, which is the central controlling software package for all the activity of the computer, may typically be about 10,000 lines of code. Thus, the Ṛk Veda with its 10,000 verses may be considered as the "kernel" of the operating system of natural law.

7.4. **The Programming Language of Natural Law.** Maharishi has done an elaborate analysis of the first verse of Ṛk Veda and concluded that it contains the concentrated knowledge of the total functioning of natural law (Maharishi, 1976). It is the seed for the entire first *Maṇḍala* and the whole of Ṛk Veda. Considering this first verse as an "instruction" in a programming language, it is possible to locate many of the properties of control, operations, and data ordinarily associated with computer programming languages. The first verse of Ṛk Veda is as follows:

Agnim īle purohitaṁ yagyasya devam ṛitwijam hotāraṁ ratna dhātamam

The first word *Agnim* is itself a complete expression of all knowledge (Maharishi Mahesh Yogi, 1976, p. 128); with its own internal dynamical structure having aspects of control, operations, and data. The word *Agnim* contains in seed form the complete knowledge of natural law, all evolutionary processes, and all structures and activities in the universe. The first letter "A" also has values of *Gyāna* (knowledge), *Gamana* (action), *Prapti* (achievement), and *Moksha* (fulfillment) [61], p. 197. Thus the entire source, course, and goal of action is contained in the first letter of Ṛk Veda, "A" representing the fullness of all possibilities.

There is an aspect of Maharishi Vedic Science that analyzes the meaning of roots contained in the words of the Veda. However, the Vedic sounds have the property that there is an exact correspondence between the sound and its meaning. Therefore, it is possible to gain insights into the meaning of a letter or word just from its sound. The first letter "A" is pronounced "ah" with a full opening of the mouth and throat that corresponds to its meaning as fullness. The second letter of *Agnim* is "G" (as in "got"), pronounced by closing the throat and stopping the continuous flow of the first letter "A." In contrast to the fullness of "A," the second letter "G" represents emptiness, the unmanifest state of the unified field.

Between the fullness of all possibilities in "A" and the unmanifest emptiness of "G"—the two most extreme values of natural law—lies the entire range of natural law.

In the first two letters of Ṛk Veda, the origin of the binary number system used in electronic computers for all computation can be located. The fullness of "A" is symbolized by the digit 1 and the emptiness of "G" is symbolized by the digit 0. In a computer, all possible information and all possible numbers can be represented as patterns of these 1s and 0s, and all computation consists of transformations of these patterns [27]. Thus, 1 and 0 in the electronic computer represent all possibilities for computation just as "A" and "G" in the Veda contain the seed of all possibilities of natural law. For a further discussion of the meaning of the first word of Ṛk Veda, the reader is referred to [61], pp. 197–198.

7.5. **Natural Law and Natural Language Processing from Pure Consciousness.** Now, let's use the basics of Maharishi Vedic Science as described above and apply it to natural law and NLP. As we have seen, the three-in-one aspect (that is, knower, process of knowing, and known) is the main "Cosmic Algorithm" that exists in the whole universe. This highly interactive and dynamic process is the main reflection of how natural law works. Ṛk Veda has the main cosmic software or code to support cosmic computation. When we practice the Transcendental Meditation and TM-Sidhi program [57], we transcend to the unified field and get access to pure consciousness. Accordingly, we get access to pure intelligence. From this level we can think, understand, and develop better intelligence, which can be applied to any field of science or engineering. We develop the appropriate intelligence for each field we are interested in. For example, in mathematics we try to think, understand, and take appropriate intelligence suitable for mathematics. For computer science, we try to develop the appropriate intelligence suitable for computer science. The same is true for other fields.

We use our natural language as the key medium of expression of our thoughts and intelligence for many fields, including mathematics, science, and the arts. However, if we try to express the field of natural language itself using our natural language, it is complicated. Even though it is self referral, we are focusing on different aspects, such as semantics, logic, deriving new knowledge, and so on. Thus, it involves more interactions with our thoughts and intelligence.

7.5.1. *Complexity in Natural Language Computation and Possible Approaches for Solution(s).* Clearly, natural language computation is very complex as briefly explained above. To make it simple, let's start with understanding how the self-referral works in general and do a parallel, say, for the field of computer science or mathematics. By definition, self-referral means referring to one's self. Let's consider a subroutine call in a computer software program which is similar to calling the same *Maṇḍala*. Consider a subroutine to compute the area of a circle. In this case, the subroutine just needs to pass one parameter which is the radius of the circle; thus, we can write Area_of_Circle (r) where r is the radius expressed as a real number.

Each time we call such a subroutine, we provide different values of the radius to get corresponding areas of circles. In general, subroutines can have several arguments or parameters which may include numbers, integers, string etc. However, for NL, the arguments or parameters cannot be represented with numbers or strings that will work in general. Number systems were invented for counting or computing but not for natural language computation, which involves the real meaning of words and sentences. Number system operations such as addition, subtraction, or multiplication are good for numerical computation but not well suited for NL computation.

Let's use an example: "I go to school". Here, as we read the sentence by reading each word, we sequentially compute the meaning of the partial sentence using the meaning of its words. So, after we read first three words, we know we are talking about "I go to 'somewhere' ". Thus "I go to" has a meaning which is computed as a running semantics of the sentence computed so far using the first three words that we have read.

If we represent all the words with numbers and if we add the first three words, the result will not be computed as the meaning of "I go to". If we define the meaning of "I go to" with another number, then it may work for this instance, but not for all instances with more words or a missing word (like "I go"). It will also not work for changing the sequence or style of saying a sentence, for example from "May I take a look?" to "Let me take a look". Thus, it does not work in general. For some small application it may work, for example in predicate logic where we define the meaning of everything. But for our natural language vocabulary and natural language computation, it does not work. Thus, the typical self-referral as used in a software subroutine does not work for NL. Besides, a word may have multiple semantics which often depends on the context. Hence, representing context and semantics are very important to ensure that we use contexts and semantics more appropriately. *Just to compare this with the area of a circle subroutine mentioned above, the meaning of radius "r" changes in each self-referral in case of NL; and when the meaning changes, the computation equation of the subroutine also changes, that is, meaning of "r" can change the computation process of the self referral call.*

Hence, the self-referral in the Law of nature is more complex that allows self-referral with wide variations—some examples of possible variations in each self-referral are:

- Different number of parameters
- Different values and/or number of parameters based on the current results
- Different representation of parameters
- Computing semantics of a sentence using the meaning of each word
- Computing meaning of a paragraph using the meaning of its sentences
- Passing parameters that can be function or functions of some previously passed parameters or results
- And more

This means that a *Maṇḍala* may pass all equivalent parameters, Sūtra, or combination of Sūtra, Sūtra from another *Maṇḍala* (for example, from *Maṇḍala* 7–10 in Figure 4). This may also include other interactions, computations, self-referral of multiple different *Maṇḍala* to support logic, deriving new knowledge and thinking in addition to computing semantics.

