On the origin of physics from mathematical logic

Alexandre Harvey-Tremblay

aht@protonmail.ch

Published: May 18, 2017
Revised: May 25, 2017 and July 11, 2017

In this work, I present a formal construction of the axiomless position. This construction ultimately leads to the theory of everything in physics (ToE).

Part I is the axiomless derivation of the ToE. From this derivation I obtain a master equation formulated as a Gibb’s ensemble and relating the algorithmic notions of program-observables to that of entropy.

Part III is the thesis that this master equation is indeed the ToE. To convince you of that, I recover, again in an axiomless manner, the exact mathematical formulation of the major theories of physics; including statistical mechanics, quantum mechanics, special and general relativity. These equations are derived entirely from pure reason with no appeal to physical observations.

Contents

1 Philosophy 3
  1.1 Explanatory gap ................................................. 3
  1.2 State of physics ................................................. 4

I An axiomless derivation of the Theory of Everything 5
  1.3 Axioms ......................................................... 5
  1.4 Language ....................................................... 6
  1.5 Arithmetic ..................................................... 13
  1.6 Translation ..................................................... 16
  1.7 Multiplication ............................................... 17
  1.8 Set theory ..................................................... 20
  1.9 Representation of any theory of knowledge ................. 21

2 The Theory of Everything 23
  2.1 Elimination of physically impossible theories .............. 24
  2.2 The universal theory of knowledge .......................... 25
  2.3 The halting probability ..................................... 26
  2.4 The universal Turing machine ............................... 28
  2.5 The halting partition ....................................... 28

II The Theory of Everything 30
III Proof of the Theory of Everything

3 Thermodynamics
   3.1 Energy ........................................... 34
   3.2 Time ........................................... 35
   3.3 Hard arrow of time ................................... 37
   3.4 Quantum information ................................... 38
   3.5 Experiments & observers ................................... 41
   3.6 The universe ....................................... 41
   3.7 Discussion ......................................... 43

4 Space
   4.1 Maximum entropy ................................... 45
   4.2 Position ........................................... 48
   4.3 Schwarzschild radius .................................... 50
   4.4 Hawking radiation ...................................... 51
   4.5 Newton’s law of inertia .................................... 52
   4.6 Newton’s law of gravity .................................... 53
   4.7 Schrödinger’s equation .................................... 54

5 Spacetime
   5.1 Light-cone ......................................... 60
   5.2 Speed of light ....................................... 61
   5.3 Lorentz transformation ..................................... 62
   5.4 General Relativity ...................................... 62
   5.5 Quantum Field Theory ..................................... 63

6 Conclusion ........................................... 64
1 Philosophy

What is the axiomless position?

Plato\(^1\) recognized that most of the disagreements in philosophy are ultimately linked to the choice of axioms made by the parties involved. He believed that by grinding away at the assumptions made in support of any argument, one could recover a kind of universal truth. He believed that this universal truth, comprised of whatever survives the grinding process, could ultimately be used to build a logical framework in a manner that is entirely irrefutable. This is the axiomless position.

The goal of obtaining an axiomless framework was revisited by René Descartes in 1641. He used a universal doubt method to obtain the ‘cogito ergo sum’ or ‘I think, therefore I am’, claiming the existence of the thinking self to be an absolute truth and suggested that it should be used as the foundation of philosophy.

The mathematician David Hilbert, in 1900, posed 23 problems to the mathematical community. Hilbert’s second problem is of interest here. He challenged mathematicians to establish a system of axioms to explain all of science and mathematics and then to prove that this system is consistent. In this spirit, Bertrand Russell published Principia Mathematica in 1910, proving that logic was the same as mathematics.

However, in 1930, Kurt Gödel would publish a proof on the incompleteness of arithmetic. His theorem proved that any formal system strong enough to construct self-referential sentences will necessarily have true, but unprovable statements. Hence it would be incomplete. This suggests that the axiomatic approach is highly problematic.

In this work, I present a construction to the axiomless position first posed by Plato. The proof references results in algorithmic information theory that were not available until very recently.

1.1 Explanatory gap

An axiomatic theory of everything will necessarily have a foundational gap in its ability to explain the universe. Indeed, no axiomatic theory can explain why its axioms are true over other axioms. A ToE instead derived from the axiomless method will not have this explanatory gap.

