Redefining Natural Language: From $\aleph_0$ to $2^{\aleph_0}$

Ramon Padilla-Reyes
Charles Woodrum

February 18, 2024

Abstract

We argue that language should organically permit expressions of countably infinite length, and thus that the size of our language is $2^{\aleph_0}$. Abstract formalism of language have brought much technical understanding to the fields of syntax, semantics and pragmatics. In this paper, we explore, through an abundance of proofs, the size of all possible human utterances. Be it a specific language or all languages. We will show that the set of all possible sentences has size $\aleph_0$ unless either the length of the sentence can be infinite or the set of all possible words is uncountably infinite. This is true for even the most permissive grammatical rules and the most expansive vocabulary possible. The fact that our arguably uncountable world must be described by a countable system of language may have interesting implications for philosophy and metaphysics, but that will not be addressed here. This work aims to be a definitive exploration of the possible cardinalities of language, depending on the size of utterances allowed by the axioms of the language.

1 Introduction

The study of language has intrigued scholars across various disciplines for centuries. Linguistics, philosophy, and mathematics have each contributed unique perspectives. Yet the intersection of these fields presents untapped potential for groundbreaking discoveries. This paper delves into one such intersection: the mathematical cardinality of language. Traditionally, language has been viewed through the lens of countability, bound by the finiteness of grammar and vocabulary. However, this exploration challenges the existing paradigm, proposing a novel perspective that considers language as potentially uncountable under specific circumstances.

The traditional view of language is that it is a set with cardinality $\aleph_0$ (CITATIONS NEEDED). This standpoint has aligned well with the constraints of finite grammatical structures and a countable vocabulary. This perspective, is effective in its consistency with observed linguistic patterns. However, it may not encapsulate the true breadth and adaptability of natural language. To explore this, we divide this paper into two sections. In section I, we introduce scenarios where language transcends these assumed boundaries—particularly. Then in section II, we will pay close attention when considering sentences of infinite length and an uncountably infinite vocabulary. Under these conditions, we hypothesize that the size of language expands dramatically, reaching the dimensions of $2^{\aleph_0}$.

Our argument is supported by a series of mathematical proofs, each carefully constructed to test the limits of language's cardinality. These proofs are not mere academic exercises; they represent a fundamental shift in understanding language's capacity to represent and articulate the human experience. In redefining the bounds of language, we open new avenues for linguistic theory, computational linguistics, and even philosophy. The implications of a language system that can accommodate expressions of infinite complexity are profound, suggesting new ways of conceptualizing thought, communication, and information processing.

By rigorously exploring these theoretical possibilities, this paper aims to not only redefine our understanding of natural language's size but also to stimulate further interdisciplinary exploration at the nexus of linguistics, mathematics, and philosophy. The journey $\aleph_0$ to $2^{\aleph_0}$ is not just a mathematical transition—it's a conceptual leap towards appreciating the unbounded potential of language as a tool for human expression and understanding.

2 Section I

Note before we begin that sentence length can refer to the number of characters or the number of words. In this paper, the term "sentence length" will refer to the number of words in a sentence unless otherwise noted. When we wish to refer to a sentence's character length, we will use the term "sentence character length".

Vocabulary Size	**Sentence Word Length**	**Size of Language**
Finite	Finite	$\aleph_0$
$\aleph_0$	Finite	$\aleph_0$
$2^{\aleph_0}$	Finite	$2^{\aleph_0}$
Finite	$\aleph_0$	$2^{\aleph_0}$
$\aleph_0$	$\aleph_0$	$2^{\aleph_0}$
$2^{\aleph_0}$	$\aleph_0$	$2^{\aleph_0}$

Table 1: A table of the possible vocabulary and sentence length, and the resulting size of natural language. Each of the results will be proven separately. "Finite" refers to sentence word length or vocabulary size that is unbounded but finite, i.e. any finite length but not infinite length.

Theorem 2.1. *Grammatical sentences can have unbounded length.*

Proof. Consider just one obviously grammatical sentence *"The mother of the mother of the mother ... fell.* For all $k \in \mathbb{N}$, we can refer to the mother in the k^{th} generation with the sentence remaining grammatical. Therefore $\forall n \in \mathbb{N}$, we can produce a sentence whose length is greater than n, so grammatical sentence length is unbounded. Furthermore, the k^{th} generation can be put into one-to-one correspondence with the natural numbers, meaning that the cardinality of the set of all such sentences is $\aleph_0$. A grammar rule permitting only sentences of this form is taken to be the baseline "most restrictive" grammar rule in upcoming arguments, so that all realistic grammar rules can produce a greater abundance of sentences than this one. □

Theorem 2.2. *With an unbounded, finite vocabulary and an unbounded, finite sentence length, $|Natural\ Language| = |\mathbb{N}|$.*

Proof. Let the vocabulary of language be denoted V and let the size of V (the number of words available for sentence construction) be some $N \in \mathbb{N}$, so $|V| = N$. Let the set of all grammatical sentences be G. Each sentence must be finite, but there is no maximum or minimum sentence length. The grammar rule that results in the largest G is simply the rule that any finite combination of words is a grammatically correct sentence. In this case, it is possible to enumerate all possible sentences. There is one sentence of length 0, N of length 1, N^2 of length 2, and N^k possible sentences of length k for all $k \in \mathbb{N}$.

Let S_k be the set of all sentences of length k. $|S_0| = 1$ and $|S_k| = k^N$ $\forall k \in \mathbb{N}$. We note that

$$G = \bigcup_{k=0}^{\infty} S_k \tag{1}$$

and we see that since G is a countable union of finite sets, $|G| = |\mathbb{N}|$. If we let the set of all grammatical sentences according to some grammar rule r be denoted G_r, we must have $G_r \subset G$, which implies $|G_r| \leq |G|$. We know from Theorem 2.1 that even the most restrictive grammatical rules can admit countably many sentences, so $|G_r| \geq \mathbb{N}$, and thus we have $|Natural\ Language| = |\mathbb{N}| = \aleph_0$ □

The question might arise "What if we have an infinite vocabulary?". In the case where the vocabulary is uncountable, we can simply have a sentence of length 1, and the set of all such sentences is uncountable, and thus we have $|Natural\ Language| = 2^{\aleph_0}$.