We also need to learn how to come up with good natural language content which can be converted to intelligence and thoughts, so we are talking about the creation of expression and listening to and understanding the expression to convert it to thoughts and intelligence.

A few key questions we need to ask are:

- What aspects of natural law are most appropriate for natural language?
- What set of cosmic algorithms are appropriate for natural language?
- Which *Maṇḍala* are appropriate for natural language?
- What *Sūkta* (of each appropriate *Maṇḍala*) are relevant to natural law?
- How exactly do all these work together under natural law using the three-in-one (Ṛishi, Devatā, and Chhandas) dynamic process?

Consider the possibility that the the process of knowing (Devatā) has sub-processes and variations depending on what we need to learn—science, engineering, the arts, and so on. The same might be true for aspects and sub-aspects of natural law, Cosmic Algorithms and sub-Cosmic Algorithms, *Maṇḍala* and their parts—*Sūkta*.

We also need to figure out how "semantics" is represented. From the example of *Agnim*, it is clear that natural law uses the equivalent of 1s and 0s. However, to represent semantics, direct use of 1s and 0s is difficult to grasp—layers of sequences of 1s and 0s at lower levels are represented by a small sequence of 1s and 0s at higher level representing semantics. Or, perhaps, semantics is not directly part of the Cosmic Algorithm; rather it is a realization of part of the computational process of the Cosmic Algorithm and its representation use different means such as symbols, possibly derived from the Veda. Or it may be due to the vibrations of a self-referral loop called the superstring. This idea is possible as whatever intelligence our mind has, we try to reflect that in natural language so that we can express it for any field—mathematics, science, art and so on. The same logic applies to finding out how to represent our knowledge.

However, at the high level—since consciousness is all there is—we know that our natural language and associated semantics come from our thoughts and intelligence, which come from consciousness.

In order to answer the above key questions, we can possibly use the following approaches:

(1) Go through all the verses of the Vedic literature and see whether some answers are there or can be derived.
(2) Explore and understand the details of natural law, Cosmic Algorithms, *Maṇḍala*, *Sūkta*, their relationships, and how all work together interactively and dynamically at all levels.
(3) Try to understand from the point of view of quantum mechanics—mainly from string theory.
(4) Practice the Transcendental Meditation and TM-Sidhi program to develop consciousness.

Among these four approaches to understanding, 4 seems to be the most practical. However, the other methods 1 to 3 should be explored.

But no matter which way we approach the above questions, it is clear that the natural language semantics we know today is a direct reflection of our thoughts and intelligence from the unified field.

To realize all the above in computing machines, we need to know how to express our thoughts and intelligence using natural language into some algorithms that we can implement in the available computing machines. As we know, the most important part of natural language is the "semantics," the "meaning" of words, sentences, and paragraphs.

It is semantics by which we truly understand and communicate; it is the key element for Natural Language Computing (NLC) and learning via natural language. This is the main reason we can communicate easily with each other, but not with machines using natural language. We do not have a way to represent semantics of our natural language and pass it to machines. As discussed, existing NLP algorithms do not perform well for semantics, logic, learning unstructured data, deriving new knowledge, lifelong machine learning, and so on.

Since SEBLA uses the approach of defining semantics based on the features and functions of each word, SEBLA follows our natural language process. LMANLP uses an approach similar to what we use for human learning. The same is true for logic and LML, as these are based on human logic and human lifelong learning. Thus, SEBLA, MLANLP, logic, LML, and their integration follow the natural way, at least partially. Hence, this is one solution to address key existing issues of NLP. While these are good algorithms with better alignment to what we think we currently do to improve NLP, these are still far from what we can do as humans using our brains and minds. We need to better align algorithms with the algorithms our minds use to align with natural law, cosmic algorithms, modules, and programming that originates in the unified field.

8. Conclusions

In this paper we have described and analyzed existing Natural Language Processing (NLP) algorithms, processes, and techniques, and discussed associated key problems. It is important to note that NLP is a rapidly growing field. Many researchers have been working on it over for 60 years and made great progress. Today's technology enables us to talk naturally to computing devices (Siri, Alexa, Google Voice, and the like).

However, the length of a conversation possible today is relatively short, mainly due to the high level complexity of NLP itself, which is strongly related to NLP's key problems of semantics: abstraction, representation, real meaning, and computational complexity. In fact, NLP remains a complex, open problem. The representation of semantics is not simple but is strongly related to knowledge representation, which also remains an open problem. The other key problems associated with NLP are learning, logic, and cognitive computing capabilities. We have emphasized using our brain-like and brain-inspired algorithms to solve the semantics and knowledge representation problems in NLP. Our idea is to borrow from our natural language as much as we can to solve NLP problems. Using this high level concept, we have shown that SEBLA (Semantic Engine Using Brain-Like and Brain-Inspired Algorithms) can:

- Paraphrase an input text.
- Translate the text into another language.
- Answer questions about the content of the text.
- Draw inferences from the text.

We have also shown that while existing Machine Learning (ML) is great for many applications, it does not work well on semantics since

- Semantics is not something suitable to learn with numerical data driven ML algorithms
- Representation of semantics is basically based on predicate logic.

We have shown that an ML algorithm for unstructured data (for example, NLP text data) is needed to help realize machine learning for NLP. We call it MLANLP (Machine Learning Algorithms for Unstructured Data). This also addresses the need for logic and cognitive computing (deriving new knowledge from existing knowledge). We have also emphasized that to more completely solve NLP problems, we need to support integrative learning, continuous learning, and cognitive computing. Accordingly, we have shown such a system which is called A Natural Language Driven Lifelong Machine Learning (LML) Architecture (LMLS_NL_SEM_LOGIC) and algorithm. Additionally, to integrate knowledge from existing ML systems, we have used NeuFuz (Neural Nets and Fuzzy Logic Combination). NeuFuz basically converts the knowledge of existing ML systems into Fuzzy Logic Rules which are similar to our natural language and hence can be integrated using the basic property of our natural language. LMLS_NL_SEM_LOGIC is well suited for Advanced Analytics, Cognitive Computing, Intelligent Agent and many other applications.

We have shown that our approach is linked with the “Cosmic Computing” of the natural laws, thus linked to pure consciousness at the level of the unified field. Intelligence and thoughts are directly related with pure consciousness. Natural language is the key for the expression of thoughts and intelligence. Thus, natural language is strongly related to pure consciousness. Logic, semantics, and learning work together very closely to create expressions from thoughts and intelligence, to understand and convert expressions to thoughts and intelligence, using natural language.