Remark 1.1. The falsifiability argument is a notable non-constructive exception consisting of claiming that these axioms best reproduce the scientific observations to date. But if such an argument is used to justify its axioms, then the theory cannot provably be the ToE and it would forever be an potentially intermediary theory.
1.2 State of physics

The formal theories of physics are built upon axioms which are not proven from reason but are instead justified by a series of experiments or observations. For example, why is the speed of light a constant? Because of the failure of the Michelson–Morley experiment. And so on. Since our observational capabilities are limited by our technology, the consequence of building physics upon a series of observations is that we end up with multiple logically independent theories which don’t quite fit together.
Part I

An axiomless derivation of the Theory of Everything

1.3 Axioms

In this work, we will use the following definition for axioms.

**Definition 1.2 (Axiom).** An axiom is an unprovable sentence of a language that can be true or false within a formal logic system, but is considered to be true without proof.

We emphasize the underlined elements of the definition;

1. An axiom must be an unprovable sentence. This is required otherwise we would could call any theorem an axiom which would negate the distinction.

2. An axiom must be a sentence that can be true or false. This prevents tautologies, necessary truths and contradictions from being axioms. For example, tautologies are considered to be theorems because they are provably always true.

3. The choice of a formal logic system must be established before we can formulate a correct axiom for it. For example, the sentences of first-order logic are not compatible with propositional logic hence writing axioms as correct sentences comes after the choice of a logic system.

In this work, we will use first order logic to write axioms and statements. However, to prove these statements we will not use the full facilities of first order logic. Instead, we will only make use of a maximally reduced logic system.

To prove first order logic sentences, we will only accept a direct construction of the object the existence of which is claimed by the sentence. As a trivial example, we could prove the existence of the symbol 1, represented in first order logic by \( \exists x [\text{isSymbol}(x) \land (x = 1)] \) by writing 1 as the proof. Sentences proven that way are said to be primitive theorems.

**Definition 1.3 (Primitive Theorem).** A primitive theorem is a sentence of first order logic that is provable by direct construction of the object the existence of which it claims. Theorems proven exclusively from other primitive theorems are also primitive theorems.
Remark 1.4 (Primitive theorems are not axioms). An axiom is an unprovable sentence of a language, whereas primitive theorems are provable with a proof by construction.

Definition 1.5 (Axiomless theory). A theory of logic is axiomless if it contains no statement respecting the definition of the axiom.

This proof method is significantly diminished from the more expressive formal logic systems that are commonly in use. To reduce assumptions to a minimum, we have gotten rid of most principles of logic such as the law of excluded middle, etc. We only preserve the proof by construction as anything less would be too weak to prove anything.

In a very real sense the pen and paper we used to write formal logic statements becomes a source of undeniable evidence for some statements about the existence of language, symbols and their properties. As we are limited to only constructing sentences or symbols to offer as proof, it is natural to use language as a starting point to build upon.

1.4 Language

It is quite difficult, and most likely impossible, to even imagine a concept that cannot be formulated in language. As such, language is a very general concept - perhaps even the most general of all concepts.

The limits of my language mean the limits of my world [...] Whereof one cannot speak, thereof one must be silent.

–Ludwig Wittgenstein

So how will we define language? The first potential concern is that we are unavoidably using language to define language. This might seem circular, but it is a consequence of how fundamental the concept is. Furthermore, this will allow us to prove the existence of language axiomlessly, so for us it will actually be an advantage.

A real problem however is that of the infinite regression of the definitions. Suppose we define a language by its symbols. Then how do we then define symbols? Do we say that symbols are unique identifiers? If so, then what is an identifier - is it a shape in one’s mind? If so, what is a mind, or a shape? This goes on forever. To break the cycle we introduce a precision cutoff in our definitions and we instead assume that the reader knows intuitively what is talked about.

Formally speaking, these cutoffs are primitive notions.

Definition 1.6 (Primitive notion). A primitive notion is a term that we use but that we do not define. The term should be understood by a mixture
of examples, intuition and by the theorems and definitions that result from its usage.

For example, Euclidean geometry under Hilbert’s axioms has five recognized primitive notions; point, line, plane, congruence, betweeness, and incidence. Set theory has two; set and element of.

To define language, we will introduce two primitive notions.

**Primitive Notion 1.7 (Symbol).** A symbol is a unique distinguishable identifier. It can be a shape, a sound, a sign, etc. By definition, there are only two predicates for a symbol and both are primitive predicates. They are,

\[ isSymbol(x) \] (true if \( x \) is a symbol)
\[ areDistinguishable(x, y) \] (true if \( x \) and \( y \) are different symbols)

**Primitive Notion 1.8 (Sentence).** A sentence consists of taking multiple symbols and joining them together in a single group or unit. The order of occurrence of the symbols in the sentence matters and repetitions are allowed. The primitive predicate related to the sentence is

\[ isSentence(p) \] (true if \( p \) is a sentence)

**Remark 1.9.** For completeness, single symbols as well as the empty sentence are also considered to be sentences. The empty sentence will be denoted with \( \epsilon \).