However, the number of all constructed human words that are not numbers must be finite, since we could, in principle, list all of the words ever uttered or written by an human throughout all of human history and still generate a finite list. Adding the natural numbers, $\mathbb{N}$, to the list of human words creates a countably-infinite vocabulary, since all natural numbers find themselves in the list of human words, and the set of all natural numbers is obviously countably-infinite. Adding the rational numbers to human vocabulary has the same effect, since $\mathbb{Q}$ is also (less-obviously) a countably-infinite set. Adding the irrational numbers $\mathbb{I} = \mathbb{R}\backslash\mathbb{Q}$ to human vocabulary is only possible if we admit sentences of countably infinite length. Each irrational $x \in \mathbb{I}$ is a non-terminating, non-repeating decimal, and its full expression (counted as a single word) requires countably-infinitely many characters, which requires the admission of sentences of countably infinite character length but not necessarily infinite word length. Let us consider the case where we admit all human words and utterances and all natural numbers in our vocabulary, but still require that sentences have a finite, unbounded length.

Theorem 2.3. *With a countably infinite vocabulary and an unbounded, finite sentence length,* $|Natural\ Language| = |\mathbb{N}| = \aleph_0$.

Proof. Let the vocabulary from which we construct sentences be denoted V with $|V| = |\mathbb{N}|$. Again, we consider the set of all grammatical sentences G with the most permissive grammar rule. Just as before, we denote the set of all sentences of length k with S_k. Again $|S_0| = 1$. However, $|S_1| = |\mathbb{N}|$. Further $|S_2| = |S_1 \otimes S_1| = \aleph_0$, (the union of two countable sets is countable). From this, we see that $|S_k| = |\mathbb{N}|\ \forall\ k \in \mathbb{N}$. This means we again have

$$G = \bigcup_{k=0}^{\infty} S_k \tag{2}$$

This time, each S_k is of countable cardinality, and the countable union of countable sets is countable, so $|G| = \aleph_0$. As in Theorem 0.3, any G_r must have smaller cardinality than G, but, from Theorem 0.1, still must have countably many sentences, so $|G_r| \leq \aleph_0$. □

It is useful at this point to argue in favor of the inclusion of sentences that are not just finite but unbounded in length, but infinite in length. Consider the decimal expansion of $\pi = 3.14159...$ which is a non-repeating, non terminating decimal. One can express this number as a sentence saying *The first digit after 3 of the ratio between a circle's circumference and its diameter is 1, then 4, then 1, ...* Just as no finite representation of π can ever completely express the fullness of a truly infinite number, no finite sentence can fully explain the ratio between a circle's circumference and its diameter. Indeed, if human language is expected to be able to express fully the ideas created in the human mind, it must necessarily include all those ideas expressed in mathematics. As we have just shown, even a simple example such as this demands the inclusion of infinite sentences.

It could be argued that infinite sentences are meaningless since they can never be fully written down, said aloud, or expressed with a computer, but the same can be said for all non-repeating, non-terminating decimals, and we still wish to include in mathematics the existence of such numbers. Furthermore, if our world exists in an uncountable reality, any full expression of the existence of a particle would also demand the inclusion of infinite sentences. The fact that our infinite sentences can never be fully grasped by a human or computer is akin to the fact that the full expression of numbers like π can also never be fully realized by man or machine and that the full knowledge of a particle's location or history also remains out of reach. Any thoughts expressible with language must be confined to an at-best countably infinite subsection of an uncountable universe. The philosophical and scientific implications of this limitation are outside the scope of this paper, but those pondering the language-based revolution of AI and those contemplating the philosophies of human thought might find this highly thought provoking.

Theorem 2.4. *With finite vocabulary and countably infinite phrases, and, thus, countably infinite sentences,* $|NaturalLanguage| > |\mathbb{N}|$.

PROOF 1

Proof. Consider sentences of the form *The first digit is* b_1 *and the second digit is* b_2 *and the third digit is* b_3*...* where b_j is some digit in $\{0, 1, .., 9\}$ Sentences of this form (if allowed to go on indefinitely) form a one to one correspondence between all the numbers on the interval $[0, 1]$, and thus the set of all such sentences is grammatically correct (assuming the admission of infinite sentences) but uncountable in number.

If we wish to form a sentence that does not rely on numbers at all, but on an extended noun phrase. Let 0 correspond to *Alice* and 1 correspond to *Bob*, and consider the set of all numbers in $[0, 1]$ written in base two. For a given number $0.b_1b_2b_3\cdots$ we can form a unique grammatical sentence for each such number by creating a sentence of the form b_1 *thinks that* b_2 *thinks that* b_3 *thinks that...the grandmother fell.* We allow this phrase within the sentence to be finite but unbounded (good for terminating decimals) or infinite in length. If any adjacent numbers are the same (say $b_j = b_{j+1}$), we adjust the sentence to say that *...thinks that* b_j *thinks that they think that* b_{j+2} *thinks that...the grandmother fell.*. This can be extended as far as is needed and still remain grammatical. For example, the number 0.00010000... (a terminating number in base two equivalent to 1/16) would be equivalent to *Alice thinks that she thinks that she thinks that Bob thinks that the grandmother fell.* Now, without invoking numbers anywhere in the actual sentences, we can create a one-to-one correspondence between the numbers in $[0, 1]$ and this set of sentences. Letting the set of all such sentences be denoted G_1 we note that $G_1 \subset G_r$ where G_r

is the set of all grammatical sentences according to some grammar rule r that permits infinite and finite word length. Thus $|G_1| = 2^{\aleph_0} \leq |G_r|$. We seek to know $|G_r|$. To do this consider the set G as before in Theorem 0.2, the set with a completely permissive grammar rule (any combination of words is grammatical, this time allowing for infinite sentence length). We will break this down into three cases based on vocabulary size to prove the last three rows of the table.

Case 1: Finite Vocabulary Size

Let the size of the vocabulary (the number of words allowed in language) be some $N \in \mathbb{N}$. The set G corresponds to the union of the sets of all sentences of length $k < \infty$ (each denoted S_k) with the set of all sentences of infinite length denoted S_∞. That is,

$$G = \bigcup_{k=0}^{\infty} S_k \cup S_\infty \tag{3}$$

For each k, $|S_k| = N^k$. However, for S_∞, we have a set that is equivalent to a base N representation of all numbers in $[0, 1]$ that has uncountable cardinality. Since we have a countable union of finite sets and one uncountable set, the cardinality of G is also $2^{\aleph_0}$. Since $G_r \subseteq G$, $|G_r| \leq |G|$. Thus $|G_1| = 2^{\aleph_0} \leq |G_r| \leq |G| = 2^{\aleph_0}$, which implies $|G_r| = 2^{\aleph_0}$