Since SEBLA uses the approach of defining semantics based on the features and functions of each word, SEBLA follows our natural language process. Similarly, LMANLP uses an approach similar to what we use for learning. The same is true for logic and LML, as these are based on human logic and human lifelong learning using our natural language. Thus, SEBLA, MLANLP, logic and LML, and their integration follow the natural way, at least partially. However, while these are good algorithms with better alignment to what we think we currently do to improve NLP, these are still far away from what we can do as human using our brain and mind. Thus, we need to better align the above algorithms with the algorithms our mind uses to align more with the natural law, cosmic algorithms, modules, and programming that originates in the unified field.

References

[1] T. Nader, “Consciousness Is All There Is: A Mathematical Approach with Applications,” International Journal of Mathematics and Consciousness, Vol. 1, 2015.

[2] T. Nader, “Consciousness is Primary—Illuminating the Leading Age of Knowledge,” Proceedings of the 2012 Faculty Symposium at Maharishi University of Management.
[3] T. Routt, *Quantum Computing, the Vedic Fabric of the Digital Universe*, 1st World Pub, 2005, Fairfield, Iowa.
[4] E. Khan, “Next generation web—intelligent search, question answering, summarization and more,” International Journal of Computers and Communications, Vol. 9, June 2015.
[5] E. Khan, “Natural Language Processing, Big Data, Bioinformatics and Biology,” International Journal of Biology and Biomedical Engineering, June 2014.
[6] E. Khan, “Processing Big Data with Natural Semantics and Natural Language Understanding using Brain-Like Approach,” International Journal of Computers and Communications, January 2014.
[7] E. Khan, “Intelligent Internet: Natural Language and Question and Answer based Interaction,” International Journal Of Computers And Communications, Oct. 2013.
[8] Wikipedia, “Big Data,” `http://en.wikipedia.org/wiki/Big\_data`
[9] E. Khan, “Natural Language based Human Computer Interaction: a Necessity for Mobile Devices,” International Journal Of Computers And Communications, Dec. 2012.
[10] E. Khan, E. Aleisa, “Internet Access by Talking and Listening Using any Phone: Help Global Development by Effectively Bridging the Digital and Language Divides,” International Journal of Information Studies, Vol. 4, No 1, Jan. 2012.
[11] E. Khan, “Education for the Base of the Pyramid People (BOP) using Voice Internet e-Learning,” Procedia—Social and Behavioral Sciences Journal, Nov. 2012.
[12] C. Eaton et al., “Understanding Big Data: Analytics for enterprise class Hadoop and Streaming Data,” `http://public.dhe.ibm.com/common/ssi/ecm/en/iml14296usen/IML14296USEN.PDF`
[13] T. White, “Hadoop: The Definitive Guide,” O’Reilly Media, p. 3.
[14] Link “Latest Big Data Activities,” `http://data-informed.com/big-data-news/`
[15] E. Khan, “Natural Language based Human Computer Interaction: a Necessity for Mobile Devices,” International Journal Of Computers And Communications, Dec. 2012.
[16] E. Khan, *Internet for Everyone—Reshaping the Global Economy by Bridging the Digital Divide*, 2011.
[17] E. Khan, “Reforming Education for Economic, Social and Other Developments,” 2014 International Conference on Education Reform and Modern Management (ERMM2014), Thailand, Jan. 10-12, 2014.
[18] Link “Hunger Statistics”—`https://www.wfp.org/hunger/stats`
[19] `www.npic.orst.edu/factsheets/ddtgen.pdf`
[20] `https://en.wikipedia.org/wiki/Blue\_Brain\_Project`
[21] J. Stiglitz, *Making Globalization Work*, W. W. Norton and Company, 2006.
[22] Craig Pearson, *The Supreme Awakening: Experiences of Enlightenment Throughout Time and How You Can Cultivate Them.* Maharishi University of Management Press, Fairfield, Iowa, 2013.
[23] Claude Yves Charron, “From Digital Divide to Digital Opportunities,” Orbicom, 2005—`http://www.itu.int/ITU-D/ict/publications/dd/material/index\_ict\_opp.pdf`
[24] World crime trends and emerging issues and responses in the field of crime prevention and criminal justice, Commission on Crime Prevention and Criminal Justice Twenty-third session, Vienna, April 12–16, 2014.
[25] C. K. Prahlad et al., “The Fortune at the Bottom of the Pyramid,” strategy+business, issue 26, first quarter 2002.
[26] Bevan Morirs, “Consciousness-Based Education, introduced in 1971 by Maharishi Mahesh Yogi” `dl.globalgoodnews.com/pdf/consciousness-Based-Education.pdf`
[27] MIU Faculty, “Consciousness-Based Education: A Foundation for Teaching and Learning in the Academic Disciplines (12-Volume Series)” `https://www.mum.edu/whats-happening/books/consciousness-based-education-book-series/`
[28] Maharishi Mahesh Yogi, *Life supported by natural law: discovery of the Unified Field of all the laws of nature and the Maharishi Technology of the Unified Field—Integration of modern science and ancient Vedic science to perpetuate the rising Age of Enlightenment and bring fulfillment to every area of society: lectures.* Age of Enlightenment Press, 1986. [29] Maharishi