**Remark 1.10.** Unlike symbols, sentences can have many more predicates. For example, containsSymbol(\( x, p \)) could be one, isLength(\( p, n \)) could be another, etc. We will not concern ourselves with these at this point. If we need any of these later on, we will define them from within the theory itself.

We will now pose definitions relying on these primitive notions.

**Definition 1.11 (Alphabetical sentence).** An alphabetical sentence \( S_\alpha \) is a sentence of finite length with no repetitions of symbols.

**Definition 1.12 (Language).** A language \( L \) is defined by a specific alphabetical sentence \( S_\alpha \) such that;

1. The order of occurrence of the symbols in \( S_\alpha \) is the alphabetical order of \( L \).
2. If a sentence contain symbols not present in \( S_\alpha \), then it is not a sentence of \( L \).

As examples, the following languages (on the left) are defined by their \( S_\alpha \) (on the right).
nullary := ε
unary := 1
binary := 01
ternary := 012
decenary := 0123456789
roman alphabet := abcdefghijklmnopqrstuvwxyz
roman numbers := IVXG

Remark 1.13 (Notation). When writing the sentences of a language, we adopt the following notational conventions to eliminate ambiguity. For example, ambiguity occurs if we write 10; is it the decimal number ten, or the binary number two.

- If we list all possible sentences of a language from shortest to longest and from alphabetical first to alphabetical last, we suffix the sentences with 1 for unary, 2 for binary, 3 for ternary, 4 for quaternary, etc. For example, binary would be enumerated as ε₂, 0₂, 1₂, 00₂, 01₂, 10₂, etc.

- If we list all possible sentences of a language according to positional notation, we suffix the sentences with u for unary, b for binary, t for ternary, q for quaternary, V for quintary, VI for sextary, etc. For example, positional binary would be enumerated as ε₁b, 0₁b, 1₁b, 10₁b, 11₁b, etc.

- We note that for unary, both enumeration methods are identical hence the suffix u can be used interchangeably with 1. However, to my eye 1₁ reads a bit more confusing than 1u, therefore we will pick u for unary in this work.

- By convention, we will name positional decenary to be decimal and no suffix is used for its sentences. The sentences of decimal are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, etc.

Remark 1.14. Positional notation skips some sentences from its enumeration, notably those with leading zeros. More on that later in the section on multiplication.

If we were to construct a conventional axiomatic theory of knowledge, we would pose our first axioms at this stage. A favorite is usually the axiom of existence taking the paraphrased form of: "We pose that a simple language exists, such as the binary language". Then using this axiom and possibly other axioms we would derive theorems, such as the existence of other languages, etc. For example, in a certain formulation of set theory, the axiom of the empty set takes the following form:
Remark 1.15 (Axiom of the empty set in set theory). In some formulation of set theory, an axiom is introduced to obtain a first set.

\[ \exists x \forall y (\neg (y \in x)) \]

In other formulations of set theory, the axiom of subsets, or the axiom of an infinite set takes the place of the generator of the first set.

Here however, we are not planning to describe sets or numbers but language. Since language is required to describe any formal system of logic, its existence is guaranteed (within logic). In the theory of knowledge that we will introduce, all initial statements of the theory will be provable as primitive theorems. Hence no axioms will be used in the making of this theory.

This is a unique property of using language to describe language. It cannot be used with any other abstract notions such as with sets. The sets of set theory although describable via language are nonetheless too abstract for language to provide a proof of their existence by construction thereof. Hence, set theory must be derived from axioms taken as true.

Definition 1.16 (Laws of unrestricted semantics).

\[ \exists x [\text{isSymbol}(x)] \] (Existence of a symbol)

\[ \exists x \exists y [\text{isSymbol}(x) \land \text{isSymbol}(y) \land \text{areDistinguishable}(x, y)] \] (Existence of another symbol)

\[ \exists p [\text{isSentence}(p)] \] (Existence of a sentence)

\[ \forall x [\text{isSymbol}(x) \rightarrow \exists e [\text{isSentence}(e) \land (ex = x)]] \] (Existence of the empty sentence)

\[ \forall p [\text{isSentence}(p) \rightarrow \exists x [\text{isSymbol}(x) \land (x := p)]] \] (Unrestricted definitions)

Equality of symbols

\[ \forall x [\text{isSymbol}(x) \rightarrow (x = x)] \] (Reflexivity)

\[ \forall x \forall y [\text{isSymbol}(x) \land \text{isSymbol}(y) \rightarrow (x = y \rightarrow y = x)] \] (Symmetry)

\[ \forall x \forall y \forall z [\text{isSymbol}(x) \land \text{isSymbol}(y) \land \text{isSymbol}(z) \rightarrow (((x = z) \land (x = y)) \rightarrow y = z)] \] (Transitivity)