Case 2: Countably Infinite Vocabulary Size

Now let the size of the vocabulary V be $\aleph_0$, which could correspond to the case of all human words plus all the integers. In this case, we again consider the size of G, the most general grammar permitting any combination of words as grammatical, whether of finite or infinite length. The set of all sentences of size k is S_k, and the set of all sentences of infinite length is S_∞. Each S_k can be written as $S_k = \{(v_1, v_2, ..., v_k) | v_j \in V \ \forall j\} = V^{\otimes k}$. Since $|V^{\otimes k}| = \aleph_0$, the cardinality of each S_k is $\aleph_0$. Meanwhile, S_∞ can be expressed as $S_\infty = \{(v_1, v_2, ...) | v_j \in V\}$. This set is uncountable since, if given a list of sentences $\{V_1, V_2, \cdots\}$ with each $V_i = v_{i1}v_{i2}v_{i3}\cdots$ where $v_{ij} \in V$ we can create a sentence $S = s_1 s_2 s_3 \cdots$ where

$$s_i = \begin{cases} yes & \text{if } v_{ii} = no, \\ no & \text{otherwise.} \end{cases} \tag{4}$$

This produces a sentence that is grammatical according to G but does not match any of the elements in the list because the i^{th} element of S never matches the i^{th} element V_i, thus making G uncountable. Since we again have $|G_1| = 2^{\aleph_0} \leq |G_r| \leq |G| = 2^{\aleph_0}$, we see that G_r must again have cardinality $2^{\aleph_0}$.

Case 3: Uncountably Infinite Vocabulary Size

Now let $|V| = 2^{\aleph_0}$, which could correspond to all human words plus the real numbers, and the length of sentences be finite or countably infinite. Once again, consider the most general grammar rule allowing all combinations of words as grammatical sentences. The set G of all such sentences is the set of all sentences of finite length combined with the set of all sentences of infinite length, just as in Case 2 above. Here, however, each S_k has cardinality $2^{\aleph_0}$ since a finite Cartesian product of an uncountable set is also countable. The case for S_∞ is perhaps less obvious. We can represent S_∞ as $\{(v_1, v_2, ...)|v_i \in V\}$. Since V has uncountable cardinality, we can form a one-to-one correspondence between all $v_1 \in V$ and all numbers in $[0, 1)$ and a one-to-one correspondence between all $v_2 \in V$ and all numbers in $[1, 2)$ and so forth for all v_i, so this infinite Cartesian product has the same cardinality as the interval $[0, \infty)$, which has cardinality $2^{\aleph_0}$, so $|G| = 2^{\aleph_0}$. The set of sentences with a grammar rule more restrictive than that in G but less restrictive that that of G_1 called G_r as before has cardinality between G_1 and G. Since we again have $|G_1| = 2^{\aleph_0} \leq |G_r| \leq |G| = 2^{\aleph_0}$, we see that G_r must again have cardinality $2^{\aleph_0}$.

□

3 Section II

Universal Algebra and Natural Language

Definition 1 (Universal Algebra)**.** *A universal algebra $\mathcal{A}$ is defined as an ordered pair (A, F) where:*

- *A is a non-empty set (termed the universe of $\mathcal{A}$).*
- *F is a set of n-ary operations on A.*

For every operation $f \in F$ that maps from A^n to A:

$$f : A^n \to A$$

every tuple $(a_1, a_2, \ldots, a_n) \in A^n$ has its image $f(a_1, a_2, \ldots, a_n)$ in A.

Definition 2 (Sentence Equality)**.** *Two sentences s_1 and s_2 in the set S are equal if and only if, for all i, the i^{th} word in s_1 is the same as the i^{th} word in s_2. Formally,*

$$s_1 = s_2 \iff \forall i,\ \textit{word}_i(s_1) = \textit{word}_i(s_2)$$

Theorem 3.1. *Within a natural language universal algebra with an adjective set A of arbitrary size, there exists a set S such that the algebra necessarily generates an infinite subset of S composed of sentences of infinite length (along with finite sentences).*

Proof. Assume a grammatical rule of adjective stacking. Given any adjective $a \in A$, the linguistic structure "The a cat" is a valid member of S. By the closure property of the universal algebra, this rule can be applied recursively, allowing us to create an infinite sequence of sentences in S:

$$\text{The } a \text{ cat, The } a\ a \text{ cat, The } a\ a\ a \text{ cat, } \ldots$$

Thus, for every adjective in A, there exists an infinite sequence of sentences of increasing length in S. As A is arbitrary in size, this further solidifies the presence of an infinite subset of S composed of sentences of infinite length in the system. □

Any observed bounded nature of sentences in practical settings is due to external social and cognitive constraints, not a limitation of the inherent mathematical properties of the natural language system. Given the previous theorem, the algebra of natural language theoretically supports the creation of an infinite subset of S composed of sentences of infinite length. However, in practice, linguistic expressions are kept finite due to human cognitive limitations such as memory and attention span, in the same way that non-repeating, non-terminating decimals exits, but are truncated due to the finite nature of humans and computers. These constraints are external to the algebraic structure of the language and are determined by the human users of the language.

4 Sentences with Infinite Adjective Stacking

The following sentences are constructed using the adjectives: black, grumpy, old, fat, lazy, grey, happy, furry, slim, and tiny. They exhibit an infinite adjective stacking using ellipsis:

1. The black black black black black black black ... cat leaped.
2. The grumpy old grumpy old grumpy old grumpy old ... cat leaped.
3. The fat lazy fat lazy fat lazy fat lazy ... cat leaped.
4. The grey happy grey happy grey happy grey happy ... cat leaped.
5. The furry slim furry slim furry slim furry slim ... cat leaped.
6. The tiny black tiny black tiny black tiny black ... cat leaped.
7. The old fat old fat old fat old fat old fat ... cat leaped.
8. The lazy grey lazy grey lazy grey lazy softred grey lazy grey ... cat leaped.
9. The happy furry happy furry happy furry happy furry happy furry ... cat leaped.
10. The slim tiny slim tiny slim tiny slim tiny slim tiny slim ... cat leaped.