Mahesh Yogi, *Vedic Knowledge for Everyone: An Introduction*. Age of Enlightenment Press, 1995.
[30] David Lynch, *Catching the Big Fish: Meditation, Consciousness, and Creativity*, 2007.
[31] Cosmic Computer, `http://www.huffingtonpost.com/ann-purcell/unplug-and-recharge\_b\_4590865.html`
[32] John Hagelin, "Restructuring physics from its foundation in light of Maharishi's Vedic Science," Modern Science and Vedic Science 3, 1989.
[33] International Journal of Psychophysiology 71 (2009); 170-176.
[34] American Journal of Hypertension 21 (2008): 310-316.
[35] Archives of International Medicine 166 (2006): 1218-1224.
[37] E. Khan, "Machine Learning Algorithms for Natural Language semantics and Cognitive Computing," The 2016 International Conference on Computational Science and Computational Intelligence (CSCI'16), Dec. 15-17, 2016, Las Vegas, USA.
[38] E. Khan, "Natural Language Computing: A Necessity for Cognitive Computing and Intelligent Internet," Plenary Talk at Artificial Intelligence Conference in Uni. Of Cambridge, UK (AIKED'17) in Feb. 2017.
[39] `http://www.alternet.org/story/147318/100,000\_americans\_die\_each\_year\_from\_prescription\_drugs,\_while\_pharma\_companies\_get\_rich`
[40] Scientific Research on the Transcendental Meditation Program, Maharishi University of Management, 2012.
[41] C. Pearson et al., "Consciousness-Based Education: A Foundation for Teaching and Learning in the Academic Disciplines," Consciousness-Based Books, an Imprint of Maharishi University of Management Press, Fairfield, Iowa.
[42] Basic NLP Steps `shttps://en.wikipedia.org/wiki/Copenhagen`
[43] E. Khan, "Lifelong Machine Learning with Logic, Semantics and Natural Language Processing," International Conference on Artificial Intelligence (ICAI 2019), Las Vegas, USA.
[44] D. Jurafsky, *Speech and Language Processing*, Pearson Prentice Hall 2009.
[45] K. Levi and P. Corazza, editors, "Consciousness Based Education: A Foundation for Teaching and Learning in the Academic Disciplines," Consciousness-Based Books, Maharishi University of Management Press, Vol. 10, 2011.
[46] Maharishi Mahesh Yogi, *Maharishi's Absolute Theory of Defence*. Age of Enlightenment Publications, 1996.
[47] Mikolov, T. et al. "Distributed Representations of Words and Phrases and their Compositionality." `https://arxiv.org/abs/1310.4546`.
[48] R. Horev. "BERT Explained: State of the art language model for NLP." `https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270`.
[50] E. Khan, Neural Networks Apparatus and Method for Pattern Recognition, US Patent # 5058180, Oct. 15, 1991.
[51] E. Khan, Neural Network Based Fuzzy Logic Systems and Methods, US Patent # 8828812, Oct. 27, 1998.
[52] E. Khan, "NeuFuz: An Intelligence Combination of Fuzzy Logic, with Neural Nets," IEEE, IJCNN, 1992, Japan.
[53] E. Khan, "Recurrent fuzzy logic using neural network", *World Wisepersons Workshop WWW 1994: Advances in Fuzzy Logic, Neural Networks and Genetic Algorithms* pp. 48–55
[54] J. Hagelin, "The Lagrangian of the unified field." In Maharishi Technology of the unified field: Results of Scientific Research. Washington D.C.: Age of Enlightenment Press, 1984.
[55] J. Hagelin, "Is consciousness the unified field? A field theorist's perspective." Modern Science and Vedic Science, 1987.
[56] J. Hagelin, "Special talk at MIU on unified field, quantum mechanics and Vedic Science," May 2020.
[57] TM Research—Scientific Research on Transcendental Meditation and TM-Sidhi Programme. `researchtm.net`.
[58] C. Mead, *Analog VLSI and Neural Systems*, ISBN 9780201059922, Jan. 1989.
[59] E. Khan, "Intelligent Internet: Natural Language and Question & Answer Based Interaction," International Journal of Computers and Communications, Oct. 2013.

[60] E. Khan, "Machine Learning Algorithms for Natural Language Semantics and Cognitive Computing." The 2016 International Conference on Computational Science and Computational Intelligence (CSCI'16), Dec. 15–17, 2016, Las Vegas, USA.
[61] *Maharishi International University Catalog*, MIU Press, 1974.

Appendix A. Brief Literature Review on Semantics and Knowledge Representation

Semantics, the meaning of words and sentences, is key for most NLP applications including question answering (QA), summarization, drawing inference, language generation, information extraction, and text analytics. Text analytics turns the unstructured information embedded in texts into structured data. The same applies for logic and cognitive computing. Thus, a complete literature review for all these is beyond the scope here. We rather would like to cover all these in a simplified manner yet including all the key points.

A good start for our proposed literature review will be information retrieval (IR) which can be defined as finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers). IR needs to deal with semantics to make the retrieved results satisfactory. However, the literature on IR itself is very rich and hence cannot be covered fully in depth here, so we have tried to provide a good summary starting from early 2000 to mid-2019. It provides a good review and guide to what is needed to help solve key NLP problems.

Retrieved information from all sources for a query must be evaluated to determine relevance with a degree. Content with highest relevance(s) would be provided as the most desired result. This is very good and logical. However, this general approach covered by many researchers has the two following issues:

(1) Contents must be retrieved first.
(2) Relevance must be determined.

The brief description provided below addresses these. It is important to note that the fundamental problem of calculating relevance is language independent, although some language specific features can refine and improve the results.

There are various solutions for item 1. Most of the existing solutions are based on a kind of extreme version of compositional semantics (semantics is considered for a complete sentence—refer to section A.1 below) in which the meaning of a document resides solely in the set of words it contains [44]. The ordering and constituency of the words that make up the sentences that make up the documents play *no* role in determining their meaning (semantics). Because they ignore syntactic information, these approaches are often referred to as bag-of-words models. Such models use term frequency (TF) and inverse document frequency (IDF). IDF uses the number $N/n(i)$ where N is the total number of documents and $n(i)$ is the number n of documents in which term (word) i occurred. Because of the large number of documents, IDF is usually squashed with a log function. In summary, documents are retrieved based on TF $\times$ IDF values that match with the words in the requesting sentence, the input.

While bag-of-words based TF-IDF models provide good results for some applications, their capabilities are limited due to not considering semantics and not considering contexts. Some recent advances have addressed these issues, especially the context and "context driven semantics" parts by using Word2Vec [47], Doc2Vec [47], and BERT [48]. These approaches use the previous words of a particular word W and the following words of word W, to define the context and to represent W as improved bag-of-words. Such approaches have produced much improved results for some applications [49]. However, since mainly context is used (and not the real meaning of the word W), such approaches are not adequate for many complex applications. So to retrieve good information, we still need good relevance. However, since reliable relevance still needs to be done (which still remains an open problem), it is logical that we mainly focus on the description of item 2. This will address the relevance needed in item 1 as well.

Many researchers have proposed various solutions to calculate relevance. Early solutions can be grouped as solutions that are based on key word match in a few paragraphs (similar to the idea that is described above). Although this can provide good results in some cases, good relevance cannot really be calculated by using key words. The relationship between words and semantic meanings are keys to determine the relevance. And in doing so, "knowledge representation" has become a key issue as it is strongly related to semantics—that is, what is real knowledge and how can it be represented? So our remaining literature review focuses on semantics and knowledge representation.

A.1. **Relevance of extracted information from a semantics and knowledge standpoint.** Semantics has multiple aspects, mainly representation, analysis, and computation. The state-of-the-art approach for semantics is based on a model-theoretic approach that defines a model to represent objects (like a word or sentence), properties of objects, and relationships among objects.

First-order logic is considered as a state-of-the-art approach for semantic representation and computation. Another main approach is description logic which has two parts—frames and semantic networks (for example, semantic web) that uses ontology. However, these are usually considered as part of first-order logic [44]. In a similar way—relational knowledge (for example, knowledge in a database table), inferential knowledge (propositional or predicate logic), inheritable knowledge (for example, is-a-relationship) and procedural knowledge (if-then rules) can be considered as part of first-order logic.

The computational aspect of semantics is called computational semantics when semantics is considered for a complete sentence. Word level semantics (called lexical semantics) is also important and includes word senses (for example, a financial bank versus a blood bank versus a river bank), relations between senses, and the like. Computational lexical semantics computes word sense disambiguation, word similarity, and semantic role labeling using ML and statistical methods.