Equality of sentences

\[ \forall p [\text{isSentence}(p) \rightarrow (p = p)] \] (Reflexivity)

Rule to construct well-formed sentences

\[ \forall x \forall p [(\text{isSymbol}(x) \land \text{isSentence}(p)) \rightarrow \text{isSentence}(px)] \] (Concatenation)

Remark 1.17. First order logic is used only to write down the above laws in a clear and unambiguous manner. We do not use the full facilities of first-order logic to prove theorems.
Remark 1.18. The only proof method that we accept is the simplest proof method possible: direct construction of the object the existence of which we want to prove.

Remark 1.19. The law of unrestricted definitions is to be interpreted in a similar manner as the deduction rule of first order logic. Once specific definitions are posed, the sentences that are provable as a result of those definitions are only so within the definitions posed. We note the following special case.

Definition 1.20 (Theorem). If a model $M$ of an axiomatic theory $T$ is constructively shown to exist within the laws of unrestricted semantics but necessitate an appeal to the law of unrestricted definitions to be constructed, then $∃T : [T ⊢ ∃M]$ is an primitive theorem of unrestricted semantics, but $∃M$ is not. In this case, we say that $∃M$ is a theorem necessitating $T$.

In what follows, we will prove the laws of unrestricted semantics as primitive theorems.

**Primitive Theorem 1.21** (Existence of a symbol).

$$∃x[isSymbol(x)]$$

*Proof.* We offer a proof by construction.

- 1

**Primitive Theorem 1.22** (Existence of another symbol).

$$∃x∃y[isSymbol(x) ∧ isSymbol(y) ∧ areDistinguishable(x, y)]$$

*Proof.* We offer a proof by construction.

- 0

**Primitive Theorem 1.23** (Existence of a sentence).

$$∃p[isSentence(p)]$$

*Proof.* We offer a proof by construction.

- 1

**Primitive Theorem 1.24** (Equality of symbols).

$$∀x[isSymbol(x) → (x = x)] \quad (1.25)$$

$$∀x∀y[isSymbol(x) ∧ isSymbol(y) → (x = y → y = x)] \quad (1.26)$$

$$∀x∀y∀z[isSymbol(x) ∧ isSymbol(y) ∧ isSymbol(z) → [(x = z) ∧ (x = y)] → y = z] \quad (1.27)$$
Proof. Equality is definable from second order logic by the following principles of Leibniz:

\[ \forall x \forall y [x = y \rightarrow \forall P(Px \leftrightarrow Py)] \]  \quad \text{(Indiscernibility of identicals)}

\[ \forall x \forall y [\forall P(Px \leftrightarrow Py) \rightarrow x = y] \]  \quad \text{(Identity of indiscernibles)}

From the primitive notion 1.7 of the symbol - a unique distinct identifier, there exists two and only two predicates available to describe a symbol. The first one is \( \text{isSymbol}(x) \) and it is true if and only if \( x \) is a symbol. The second is \( \text{areDistinguishable}(x, y) \) and it is true if and only if \( x \) is distinct from \( y \).

For any symbol and from these two predicates, it is clear that the Leibniz principles are respected when we use the same symbol twice. As reflexivity, symmetry and transitivity of the equality are derivable from the Leibniz principles, we have proven the theorem. \( \square \)

Remark 1.28. The fact that \( \text{areDistinguishable}(x, y) \) is a "soft" converse of the notion of equality does not invalidate the proof. The predicate \( \text{areDistinguishable}(x, y) \) has the same value \( \forall x \forall y(x = y) \) therefore it does respect the Leibniz principles and the proof is valid.

Primitive Theorem 1.29 (Existence of the empty sentence).

\[ \forall x [\text{isSymbol}(x) \rightarrow \exists \epsilon [\text{isSentence}(\epsilon) \land (\epsilon x = x)]] \]  \quad (1.30)

Proof. We offer a blank statement as the proof. For convenience, we use the symbol \( \epsilon \) to represent it.

\( \epsilon \)

Since \( \epsilon \) is defined as a blank statement it follows that replacing it with its definition yields \( \epsilon x \leftrightarrow x \), hence \( \epsilon x = x \).

Primitive Theorem 1.31 (Unrestricted definitions).

\[ \forall p [\text{isSentence}(p) \rightarrow \exists x [\text{isSymbol}(x) \land (x := p)]] \]

Proof. We offer a proof by construction. Suppose a language \( L \) with \( n \) symbols \( s_1, s_2, \ldots, s_n \), where \( n \) is finite. We can construct every possible sentence of the language from shortest to longest in alphabetical order as shown in table 1.