4.1 Binary Representation

Syntax categories are represented as follows:

- Noun (N) → 10
- Adjective (Adj) → 01
- Verb (V) → 00

Words are represented by unique binary codes:

cat (N) :	10
black (Adj) :	01
grumpy (Adj) :	01
old (Adj) :	01
fat (Adj) :	01
lazy (Adj) :	01
grey (Adj) :	01
happy (Adj) :	01
furry (Adj) :	01
slim (Adj) :	01
tiny (Adj) :	01
leaped (V) :	00

Original Binary	**Adjective**	**Flipped Binary**	**New Adjective**
01 0100	black	01 1011	fat
01 0110	old	01 1001	furry
01 0111	fat	01 1000	grumpy
01 1010	happy	01 0101	tiny
01 1011	furry	01 0100	old
01 1001	grey	01 0110	slim
01 1000	lazy	01 0111	black
01 1100	grumpy	01 0011	lazy
01 1101	slim	01 0010	grey
01 1110	tiny	01 0001	happy

4.2 Diagonal Argument

Not flipped:

1. The 01_0100$_1$ 01_0100 01_0100 01_0100 01_0100 01_0100 ... 10_1001 00_0010.

2. The 01_0101 01_0110$_2$ 01_0101 01_0110 01_0101 01_0110 ... 10_1001 00_0010.
3. The 01_0111 01_1000 01_0111$_3$ 01_1000 01_0111 01_1000 ... 10_1001 00_0010.
4. The 01_1001 01_1010 01_1001 01_1010$_4$ 01_1001 01_1010 ... 10_1001 00_0010.
5. The 01_1011 01_1100 01_1011 01_1100 01_1011$_5$ 01_1100 ... 10_1001 00_0010.
6. The 01_1101 01_0100 01_1101 01_0100 01_1101 01_0100$_6$ 01_0100 ... 10_1001 00_0010.
7. The 01_0110 01_0111 01_0110 01_0111 01_0110 01_0111 01_0110$_7$ 01_0111 ... 10_1001 00_0010.
8. The 01_1000 01_1001 01_1000 01_1001 01_1000 01_1001 01_1000 01_1001$_8$ 01_1001 ... 10_1001 00_0010.
9. The 01_1010 01_1011 01_1010 01_1011 01_1010 01_1011 01_1010 01_1011 01_1010$_9$ 01_1011 ... 10_1001 00_0010.
10. The 01_1100 01_1101 01_1100 01_1101 01_1100 01_1101 01_1100 01_1101 01_1100 01_1101$_{10}$ 01_1101 ... 10_1001 00_0010.

From the list of binary representations of sentences, we extract a diagonal:

01 0100 01 0110 01 0111 01 1010 01 1011 . . .

Flipping the bits yields:

01 0111 01 0101 01 0100 01 1001 01 1000 . . .

4.3 Decoding the Sentence

Matching the flipped bits with our table:

Original Binary	**Adjective**	**Flipped Binary**	**New Adjective**
01 0100	black	10 1011	fat
01 0110	old	10 1001	furry
01 0111	fat	10 1000	grumpy
01 1010	happy	10 0101	tiny
01 1011	furry	10 0100	old
01 1001	grey	10 0110	slim
01 1000	lazy	10 0111	black
01 1100	grumpy	10 0011	lazy
01 1101	slim	10 0010	grey
01 1110	tiny	10 0001	happy

Using the new sequence, we get the following table with the bits flipped:

The list with the flipped bits below. We flip each binary code associated with an adjective. For example, 0100 will become 1011. Remember that no matter what, the flipped bit corresponds to another unique adjective.

1. The 01_1011$_1$ 01_0100 01_0100 01_0100 01_0100 01_0100 ... 10_1001 00_0010.
2. The 01_0101 01_1001$_2$ 01_0101 01_0110 01_0101 01_0110 ... 10_1001 00_0010.
3. The 01_0111 01_1000 01_1000$_3$ 01_1000 01_0111 01_1000 ... 10_1001 00_0010.
4. The 01_1001 01_1010 01_1001 01_0101$_4$ 01_1001 01_1010 ... 10_1001 00_0010.
5. The 01_1011 01_1100 01_1011 01_1100 01_0100$_5$ 01_1100 ... 10_1001 00_0010.
6. The 01_1101 01_0100 01_1101 01_0100 01_1101 01_1011$_6$ 01_0100 ... 10_1001 00_0010.
7. The 01_0110 01_0111 01_0110 01_0111 01_0110 01_0111 01_1001$_7$ 01_0111 ... 10_1001 00_0010.
8. The 01_1000 01_1001 01_1000 01_1001 01_1000 01_1001 01_1000 01_0110$_8$ 01_1001 ... 10_1001 00_0010.
9. The 01_1010 01_1011 01_1010 01_1011 01_1010 01_1011 01_1010 01_1011 01_0101$_9$ 01_1011 ... 10_1001 00_0010.
10. The 01_1100 01_1101 01_1100 01_1101 01_1100 01_1101 01_1100 01_1101 01_1100 01_0010$_{10}$ 01_1101 ... 10_1001 00_0010.

Once we have them flipped we decode the infinite sentence. Notice again that each flipped binary sequence corresponds to a new adjective not in that position. With this, no matter what row you choose, the adjective at the nth position will be different than any sentence on each row of the infinite list.

1. The **fat** 01_0100 01_0100 01_0100 01_0100 ... 10_1001 00_0010.
2. The 01_0101 **furry** 01_0101 01_0110 01_0101 01_0110 ... 10_1001 00_0010.
3. The 01_0111 01_1000 **grumpy** 01_1000 01_0111 01_1000 ... 10_1001 00_0010.
4. The 01_1001 01_1010 01_1001 **tiny** 01_1001 01_1010 ... 10_1001 00_0010.
5. The 01_1011 01_1100 01_1011 01_1100 **old** 01_1100 ... 10_1001 00_0010.

6. The 01_1101 01_0100 01_1101 01_0100 01_1101 **slim** 01_0100 ... 10_1001 00_0010.

7. The 01_0110 01_0111 01_0110 01_0111 01_0110 01_0111 **black** 01_0111 ... 10_1001 00_0010.

8. The 01_1000 01_1001 01_1000 01_1001 01_1000 01_1001 01_1000 **lazy** 01_1001 ... 10_1001 00_0010.

9. The 01_1010 01_1011 01_1010 01_1011 01_1010 01_1011 01_1010 01_1011 **grey** 01_1011 ... 10_1001 00_0010.

10. The 01_1100 01_1101 01_1100 01_1101 01_1100 01_1101 01_1100 01_1101 01_1100 **happy** 01_1101 ... 10_1001 00_0010.

And with that, we have successfully decoded a unique sentence that is not present in the infinite list, thus illustrating the diagonal argument.

You can verify that the new sentence is different than any sentence at any nth position on each row. By flipping the bits and decoding we ensure each new adjective is different from the one that was there originally in the nth position. The first adjective is different than the adjective that was in the first position, the second is different from the adjective that was in the second position and so forth making sure that the new sentence is different than any sentence on any row.