As mentioned, although numerous works have been done for item 2, it still remains an open problem. To continue with relevance (item 2), we are assuming that some important information is retrieved somehow. One approach proposed a solution by taking a small sample from all the retrieved results, thus saving time by not retrieving the full content before evaluation is completed, and determining relevance. However, it does not provide details of how relevance is computed.

Another approach describes that while current approaches to ontology mapping produce good results by mainly relying on label and structure based similarity measures, there are several cases in which they fail to discover important mappings. This describes a new approach to ontology mapping by exploiting the increasing amount of semantic resources available online. As a result, there is no need either for a manually selected reference ontology—the relevant ontologies are dynamically selected from an online ontology repository—or for transforming background knowledge in an ontological form.

Yet another approach discusses a number of important issues that drive knowledge representation research. It begins by considering the relationship between knowledge and the world and the use of knowledge by reasoning agents (both biological and mechanical) and concludes that a knowledge representation system must support activities of perception, learning, and planning to act. An argument is made that the mechanisms of traditional formal logic, while important to our understanding of mechanical reasoning, are not by themselves sufficient to solve all of the associated problems. In particular, notational aspects of a knowledge representation system are important, both for computational and conceptual reasons. Two such aspects are distinguished: expressive adequacy and notational efficacy. Researchers also discuss the structure of conceptual representations and argues that taxonomic classification structures can advance both expressive adequacy and notational efficacy. They predict that such techniques will eventually be applicable throughout computer science and that their application can produce a new style of programming, more oriented toward specifying the desired behavior in conceptual terms. Such "taxonomic programming" can have advantages for flexibility, extensibility, and maintainability, as well as for documentation and user education.

Another approach mentions that although knowledge representation is one of the central and, in some ways, most familiar concepts in AI, the most fundamental question—What is it?—has rarely been answered directly. Numerous papers have lobbied for one or another variety of representation, other papers have argued for various properties a representation should have, and still others have focused on properties that are important to the notion of representation in general. An approach that addresses the question directly proposes that the answer can best be understood in terms of five important and distinctly different roles that a representation plays, each of which places different and, at times, conflicting demands on the properties a representation should have.

Another approach explains that knowledge is far more complex than propositions. Semantics, relations, and various other unquantifiable material make up knowledge. This approach mentions that conceptual graphs are equivalent to predicate calculus and emphasizes that knowledge representation needs to be mapped to today's database.

A further approach presents a new approach to knowledge representation where knowledge bases are characterized not in terms of the structures they use to represent knowledge, but functionally, in terms of what they can be asked or told about some domain. It starts with a representation system that can be asked questions and told information in a full first-order logical language. It then defines ask-and-tell operations over an extended language that can refer not only to the domain but also to what the knowledge base knows about that domain. The major technical result claimed is that the resulting knowledge, which now includes auto-epistemic aspects, can still be represented symbolically in first-order terms. The overall result is a formal foundation for knowledge representation which, in accordance with current principles of software design, cleanly separates functionality from implementation structure.

Another approach argues the proposal by Levesque and Brachman that general-purpose knowledge representation systems should restrict their languages by omitting constructs which require nonpolynomial worst-case response times for sound and complete classification. Levesque and Brachman also separate terminological and assertional knowledge, and restrict classification to purely terminological information. This approach demonstrates that restricting the terminological language and classifier in these ways limits these "general-purpose" facilities so severely that they are no longer generally applicable. This approach argues that logical soundness, completeness, and worst-case complexity are inadequate measures for evaluating the utility of representation services and that this evaluation should employ the broader notions of utility and rationality found in decision theory.

This approach also suggests that general-purpose representation services should provide fully expressive languages, classification over relevant contingent information, "approximate" forms of classification involving defaults, and rational management of inference tools.

Another approach proposes that the Internet poses challenges to knowledge representation systems that fundamentally change the way we should design KR languages. They describe the simple HTML ontology extensions (SHOE), a KR language which allows web pages to be annotated with semantics. It also describes some generic tools for using the language and demonstrates its capabilities by describing two prototype systems that use it.

Yet another approach expresses web page content in a format that machines can understand—the semantic web—which provides huge possibilities for the Internet and for machine reasoning. Unfortunately, there is a considerable distance between the present-day Internet and the semantic web of the future. The process of annotating the Internet to make it semantic web-ready is quite long and not without resistance. One mechanism for semanticizing the Internet is known as AutoSHOE, and it is capable of categorizing pages according to one of the present HTML semantic representations (simple HTML ontology extensions) by Heflin et al. We are also extending this system to other semantic web representations, such as the Resource Description Framework (RDF). The AutoSHOE system includes mechanisms to train classifiers to identify web pages that belong in an ontology, as well as methods to classify pages within an ontology and to learn relations between pages with respect to an ontology. The modular design of AutoSHOE allows for the addition of new ontologies as well as algorithms for feature extraction, classifier learning, and rule learning.

This system has the promise to help transparently bridge traditional web technology to the semantic web using contemporary machine learning techniques rather than tedious manual annotation.

Another approach reviews the use of ontologies for the integration of heterogeneous information sources. Based on an in-depth evaluation of existing approaches to this problem, they discuss how ontologies are used to support the integration task. They also ask for ontology engineering methods and tools used to develop ontologies for information integration. They also mention all key issues with ontology integration and associated tools.

Another approach by Wolters Kluwer Italy, part of the Wolters Kluwer group, announced Cogito, which uses expanded search words using synonym words for better advanced search results. This is basically an extension of existing advanced search by using richer synonyms.

This semantics method is also proposed for the Internet page content (the semantic web) in a format that allows machines to understand the web content. The semantic web provides huge possibilities for the Internet and for machine reasoning. Unfortunately, there is a considerable distance between the present-day *World Wide Web* and the semantic web of the future. Semanticizing the web approach allows for the addition of new ontologies as well as algorithms for feature extraction, classifier learning, and rule learning. This system has the promise to help transparently bridge traditional web technology to the semantic web using contemporary machine learning techniques rather than tedious manual annotation.

Another approach describes a semantic reasoner, reasoning engine, rules engine, or simply a reasoner, which is a piece of software able to infer logical consequences from a set of asserted facts or axioms. The notion of a semantic reasoner generalizes that of an inference engine by providing a richer set of mechanisms to work with. The inference rules are commonly specified by means of an ontology language and often a description language. Many reasoners use first-order predicate logic to perform reasoning; inference commonly proceeds by forward chaining and backward chaining. There are also examples of probabilistic reasoners, including Pei Wang's non-axiomatic reasoning system, Novamente's probabilistic logic network, and Pronto, a probabilistic description logic reasoner.

A semantic engine extracts the meaning of a document to organize it as partially structured knowledge. For example, you can submit a batch of news stories to a semantic engine and get back a tree categorization according to the subjects they deal with.