<table>
<thead>
<tr>
<th>( \epsilon )</th>
<th>( s_1 )</th>
<th>( s_2 )</th>
<th>( \ldots )</th>
<th>( s_n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_1s_1 )</td>
<td>( s_1s_2 )</td>
<td>( \ldots )</td>
<td>( s_1s_n )</td>
<td>( s_2s_1 )</td>
</tr>
<tr>
<td>( s_2s_2 )</td>
<td>( \ldots )</td>
<td>( s_2s_n )</td>
<td>( s_1s_1s_1 )</td>
<td></td>
</tr>
<tr>
<td>( \ldots )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To each of the sentences herein constructed, we associate a unique symbol. For example, we can associate symbols to sentences as such:

\[ \Delta := \epsilon \]
\[ \Box := s_1 \]
\[ \emptyset := s_2 \]
\[ \ldots \]

Table 1: We construct all possible sentences of \( L \) as shown here.
For each newly constructed sentence we associate a symbol different than the previous one. Without loss of generality, this can be done by using the regular polygons starting from the triangle and each time adding an additional side to the shape. We never run out of symbols and we associate every sentences to a unique symbol.

**Axiomless Theorem 1.32** (Reflexive equality of sentences).

\[ \forall p [\text{isSentence}(p) \rightarrow (p = p)] \]

**Proof.**

\[
\begin{align*}
  x &= x & \quad & \text{(Reflexive equality of symbols)} \\
  x &:= p & \quad & \text{(Unrestricted definitions)} \\
  (x = x) &\leftrightarrow (p = p) & \quad & \text{(Replacement of \(x\) by its definition)} \\
  p &= p & \quad & (x = x \text{ is always true})
\end{align*}
\]

We will now prove more advanced theorems from the basic laws.

**Primitive Theorem 1.33.** There exist the unary language

**Proof.** As a proof by construction, take the language generated by the following alphabetical sentence

1

**Definition 1.34** (Unary language). *The unary language is defined by the alphabetical sentence of one symbol: 1. Some examples of the sentences of unary are: \(\epsilon, 1_u, 11_u, 111_u, \ldots\). The subscript \(u\) (for unary) is optional but it is added in this work to avoid confusion with other usages of the symbol 1.

Other languages can be defined (and proved) in a similar manner. For example, binary.

**Primitive Theorem 1.35.** There exists the binary language

**Proof.** As a proof by construction, take the language generated by the following alphabetical sentence

01

**Definition 1.36** (Binary language). *The binary language is defined by the alphabetical sentence of two symbols: 01. Some examples of the sentences of binary are: \(\epsilon, 0_b, 1_b, 00_b, 01_b, 10_b, 11_b, 000_b, \ldots\). The subscript \(b\) (for binary) is optional but it is added in this work to avoid confusion with other usages of the symbols 0 and 1.*
Primitive Theorem 1.37. There exists the natural numbers

Proof. The natural numbers are defined by recursion against the starting element, such that the successor of any natural number is also a natural number. To recover a model using the unary language, we

1. associate the empty sentence with the starting element,
2. associate each successively longer sentence with the next element.

To translate between different notations of natural numbers such as from unary to decimal we can pose the following definitions

\[
\begin{align*}
\triangle &:= \epsilon_u \\
\box &:= 1_u \\
\circ &:= 11_u \\
\circ &:= 111_u \\
& \vdots
\end{align*}
\]

\[
\begin{align*}
\triangle &:= 0 \\
\box &:= 1 \\
\circ &:= 2 \\
\circ &:= 3 \\
& \vdots
\end{align*}
\]

\[
\begin{align*}
0 &\Rightarrow 0 = \epsilon_u \\
1 &\Rightarrow 1 = 1_u \\
2 &\Rightarrow 2 = 11_u \\
3 &\Rightarrow 3 = 111_u \\
& \vdots
\end{align*}
\]

Definition 1.38 (Alphabetical position of a sentence). We repeat theorem 1.37 for other languages. To every sentence of a language we associate a natural number in the following way:

1. The number 0 is the empty sentence \(\epsilon\).
2. The number 1 is the alphabetical first symbol.
3. If there are more symbols, then the next number is associated to the alphabetical second symbol. Otherwise, it is associated to the sentence with two symbols.
4. And so on.

Each natural number is associated to a corresponding sentence from shortest to longest and from alphabetical first to alphabetical last.

Table 2 shows the alphabetical enumeration of select languages along with their alphabetical position indicated on the left in decimal.