1. The **fat black black black black** ... 10_1001 00_0010.

2. The **old furry old old old** ... 10_1001 00_0010.

3. The **fat lazy grumpy lazy fat** ... 10_1001 00_0010.

4. The **grey happy grey tiny grey** ... 10_1001 00_0010.

5. The **furry grumpy furry grumpy old** ... 10_1001 00_0010.

6. The **slim black slim black slim slim** ... 10_1001 00_0010.

7. The **old fat old fat old fat black** ... 10_1001 00_0010.

8. The **lazy grey lazy grey lazy grey lazy lazy** ... 10_1001 00_0010.

9. The **happy furry happy furry happy furry happy furry grey** ... 10_1001 00_0010.

10. The **grumpy slim grumpy slim grumpy slim grumpy slim grumpy happy** ... 10_1001 00_0010.

From the diagonal after flipping the bits and decoding we get the infinite sentence below:

The fat furry grumpy tiny old slim black lazy grey happy... cat leaped.

This new sentence doesn't match any of the original ones, demonstrating the diagonal argument's validity.

Definition 3 (Universal Algebra)**.** *A universal algebra is a structure $\mathcal{A} = (A, F)$ where:*

- *A is a non-empty set called the domain or carrier set.*
- *F is a set of operations $f : A^n \to A$, with $n \geq 0$. Each operation is n-ary where n is called its arity. The set F may contain operations of different arities.*

The important property of a universal algebra is that it is closed under its operations. That is, for any operation $f \in F$ and any elements $a_1, a_2, \ldots, a_n \in A$, the result $f(a_1, a_2, \ldots, a_n)$ is also in A.

Definition 4 (Sentence Equality in Natural Language)**.** *Two sentences S_1 and S_2 in natural language are considered equal if and only if they have the exact same sequence of words. That is, word-by-word they are identical.*

5 The Case for Infinite Sentences

Lemma 1 (Existence of Infinite Sentences)**.** *In a universal algebra modeling natural language (denoted as $\mathcal{NLA}$), there necessarily exist an infinite number of sentences of infinite length.*

Proof. Let $\mathcal{NLA} = (A, F)$ represent our natural language algebra where:

- A is a set containing all valid grammatical constructs, including words, phrases, and sentences of the language.
- F is a set of syntactic operations that combine constructs to produce new grammatical constructs.

Given that A includes words like adjectives and that there exists an operation $f \in F$ that allows for adjective stacking (i.e., placing one adjective after another), we can recursively apply f to generate sentences of increasing length.

For any adjective $a \in A$, applying f repeatedly, we get:

$$f(a) \to f(f(a)) \to f(f(f(a))) \to \ldots$$

This process can continue indefinitely, leading to sentences of infinite length.

Furthermore, since for any two distinct adjectives a_1 and a_2 the results $f(a_1)$ and $f(a_2)$ are distinct (due to our definition of sentence equality), we can generate an infinite number of distinct sentences of infinite length by varying our choice of adjectives.

Thus, $\mathcal{NLA}$ necessarily generates an infinite set of infinite-length sentences. □

Let's suppose, for the sake of being extra sure, that we take the definition of a sentence at face value. A sentence needs to be complete to be a sentence. The problem here is when we consider sentences like the one below. If we were to make it an infinitely long sentence, it would look like the next sentence. Viewed this way, somebody can argue "that's not a complete sentence. There is no verb and it does not end." To that, we say, let's consider the third sentence below. This sentence is clearly grammatical, even if we make it infinite. It has all the elements already; we can call it a grammatical sentence even if it never ends the list of adjectives.

- The black cat fell.
- The big black Adj Adj Adj....
- The cat that fell is big black Adj Adj Adj....

6 Natural Language Reacursivity and $2^{\aleph_0}$

6.1 The Recursive Grammatical Structure

Definition 5 (Recursive Grammar G)**.** *Our grammar G is based on a template that can be succinctly represented by $[SP[NP][VP[NP]]]$. Let's delve deeper into understanding each of its constituents. This in line with the current accepted linguistic theory developed by Chomsky (1957). However, for our current purposes we can define $[SP[NP][VP[NP]]]$ as a context free grammar.*

Definition 6 (Sentence Phrase - SP)**.** *An SP stands for an entire sentence in our grammar G. It essentially combines a Noun Phrase and a Verb Phrase to form a coherent sentence: $[NP][VP]$.*

Definition 7 (Noun Phrase - NP)**.** *An NP is essentially a placeholder for entities. In our grammar G, we limit ourselves to the following entities: {John, Mary, Alice, Bob}.*

Definition 8 (Verb Phrase - VP)**.** *A VP stands for actions or relationships. In our grammar G, this can be further elucidated with examples like: {loves [NP], hates [NP], knows [NP], sees [NP]}.*

7 Definition of the Context-Free Grammar

Definition 9 (Context-Free Grammar for [SP [NP] [VP [NP]]])**.** *A context-free grammar G for the sentence structure [SP [NP] [VP [NP]]] is a* 4*-tuple* $G = (V, \Sigma, R, S)$ *where:*

- *V is a set of variables,* $V = \{S, NP, VP, N', N, Adj, D, V\}$.
- *Σ is a set of terminals. For the sake of this exposition, we will assume* $\Sigma = \{\textit{The, neighbors, big, black, fat, grey, cat, jumped}\}$. *In a real-world scenario, Σ would encompass a much larger vocabulary.*
- *R is a set of production rules defined as:*

$$\begin{aligned}
S &\to \textit{NP VP} \\
NP &\to \textit{D N'} \\
N' &\to \textit{Adj N'}|\, N \\
N &\to \textit{cat}\,|\,\textit{dog}\,|\,\textit{bird} \\
Adj &\to \textit{neighbors}\,|\,\textit{big}\,|\,\textit{black}\,|\,\textit{fat}\,|\,\textit{grey} \\
D &\to \textit{The} \\
VP &\to V|\; \textit{V NP} \\
V &\to \textit{jumped}\,|\,\textit{chased}
\end{aligned}$$

- *S is the start symbol, representing a sentence.*

8 Infinite Adjective Stacking

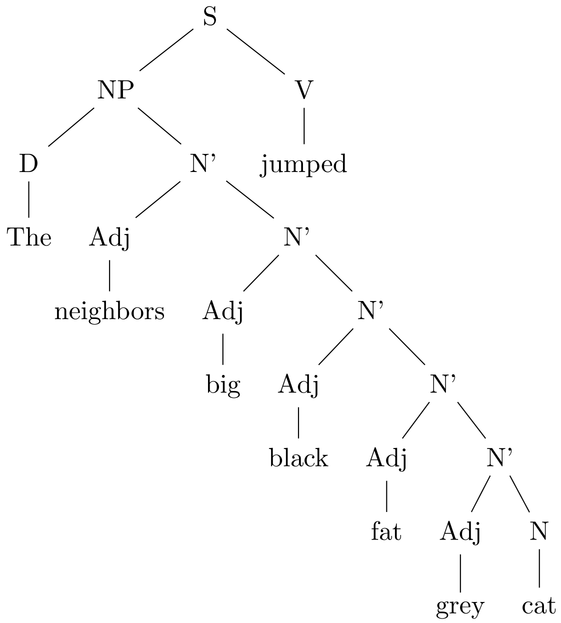
In this section, we demonstrate that standard everyday syntactic structures can generate an infinite list of sentences. Consider the sentence below:

The cat jumped.