Current semantic engines can typically:

- Categorize documents (Is this document written in English, Spanish, Mandarin? Is this an article that should be filed under the Business, Lifestyle, or Technology categories?)
- Suggest meaningful tags from a controlled taxonomy and assert their relative importance with respect to the text content of the document
- Find related documents in the local database or on the web
- Extract and recognize mentions of known entities such as famous people, organizations, places, books, and movies and link the document to its knowledge base entries (like a biography for a famous person)
- Detect yet unknown entities of the same aforementioned types to enrich the knowledge base

- Extract knowledge assertions that are present in the text to fill up a knowledge base along with a reference to trace the origin of the assertion. Examples of such assertions could be the fact that one company is buying another along with the amount of the transaction, the release date of a movie, and the new club of a football player.

So we see that it basically uses structured information, not natural sentences.

From this short review, it is clear that calculating relevance is dependent on semantics, which is dependent on knowledge representation. Among various approaches of knowledge representation, the ontology-based approach appears to be more widely used, especially when we are talking about the Internet. As discussed by various authors, the key issues with the ontology-based approach are:

(1) Developing ontologies
(2) Mapping ontologies
(3) Integrating various ontologies
(4) Developing associated tools
(5) Automating ontology development using machine learning techniques
(6) The "mechanical" nature of the semantic-by-ontology representation

And, of course, manual and semi-automated development of ontologies for over 3 billion websites on the Internet is impractical. Such an approach will be good for certain web applications. Besides, "mechanical semantic" and "mechanical reasoning" will significantly limit the calculation of relevance.

The other key technique for relevance is statistical techniques including maximum likelihood. While this approach provides excellent results for predicting the next word(s) when words are typed in the search field in a search engine, it is not effective in calculating semantics. Thus, the best existing methods are limited by "mechanical semantics" and its scalability. This affects almost all applications of NLU including information retrieval, Q&A, summarization, language translation, and conversational systems.

Appendix B. More details on Major Steps in Natural Language Processing

Natural Language Processing has multiple stages—morphology, tokenization, part-of-speech (POS) tagging, name entity recognition (NER), parsing, semantics, information retrieval, information extraction, making recommendations, question answering, summarization, drawing inference, and the like. A typical NLP pipeline is shown below along with a short description of all stages. See Figure 5.

Step-by-Step

Consider the following description of the city of Copenhagen:

> Copenhagen is the capital and most populous city of Denmark and sits on the coastal islands of Zealand and Amager. It's linked to Malmo in southern Sweden by the Oresund Bridge. Indre By, the city's historic center, contains Frederiksstaden, an 18th-century rococo district, home to the royal family's Amalienborg Palace.

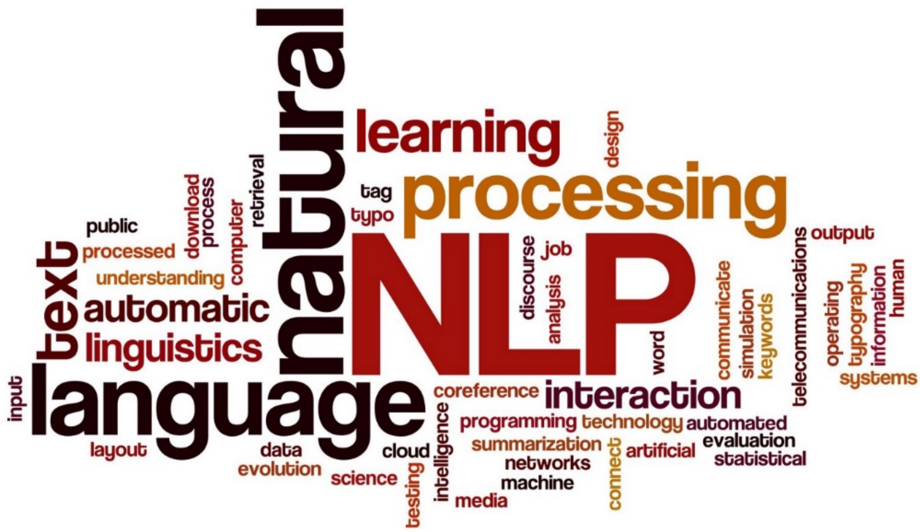


FIGURE 5. Natural Language Processing word cloud.

> Nearby is Christiansborg Palace and the Renaissance-era Rosenborg Castle, surrounded by gardens and home to the crown jewels. [42]

This paragraph contains several useful facts. It would be great if a computer could read this text and understand that Copenhagen is a city, Copenhagen is located on coastal islands, and the royal family's Amalienborg Palace is in Copenhagen. But to get there, we have to first teach our computer the most basic concepts of written language and move up from there. See Figure 6.

Step 1: Sentence Segmentation: The first step in the pipeline is to break the text apart into separate sentences. That gives us this:
(1) "Copenhagen is the capital and most populous city of Denmark."
(2) "Copenhagen sits on the coastal islands of Zealand and Amager."
(3) "Copenhagen is linked to Malmo in southern Sweden by the Oresund Bridge."

We can assume that each sentence in English is a separate thought or idea. It will be easier to write a program to understand a single sentence than to understand a whole paragraph. Coding a sentence segmentation model can be as simple as splitting apart sentences whenever you see a punctuation mark. But modern NLP pipelines often use more complex techniques that work even when a document isn't formatted cleanly.

Step 2: Word Tokenization: Now that we've split our document into sentences, we can process them one at a time. Let's start with the first sentence from our document: "Copenhagen is the capital and most populous city of

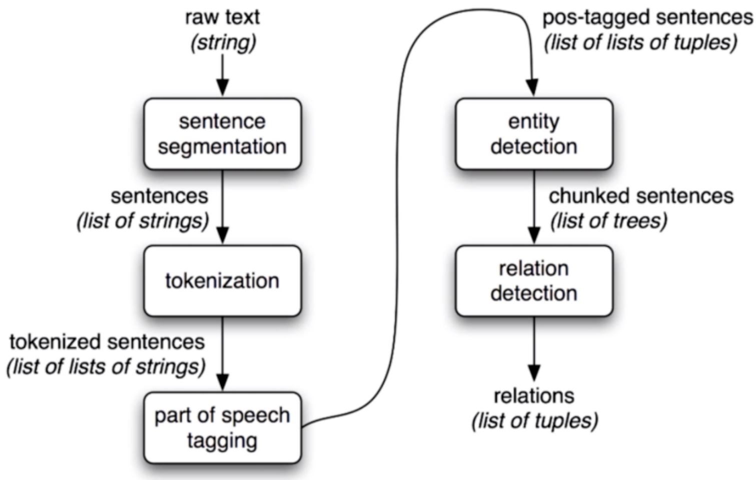


FIGURE 6. Neural Network.