1.5 Arithmetic

Definition 1.39 (Disjoin). A disjoin is made by inserting a period “.” between two symbols of a sentence.
To avoid confusion with the usage of the period in normal text, sentences that are disjoined will be enclosed in brackets when used in normal text. Example: \([111.11_u]\).

**Definition 1.40 (Join).** A join is made by removing a period "." between two symbols of a sentence.

Since the period symbol "." is not a symbol of the language of the sentences that are being split, the presence or the absence of the symbol does not affect the equality. For example, \([111.111_u]\) is concatenated to \([1111111_u]\), and \([111.1111_u] = [1111111_u]\). The "." symbol should be seen in a manner similar to a comment in computer code. Although it does not have any impact on the laws of unrestricted language, it will nonetheless significantly improve the readability of the coming proofs.

**Primitive Theorem 1.41.** Joining any two sentences produces another sentence.

**Proof.** The proof will be offered for the unary language, but it is trivial to generalize it to other languages. The unary suffix \(_u\) will be omitted.

We now list the possible joins of the unary language.

\[
\begin{bmatrix}
\epsilon.\epsilon = \epsilon & \epsilon.1 = 1 & \epsilon.11 = 11 & \ldots \\
1.\epsilon = 1 & 1.1 = 11 & 1.11 = 111 & \ldots \\
11.\epsilon = 11 & 11.1 = 111 & 11.11 = 1111 & \ldots \\
\vdots & \vdots & \vdots & \ddots
\end{bmatrix}
\]

The period "." behaves like comments in code hence are erasable from the equalities. Let us see an example of when they are removed.

---

Table 2: A natural number is associated to each sentence of select languages. Note that the the natural number 0 is associated to \(\epsilon\) and not \(1_u\), \(0_b\), \(0_t\) or \(0_d\). Therefore the decenary sentences are shifted by one with respect to the decimals.
111 = 11.1\epsilon
111 = 111\epsilon \quad \text{(removing .)}
111 = 111 \quad \text{(removing \(\epsilon\))}

Disjoining a sentence with the period, or by inserting \(\epsilon\), has no impact on the equality. \(\square\)

**Primitive Theorem 1.42.** *The joins over unary are commutative.*

*Proof.* It is easy to see in the table for the proof of theorem 1.41, each element of the table has a corresponding commuted and equal join. For example, both \([1.\epsilon_u] = [1_u]\) and \([\epsilon.1_u] = [1_u]\) are present. This is the case for all terms. Furthermore, the diagonal terms are equal to their commutation. \(\square\)

**Primitive Theorem 1.43.** *The joins over the unary is addition, where the symbol \(+\) is replaced by the disjoin symbol "," (period).*

*Proof.* The Peano axioms of addition are;

\[
a + 0 = a \quad \text{(1.44)}
\]

\[
a + S(b) = S(a + b) \quad \text{(1.45)}
\]

**Lemma 1.46.**

\[a + 0 = a\]

*Proof.* Without loss of generality, we pose \(a = 1\).

\[
1 + 0 = 1_u + \epsilon_u \quad \text{(decimal to unary)}
\]

\[
= 1_u.\epsilon_u \quad \text{(definition of addition)}
\]

\[
= 1\epsilon_u \quad \text{(join)}
\]

\[
= 1_u \quad \text{(elimination of \(\epsilon\))}
\]

\[
= 1 \quad \text{(unary to decimal)}
\]

\(\square\)

**Lemma 1.47.**

\[a + S(b) = S(a + b)\]

*Proof.* Without loss of generality, we pose \(a = 1\) and \(b = 2\) and \(S(b) = 3\).

\[
1 + 3 = 1_u + 111_u \quad \text{(decimal to unary)}
\]

\[
= 1.111_u \quad \text{(definition of addition)}
\]

\[
= 1111_u \quad \text{(join)}
\]

\[
= 4 = S(1 + 3) \quad \text{(unary to decimal)}
\]

\(\square\)
1.6 Translation

Definition 1.48 (Translation sentence). A translation sentence is a binary sentence that can be used to arbitrarily associate each sentence of a language to a matching sentence in another language.

Primitive Theorem 1.49. There exists a translation sentence.

Proof. Take any binary sentence $S_2$. This is how $S_2$ can be interpreted as a translation sentence from $L_1$ to $L_2$.

We disjoin the sentence after any occurrence of the symbol 0.
Each segment is unitary encoded. For example, suppose the $S_2$ is 1011011100.... The disjoined segments are $\{10, 110, 1110, 0, \ldots\}$. Each segment can be made to correspond to a natural number such that:

\[
\begin{align*}
0 &:= 0 \\
1 &:= 10 \\
2 &:= 110 \\
3 &:= 1110 \\
4 &:= 11110 \\
& \vdots
\end{align*}
\]

We then interpret the translation sentence as follows: the first unitary segment of $S_2$ associates the first sentence of $L_1$ to the $n^{th}$ sentence of $L_2$, where $n$ is the natural number associated with the unitary segment, and so on.