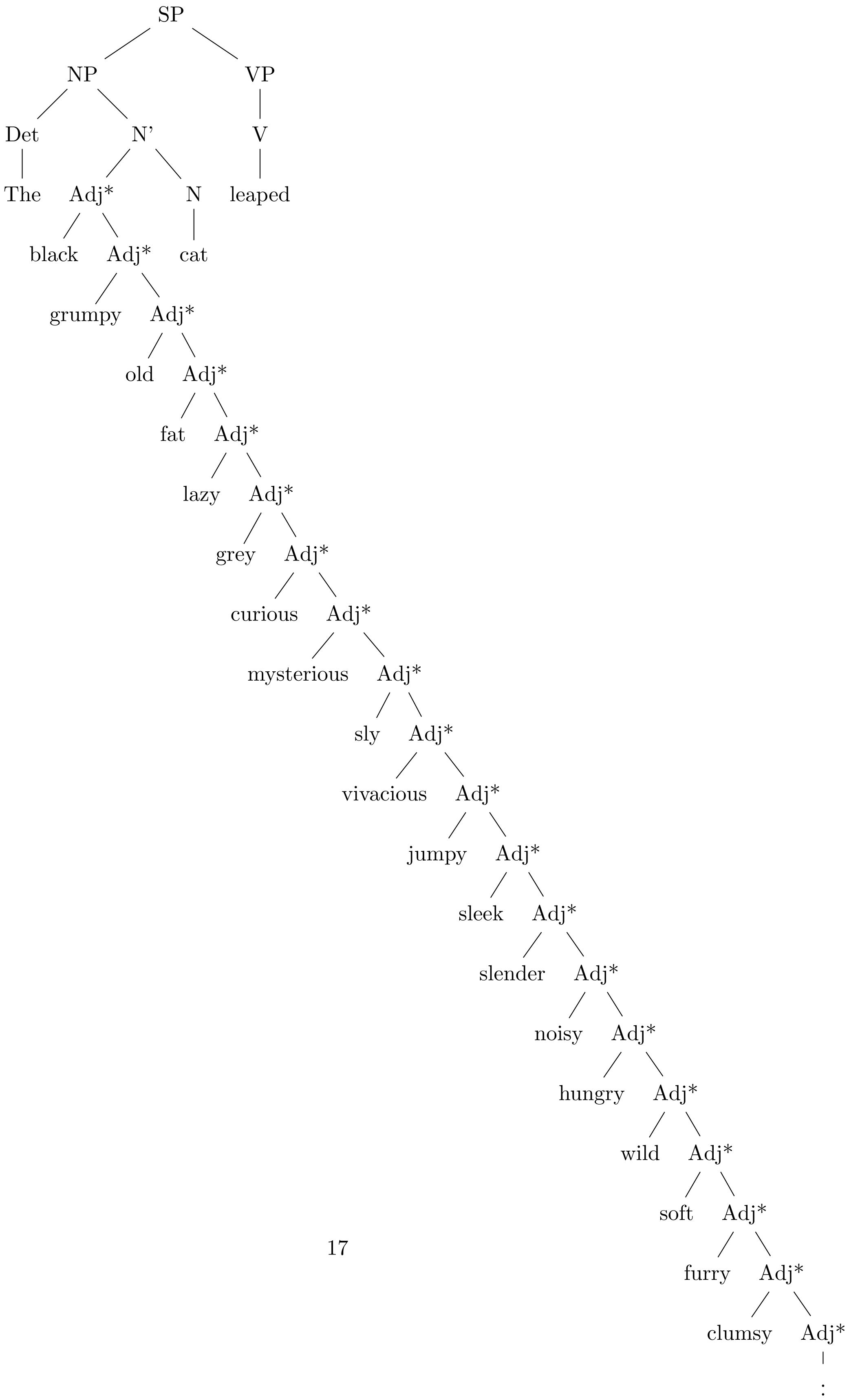
This sentence, commonly used in everyday life, is finite. However, due to the recursive nature of phrases, we can infinitely expand this sentence through adjective stacking. An illustration is provided below:

1. The cat jumped.
2. The grey cat jumped.
3. The fat grey cat jumped.
4. The black fat grey cat jumped.
5. The big black fat grey cat jumped.
6. The neighbors big black fat grey cat jumped.
7. The $\mathrm{Adj}_1\mathrm{Adj}_2\ldots\mathrm{Adj}_{n+1}$ cat jumped.

One might consider listing all sentences of length n and pairing them one-to-one with the natural numbers. However, as we will explore, this inherent attribute of language makes it uncountable. Below, the tree representation of the syntactic structure is provided:



The core structure of NP in our grammar G encompasses a single entity, such as “John” or “Mary.” However, this simplicity belies the true depth of linguistic expressiveness. Noun Phrases can be endlessly expanded by adding more adjectives, creating structures like “the tall man in the red shirt from the corner of the street.” Consider the sentence “The intelligent robot built by the brilliant scientist won the chess tournament.” In this sentence, the Noun Phrase “the intelligent robot built by the brilliant scientist” contains a cascade of adjectives and additional Noun Phrases, each nested within the other. This recursive expansion, while theoretically infinite, can be represented within the finite framework of our grammar G. Below we provide a representation of the infinite stacking:


SP
NP
VP
Det
N’
V
The
Adj*
N
leaped
black
Adj*
cat
grumpy
Adj*
old
Adj*
fat
Adj*
lazy
Adj*
grey
Adj*
curious
Adj*
mysterious
Adj*
sly
Adj*
vivacious
Adj*
jumpy
Adj*
sleek
Adj*
slender
Adj*
noisy
Adj*
hungry
Adj*
wild
Adj*
soft
Adj*
furry
Adj*
clumsy
Adj*
⋮

8.1 Encoding Ajective Stacking

Our binary encoding scheme B accommodates this infinitude by offering a systematic way to represent these expanding structures. Each additional adjective or embedded NP is assigned a unique binary sequence, ensuring that even infinitely expanding Noun Phrases can be mapped to distinct binary strings.