Denmark and the coastal islands." The next step in our pipeline is to break this sentence into separate words or tokens. This is called tokenization. This is the result: "Copenhagen", "is", "the", "capital", "and", "most", "populous", "city", "of", "Denmark", "and", "the", "coastal", "islands", and ".".

Tokenization is easy to do in English. We just split apart words whenever there's a space between them. We also treat punctuation marks as separate tokens since punctuation also has meaning.

Step 3: Predicting Parts of Speech for Each Token: Next, we look at each token and try to guess its part of speech, whether it is a noun, a verb, an adjective, and so on. Knowing the role of each word in the sentence will help us start to figure out what the sentence is talking about. We can do this by feeding each word (and some extra words around it for context) into a pre-trained part-of-speech classification model, as shown in Figure 7.

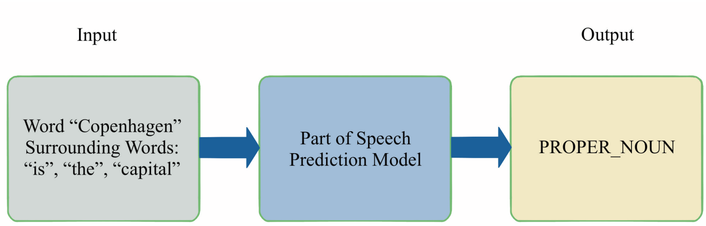


FIGURE 7. Example of art-of-speech classification of the word "Copenhagen".

The part-of-speech model was originally trained by feeding it millions of English sentences, with each word's part of speech already tagged and then having it learn to replicate that behavior. Keep in mind that the model is completely based on statistics—it doesn't actually understand what the words mean in the same way that humans do. It just knows how to guess a part of speech based on similar sentences and words it has seen before. After processing the whole sentence, we get the result shown in Figure 8.

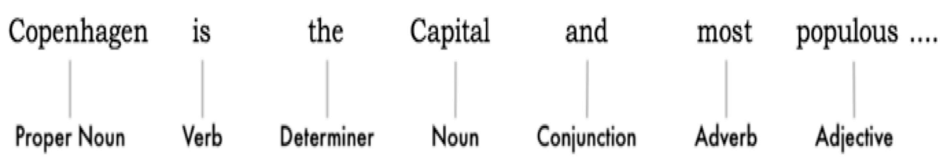


FIGURE 8. Result of processing the sentence "Copenhagen is the capital and most populous . . . "

With this information, we can already start to glean some very basic meaning. For example, we can see that the nouns in the sentence include "Copenhagen" and "capital," so the sentence is probably talking about Copenhagen.

Step 4: Text Lemmatization: In Danish (and most languages), words appear in different forms. Look at these two sentences:

I had a MacBook.

I had two MacBooks.

Both sentences talk about the noun "MacBook", but they are using different inflections. When working with text in a computer, it is helpful to know the base form of each word so that you know that both sentences are talking about the same concept. Otherwise, the strings "MacBook" and "MacBooks" look like two totally different words to a computer. In NLP, we call finding this process lemmatization—figuring out the most basic form or lemma of each word in the sentence.

The same thing applies to verbs. We can also lemmatize verbs by finding their root, unconjugated form. So "I had two MacBooks" becomes "I [have] two [MacBook]." Lemmatization is typically done by having a look-up table of the lemma forms of words based on their part of speech and possibly having some custom rules to handle words that you've never seen before. Figure 9 shows what our sentence looks like after lemmatization adds in the root form of our verb.

The only change we made was turning "is" into "be."

Step 5: Identifying Stop Words: Next, we want to consider the importance of each word in the sentence. English has a lot of filler words that appear very frequently like "and," "the," and "a." When doing statistics on text, these words introduce a lot of noise since they appear much more frequently than other words. Some NLP pipelines will flag them as stop words—that is, words that you might want to filter out before doing any

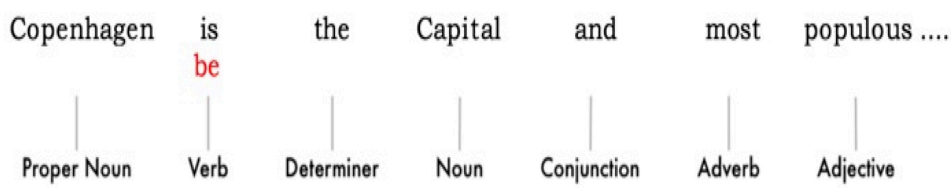


FIGURE 9. Neural Network

statistical analysis. How our sentence looks with the stop words grayed out is shown in Figure 10.

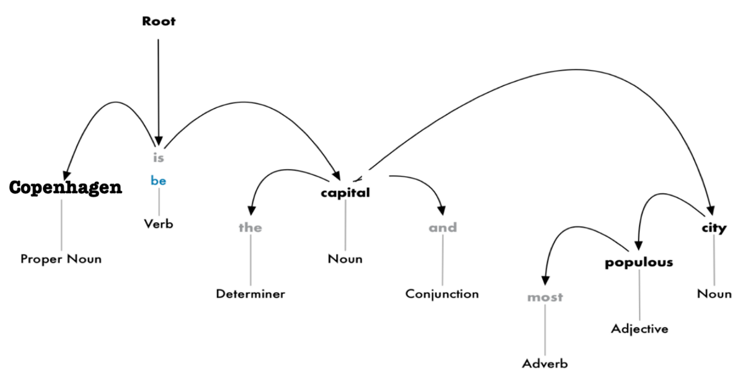


FIGURE 10. Sentence with stop words in gray.

Stop words are usually identified by checking a hardcoded list of known stop words. But there's no standard list of stop words that is appropriate for all applications. The list of words to ignore can vary depending on your application. For example if you are building a rock band search engine, you want to make sure you don't ignore the word "The"because not only does the word "The" appear in a lot of band names, there's a famous 1980s rock band called The The!

Step 6: Dependency Parsing: The next step, dependency parsing, is to figure out how all the words in our sentence relate to each other. The goal is to build a tree that assigns a single parent word to each word in the sentence. The root of the tree will be the main verb in the sentence. The beginning of the parse tree for our sentence was shown in Figure 10.

But we can go one step further. In addition to identifying the parent word of each word, we can also predict the type of relationship that exists between those two words as shown in Figure 11.