Using this construction, we can arbitrarily translate any sentence of $L_1$ to $L_2$.

Definition 1.50 (Encoding). An encoding is a bijective translation sentence to and from the same language $L$.

Definition 1.51 (Unitary encoding). Unitary encoding maps every sentence of binary to the following sentences of binary

\[\{0_2, 10_2, 110_2, 1110_2, 11110_2, \ldots\}\]

The translation sentence from binary to unitary encoding is

\[1011111011111111110\ldots\quad (1.52)\]

Theorem 1.53. All languages can be encoded in unitary.

Proof. Using an appropriate translation sentence, first translate any language $L$ to binary. Then using the unitary translation sentence (1.52), translate a second time from the binary language to the unitary encoding.
1.7 Multiplication

We can prove the existence of a model $t$ for an axiomatic theory $k$ within unrestricted semantics if we can produce such a model using the law of unrestricted definitions. This will now be done for multiplication. We then obtain a primitive theorem of the form $\exists k[(k \in L) \land (k \vdash t)]$ which has a similar interpretation as the deduction rule in first order logic.

For the purposes of multiplication, the alphabetical enumeration shown in (Table 2) is of no help here. We must introduce a restricted enumeration shown in (Table 3) which removes certain sentences from the list. We will call this enumeration the positional enumeration. It should not come as a surprise that some sentences must be removed from the enumeration as we introduce multiplication because, as is well-known, any theory of arithmetic with multiplication must be incomplete (Gödel).

There are two main culprits responsible for the need for this restricted enumeration.

1. The first one is the multiplication by 1, or in binary: $0_b$. To multiply 1 as a join, we obtain $0_b \times 1_b = 0.1_b = 01_b$. But this must also be equal to $1_b$, because 1 is the identity of multiplication e.g. $(01_b = 1_b$ under multiplication). As a workaround we define 1 as $\epsilon$ instead of as 0 in Table 3 for languages of 2 or more symbols.

2. The other culprit is that the concatenation associated to a multiplication is similar to the bit-shift operation of binary numbers producing a multiplication by 2. For the bit-shift operation to be connected to a multiplication by 2, we require that the natural numbers must be associated to binary sentences under the rule that $001_b = 01_b = 1_b$, etc.

**Definition 1.54 (Multiplication).** We define the multiplication of two natural numbers as a join of two sentences of a language enumerated according to table 3, such that;

- The multiplication of any $n \in \mathbb{N}^+$ by 2 is the join of $n$ in binary with $1_b$.
- The multiplication of any $n \in \mathbb{N}^+$ by 3 is the join of $n$ in ternary with 2$t$.
- The multiplication of any $n \in \mathbb{N}^+$ by 4 is the join of $n$ in quaternary with 3$q$.
- The multiplication of any $n \in \mathbb{N}^+$ by 5 is the join of $n$ in quinary with 4$q$.
- etc.
Already we recover all possible multiplication operations for all $n > 1$. Binary, ternary, quinary, septenary, etc. are the prime languages. We are left with the need to define the multiplication by and of 0 and multiplication by 1 as special cases.

We start with multiplication by and of 0.

- The multiplication of any $n \in \mathbb{N}$ by 0 is the join of $n$ in nullary with $e$.
- The multiplication of 0 by any $n \in \mathbb{N}$ is the join of $e$ with $n$ in nullary.

Now we define the multiplication of 1. The unary language is already defined for arithmetic. We therefore cannot use it for the multiplication by 1. Instead we define multiplication by 1 as such

- The multiplication of any $n \in \mathbb{N}^+$ by 1 is the join of 1 in $n$-ary with the alphabetical last symbol of $n$-ary.

All other joins are undefined under multiplication.