8.2 Binary Encoding for G

Definition 10 (Binary Encoding Rationale)**.** *To perform an in-depth analysis of the infinite potential of G-sentences, we employ a binary encoding. The primary reason for this is the dichotomous nature of binary numbers which can clearly differentiate between the various components of our grammar. The binary codes in the interval* $[0, 1]$ *are defined as follows:*

$$B = \left\{ x \in [0,1] \mid x = \sum_{n=1}^{\infty} \frac{a_n}{2^n}, \text{ where } a_n \in \{0,1\} \text{ for all } n \right\}.$$

Definition 11 (Binary Encoding Function B)**.** *B functions as a bijection between the elements of our grammar G and a set of binary strings.*

We will assign to DPs the prefix 00, NPs the prefix 11, VPs 01 and Adjectives 10. And to each word a unique binary code. The binary code can be infinite.

Nouns Encoding:
- The [DP] - `00 1100`

Noun Phrases Encoding:
- cat [NP] - `11 1101`

Verb Phrases Encoding:
- jumped [VP] - `01 1101`

Adjective Phrases Encoding:
- black [AP] - `10 00110100`
- old [AP] - `10 10110110`
- grumpy [AP] - `10 00100110`
- .
- .
- .

To make sure we always get a new adjective when we flip the bit we will first make the set of all AP, A. To each member of A we will assign the corresponding category prefix and a unique finite binary code. Each binary code is unique. Now we define a function that flips all the available binary code to have a new set of binary codes. For example 10 0011001, would be 10 1100110. We leave the category prefix alone. Now we reassign a new flipped binary code to each adjective in A. Each a in A, a = ddddd, ddddd, is a set of two binary codes. Both represent a. However, the second dddd corresponds to another adjectives when not flipped. The same for the first binary code. This ensures that no matter what we will get a new adjective when we flip the bits.

Remark. It is imperative to note that our binary representations have been judiciously chosen to avoid any redundancy. This ensures the preservation of the bijection of B.

8.3 Decoding Procedure

Definition 12 (Decoding Algorithm D)**.** *D is a well-defined mapping that takes any given binary string from the set $\{0,1\}^*$ and transforms it into its corresponding linguistic counterpart in G.*

8.3.1 Algorithm D

Given a binary string b of length n, execute the following steps:

1. Initialize an empty sentence S.
2. Initialize a pointer p at position 1 of b.
3. While $p \leq n$:
   (a) Based on the bit value at pointer p, classify the word type using the prefix table.
   (b) Using the classified type and subsequent bits, look up the corresponding linguistic representation in the encoding table.
   (c) Append the linguistic representation to sentence S.
   (d) Move the pointer p forward by the number of bits corresponding to the identified linguistic representation.
4. Once $p > n$, terminate and return the reconstructed sentence S.

Remark. The decoding algorithm's precision depends on the non-overlapping and deterministic nature of the encoding scheme. As a result, each valid encoded string is associated with a unique sentence in G.

9 Properties of the Encoding Function

9.1 Injectivity of B

Lemma 2. *The encoding function $B : G \to \{0,1\}^*$ is injective.*

Proof. Let's hypothesize that two distinct sentences S_1 and S_2 within the confines of G share an identical encoding, denoted as $B(S_1) = B(S_2)$. However, the essence of our encoding scheme is its promise of distinctiveness for each element. Consequently, such a scenario directly challenges the foundational premise of our encoding scheme. As this situation is untenable, it becomes manifest that B is inherently injective. □

9.2 Surjectivity of B

Before we can proceed to demonstrate the surjectivity of B, it is crucial to precisely understand the nature of its codomain.

9.2.1 Codomain of B

The codomain of B, which consists of all valid binary strings, is meticulously crafted via our encoding scheme. A valid binary string in this context refers to any binary sequence that corresponds to a legitimate sentence constructed using grammar G.

Lemma 3. *The encoding function B is surjective.*

Proof. Let b be any valid binary string in the codomain of B.

Step 1: Using our decoding algorithm, we can map b to a sentence S in grammar G.

Step 2: Now, by the definition of our encoding function, we know there exists an encoding $B(S)$ that maps S back to a binary string.

Step 3: Given that our encoding is consistent and unique, it follows that $B(S) = b$.

Thus, for every valid binary string b in the codomain of B, there exists a sentence S in G such that $B(S) = b$. This establishes the surjectivity of B. □

9.3 Bijective Nature of B

With the injectivity and surjectivity of B established, we can now deduce its bijective nature.

9.3.1 Bijection Proof

Corollary 1. The encoding function B is a bijection.

Proof. The function B has already been proven to be injective, meaning every distinct sentence in G maps to a unique binary string. Additionally, we have established its surjectivity, indicating that every valid binary string in its codomain corresponds to a sentence in G.

Since B meets both these criteria — injectivity and surjectivity — it is by definition a bijection between the set of sentences in G and the set of valid binary strings. □

Remark. The precision and reliability of our decoding procedure stems from the unique and non-overlapping nature of our encoding mechanism.

10 Properties of the Encoding Function

10.1 Injectivity of B

Lemma 4. *The function B, as defined from G to $\{0,1\}^*$, is injective.*

Proof. Let's assume for the sake of contradiction that two distinct sentences from G map to the same binary string. Given our construction where each component of a sentence has a unique binary representation, this assumption leads to a direct contradiction. Hence, it can be conclusively stated that B is injective. □

10.2 Surjectivity and the Decoding Function

Lemma 5. *The function B, as mapped from G to $\{0,1\}^*$, is surjective. Moreover, the decoding function D acts as its precise inverse.*

Proof. For any string s in the codomain of B, the function $D(s)$ invariably generates a valid sentence in G. Hence, it can be deduced that B is surjective. □

11 Main Result: Uncountability of G-sentences

11.1 Ensuring Syntactic Validity in Diagonalization

Definition 13 (Fallback Syntactic Patterns)**.** *We choose certain special patterns, which are termed as fallback syntactic patterns. The chief aim of these patterns is to ensure that any binary string created during diagonalization represents a valid sentence within G.*

11.1.1 Fallback Syntactic Patterns

- SP can be constructed using any NP followed by any VP.
- VP can be built from any of the four VPs or from the fallback pattern {0, 1}.
- NP can be created using any of the four NPs or from the fallback pattern {0, 0}.

11.2 Cantor's Diagonalization

We have established the injectivity of B and the surjectivity of B with its inverse function D. Now, we will utilize Cantor's diagonalization argument to prove the uncountability of sentences in G.