This parse tree shows us that the subject of the sentence is the noun "Copenhagen" and it has a "be" relationship with "capital." We finally know something useful—Copenhagen is the capital! And if we followed the

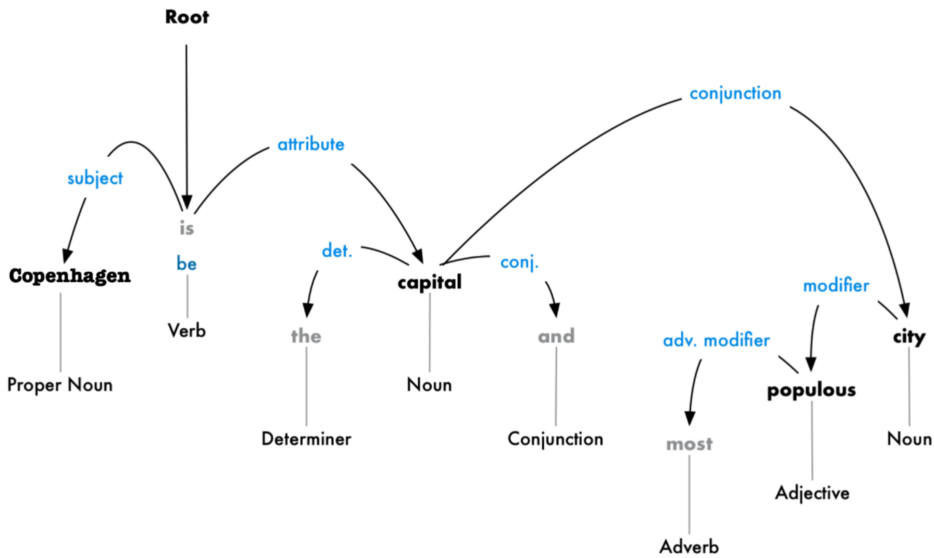


Figure 11. Dependency parsing of the sentence "Copenhagen is the capital and most populous city".

complete parse tree for the sentence (beyond what is shown), we would even find out that Copenhagen is the capital of Denmark. Just as we predicted parts of speech earlier using a machine learning model, dependency parsing also works by feeding words into a machine learning model and outputting a result. But parsing word dependencies is a particularly complex task and would require an entire article to explain in any detail. If you are curious about how it works, a good place to start is Matthew Honnibal's excellent article "Parsing English in 500 Lines of Python."

Step 7: Named Entity Recognition (NER): Now that we have done all that hard work, we can finally move beyond grade-school grammar and start actually extracting ideas. In our sentence, we have the following nouns, "Copenhagen", "capital", "city", "Denmark", and "coastal islands" shown in blue in Figure 12.

Figure 12. Sentence with nouns in blue

Some of these nouns represent real things in the world. For example, "Copenhagen," "Denmark", and "coastal islands" represent physical places on a map. It would be nice to be able to detect that! With that information, we could automatically extract a list of real-world places mentioned in a document using NLP. The goal of named entity recognition, or NER, is to detect and label these nouns with the real-world concepts that they represent. Our sentence after running each token through our NER tagging model is shown in Figure 13.

FIGURE 13. Sentence with nouns in blue

But NER systems aren't just doing a simple dictionary lookup. Instead, they are using the context of how a word appears in the sentence and a statistical model to guess which type of noun a word represents. A good NER system can tell the difference between "Brooklyn Decker" the person and the place "Brooklyn" using context clues. Here are just some of the kinds of objects that a typical NER system can tag:

- People's names
- Company names
- Geographic locations (both physical and political)
- Product names
- Dates and times
- Amounts of money
- Names of events

NER has many uses since it makes it so easy to grab structured data out of text. It's one of the easiest ways to quickly get value out of an NLP pipeline.

The above diagrams show the basic stages of NLP. The algorithms to handle such stages are quite good, especially considering the use of machine learning in some complex stages like NER. These provide a good level of information to a computer and suggest simple questions like

- "What is the most populous city in Denmark?" This can be answered using some logic and associate coding using the relevant sentences and NER. For example, Copenhagen (NER) is related to the most populous city in the first sentence.

However, more complex questions like

- "How is Copenhagen linked to Malamo?" This question is hard to answer—it can be done if we write more logic with more parsing (for example, we need to define what "it" means in the second sentence and how that is related to Copenhagen); for humans it (called universe of discourse—that is, the meaning of Copenhagen is connected to the second sentence by "it") is very simple. But for computers, it is hard unless we spell out everything for the computer. The reason we need to write more logic to compute an answer is that we would need to define the semantics or meaning of each word and then apply some logic on those to see their relations.

However, for more complex questions, defining semantics for each word and each sentence and also writing complex logic becomes very difficult. Besides, if there is a change to some words, the logic may fail to find an answer (See Section 4 for details). Also, we continue to learn as we grow. We develop an important thing called common sense that can help us derive new facts and, new sentences (sentence generation). This is one of the key reasons why NLP does not follow a traditional process in solving, say, a mathematical type problem.

Department of Computer Science, Maharishi International University, Fairfield, IA

International Journal of Mathematics and Consciousness

In recent centuries, scientists have found that many phenomena in nature obey physical laws that can be expressed precisely in the language of mathematics. Their successes have led scientific inquiry beyond the physical world to include what was previously considered metaphysical or non-material. Today, an increasing number of scientists are examining the nature of consciousness and its relationship to the human brain.

While most models of consciousness propose that it is a product of chemical and electrical behavior within the brain, no current theory resolves the so-called "hard problem of consciousness"—how physical processes in the nervous system give rise to subjective experiences such as experiencing, thinking, feeling, analyzing, and creating. At the same time, it is undeniable that without awareness—without consciousness—we cannot think, perceive, dream, or love. On this basis alone, a scientific journal dedicated to exploring the nature of consciousness is timely and appropriate.

While consciousness can be studied within a variety of disciplines, mathematics especially lends itself to examine the relationship between consciousness and physical phenomena. Mathematics is precise and rigorous in its methodology, giving symbolic expression to abstract patterns and relationships. Although developed subjectively, using intuition along with the intellect and logical reasoning, mathematics allows us to make sense of our outer physical universe. Mathematics is the most scientific representation of subjective human intelligence and thought, formalizing how individual human awareness perceives, discriminates, organizes, and expresses itself.

The scientific consideration of consciousness by itself is a formidable challenge, for consciousness is a purely abstract reality. But the study of what we might call "consciousness at work"— how consciousness expresses itself in our daily activity of thinking, analyzing, creating, theorizing, and feeling—is inherently more accessible. For this exploration also, mathematics is the ideal tool, because its ability to express patterns of abstract human awareness helps us make sense of our physical universe. One could in fact argue that mathematics is the most scientifically reliable tool for the exploration of the dynamics of consciousness, for it alone can be seen as the symbolic representation of "consciousness at work."

The International Journal of Mathematics and Consciousness will help to fulfill the need for a forum of research and discussion of consciousness and its expressions. The editors invite mathematicians, scientists, and other thinkers to present their theories of consciousness without restriction to proposed axioms and postulates, with the stipulation only that such theories follow strict logical argumentation and respect proven facts and observations. Articles that use factual or logical counterarguments to challenge commonly believed but not fully established facts and observations are also welcome.