Theorem 11.1 (Uncountability of Sentences in G)**.** *The set of all sentences generated using the recursive grammar G is uncountable.*

Proof. To prove the uncountability of sentences in G, we assume, for the sake of contradiction, that there exists a countable enumeration of all sentences in G. Let's list them as:

$$\begin{aligned}
s_1 &= b_{11}b_{12}b_{13}\ldots \\
s_2 &= b_{21}b_{22}b_{23}\ldots \\
s_3 &= b_{31}b_{32}b_{33}\ldots \\
&\vdots
\end{aligned}$$

Where each b_n is a word and composed of possibly infinitely many binary digits.

$$b_n = d_1d_2d_3\ldots d_{n+1}\ldots$$

Now, we will construct a new binary string s' using Cantor's diagonalization approach:

$$s' = d_1d_2d_3\ldots$$

where

$$d_i = \begin{cases} 0 & \text{if } b_{ii} = 1 \\ 1 & \text{if } b_{ii} = 0 \end{cases}$$

The string s' is clearly different from each s_i because it differs in the i^{th} position. However, given our fallback syntactic patterns, we can ensure that s' also represents a valid sentence in G.

This new sentence s' contradicts our initial assumption of having an enumeration of all sentences. Thus, our initial assumption was false, and the set of all sentences constructed using G is indeed uncountable. □

12 Encoding Natural Language using Decimal Numbers

In this section we will construct a more direct proof. I will assign random adjective the numbers from 0-9. The create sentences where we have "The 01234... cat jumped" which then could be decoded into "The black big grumpy long cat jumped." By encoding adjectives as irrational numbers we can proof more directly that they generate a set of uncountable sentences.

12.1 Adjective Encoding

We begin by encoding adjectives using the decimal digits. Let's represent this encoding with the function $E : \text{Adjective} \to \{0, 1, 2, \ldots, 9\}$. For instance:

$$\begin{aligned} E(\text{"black"}) &= 0 \\ E(\text{"grumpy"}) &= 1 \\ E(\text{"happy"}) &= 2 \\ E(\text{"large"}) &= 3 \\ &\vdots \end{aligned}$$

12.2 Constructing Sentences

Sentences in our schema are of the form:

$$s = \text{"The "}a_1a_2a_3\ldots\text{" cat jumped"}$$

where each a_i is a decimal digit corresponding to an adjective as per the encoding E.

12.3 Proof of Uncountability

Theorem 12.1. *The set of all sentences constructed using the grammar G and the encoding E is uncountable.*

Proof. Suppose, for the sake of contradiction, that the set of all such sentences is countable. Then, there exists an enumeration:

$$\begin{aligned} s_1 &= \text{"The "}a_{11}a_{12}a_{13}\ldots\text{" cat jumped"} \\ s_2 &= \text{"The "}a_{21}a_{22}a_{23}\ldots\text{" cat jumped"} \\ s_3 &= \text{"The "}a_{31}a_{32}a_{33}\ldots\text{" cat jumped"} \\ &\vdots \end{aligned}$$

We now construct a new sentence s':

$$s' = \text{"The "}d_1d_2d_3\ldots\text{" cat jumped"}$$

where

$$d_i = \begin{cases} 0 & \text{if } a_{ii} \neq 0 \\ 1 & \text{if } a_{ii} = 0 \end{cases}$$

The sentence s' is distinct from every sentence s_i in our enumeration because it deviates in the i^{th} adjective. However, s' is still a valid sentence in G. This leads to a contradiction, meaning our initial assumption that the set of sentences was countable is incorrect. Therefore, the set of sentences constructed using G and E is uncountable. □

13 Uncountability in Natural Language

Theorem 13.1 (Uncountability of Natural Language Sentences)**.** *Any natural language with property X, seen from a mathematical point of view, generates a set of uncountable infinite sentences.*

Definition 14 (Property X: Unbounded Modifiability)**.** *A natural language exhibits unbounded modifiability if it has linguistic structures that can be extended indefinitely without violating the syntactic or semantic rules of the language.*

Proof. Assume a language L exhibits unbounded modifiability. This implies that there exists at least one linguistic construct in L that can be indefinitely extended. For the sake of argument, let this construct be represented as a sequence S such that every element s in S corresponds to a linguistic entity in L.

We can create a bijective mapping between each element s in S and a digit in the set $\{0, 1, 2, ..., 9\}$. By Cantor's diagonalization argument, given any countable enumeration of sequences in S, we can construct a new sequence S' that differs from every sequence in the enumeration.

As L allows for unbounded modifiability, S' also corresponds to a valid linguistic entity in L. Thus, L contains uncountable infinite sentences.

□

14 Another Argument in Favor of Infinite Sentences

We might consider the point that sentences must terminate because they must express a complete thought. The notion that a sentence must complete a thought leads to the inclusion of infinite sentences.

Consider the example "The ratio of a circle's diameter to it's radius is 3.14159...". In order to express this thought fully, the sentence must go on forever. If it terminates, we have not expressed the thought fully.

Now consider the argument from another angle. Consider a series of sentences enumerated below

$$The\ first\ digit\ is\ 1. \tag{5}$$

$$The\ second\ digit\ is\ 4. \tag{6}$$

$$\vdots \tag{7}$$

There is no argument that can be made that we cannot continue producing sentences like this forever. There is also no reason why we cannot start a new enumeration for a new number that also goes on forever. Allowing this list to go on forever (i.e. the number of elements of a list to be infinite), we can put each separate list in 1-1 correspondence to each number on the interval $[0, 1]$, which means that natural language can express an uncountable number of ideas if the set of sentences can continue on forever, even if the number of possible sentences themselves must be of countable cardinality.

However, if one simply replaces “.” with “ and ” in a list like this, one immediately realizes that we must permit infinite sentences since we clearly permit an infinite list of sentences like this.

One might also argue that we can only generate such sentences from math, since the above example is so clearly mathematical. Let us dispel that notion. Consider sentences of the form “The n^{th} digit is k.” Where $n \in \mathbb{N}$ and $k \in \{0,1\}$. Let the base two representation of some arbitrary number in $[0,1]$ be $b_1 b_2 \cdots$ with $b_i \in \{0,1\} \forall i$. We list the sentence i^{th} sentences a s “Bob said the paint is red.” if $b_i = 0$ and “Bob said the paint is green.” if $b_i = 1$.

$$\textit{Bob said the paint is red.} \tag{8}$$

$$\textit{Bob said the paint is green.} \tag{9}$$

$$\vdots \tag{10}$$

Such sentences always terminate , and, so long as the sequence of sentences is allowed to go on forever, can form a 1-1 correspondence to the numbers in [0,1], which have cardinality $2^{\aleph_0}$. And, if we accept that we can conjoin the sentences in order with “ then ”, must form uncountably many infinite sentences.