Using these operations, we can prove their existence by writing a sentence of a language that has the required properties. For example, the sentence $[10.2_t] = [102_t]$ is the multiplication of 4 by 3.

\[
4 \times 3 = 4 \times 3 \\
= 10_t \times 3_t \quad \text{(conversion from decimal to ternary)} \\
= 10.2_t \quad \text{(definition of multiplication)} \\
= 102_t \quad \text{(join)} \\
4 \times 3 = 12 \quad \text{(conversion from ternary to decimal)}
\]

Commutation is respected.
\[
3 \times 4 = 3 \times 4 \\
= 10_b \times 11_b \quad \text{(conversion from decimal to binary)} \\
= 10.11_b \quad \text{(definition of multiplication)} \\
= 1011_b \quad \text{(join)} \\
3 \times 4 = 12 \quad \text{(conversion from binary to decimal)}
\]

**Theorem 1.55.** The join over Table 3 as defined in 1.54 is the arithmetic multiplication.

*Proof.* The Peano axioms of multiplication are:

\[
a \times 0 = 0 \\
a \times S(b) = a + (a \times b)
\]

**Lemma 1.56.**

\[
a \times 0 = 0
\]

*Proof.* Without loss of generality, we pose \(a = 1\).

\[
a \times 0 = 1 \times 0 \\
= \epsilon_n \times \epsilon_n \quad \text{(conversion from decimal to nullary)} \\
= \epsilon \cdot \epsilon_n \quad \text{(definition of multiplication)} \\
= \epsilon \epsilon_n \quad \text{(join)} \\
= \epsilon_n \quad \text{(elimination of redundant } \epsilon \text{)} \\
= 0 \quad \text{(conversion from nullary to decimal)}
\]

**Lemma 1.57.**

\[
a \times S(b) = a + (a \times b)
\]

*Proof.* Without loss of generality, we pose \(a = 1\), \(b = 2\) and \(S(b) = 3\).

\[
a \times S(b) = 1 \times 3 \\
= \epsilon \times 2_t \quad \text{(conversion decimal to ternary)} \\
= \epsilon.2_t \quad \text{(definition of multiplication by 3)} \\
= \epsilon 2_t \quad \text{(join)} \\
= 2_t \quad \text{(elimination of } \epsilon \text{)} \\
1 \times 3 = 3 \quad \text{(conversion from ternary to decimal)}
\]

It is a theorem that, in the context of an appropriate set of unrestricted definitions, multiplication is proven to exist.
1.8 Set theory

We can repeat the process used for multiplication but now for set theory. We first find a context for unrestricted definitions which produces a model for set theory. We will only offer a sketch here.

**Sketch 1.58.** Set theory

**Lemma 1.59.** Equality

\[ \forall x \forall y [\forall z (z \in x \leftrightarrow z \in y) \implies x = y] \]

*Proof.* We can create a set $S$ of $n$ elements by taking the alphabetical sentence of a language with $n$ symbols, then posing all permutation of the alphabetical sentence to be equal to $S$. Taking as an example the case where $n = 3$;

\[
S := 012 \\
:= 021 \\
:= 102 \\
:= 120 \\
:= 201 \\
:= 210
\]

We can iterate over the different formulations of the same set switching to $S$ and back, such as

\[ 012 = S = 102 \]

**Lemma 1.60.** Subset

*Proof.* Subsets are simply the possible joins and disjoins of $S$, where we replace the "," with \(

\[
S := 01.2 = 01 \cup 2 = 0.12 = 0 \cup 12 \\
:= 02.1 = 02 \cup 1 = 0.21 = 0 \cup 21 \\
:= 10.2 = 10 \cup 2 = 1.02 = 1 \cup 02 \\
:= 12.0 = 12 \cup 0 = 1.20 = 1 \cup 20 \\
:= 20.1 = 20 \cup 1 = 2.01 = 2 \cup 01 \\
:= 21.0 = 21 \cup 0 = 2.10 = 2 \cup 10
\]

\[ \square \]
A question we might have at this point is, can the laws of unrestricted semantics "talk" about any theory? We will see that the answer is yes. Have we got ourselves something too general to be useful? In the next two sections, we will see that this is not the case at all.

1.9 Representation of any theory of knowledge

We define two languages Q and A. Without loss of generality, suppose that the languages are binary.

\[
\begin{array}{ll}
Q & A \\
q_1 := \varepsilon_2 & a_1 := \varepsilon_2 \\
q_2 := 0_2 & a_2 := 0_2 \\
q_3 := 1_2 & a_3 := 1_2 \\
q_4 := 00_2 & a_3 := 00_2 \\
& \vdots \\
\end{array}
\]

Although these languages look identical, no equality for languages has been defined in the laws of unrestricted semantics. If a definition of equality for languages is desired, then it must be derived from the law of unrestricted definitions. From Q, A and this law, we can define theories of knowledge.

**Definition 1.61** (Theory of knowledge). We create a theory of knowledge by associating to every \( q_i \) either an \( a_j \), or we mark it as undefined \( \exists \).

**Remark 1.62.** Although for languages with the same number of symbols we can pose the trivial case \( \forall n (q_n := a_n) \), it gets more interesting for other cases.

**Primitive Theorem 1.63.** Each \( q_i := a_j \) definition that we pose is a claim that there exists a theory of knowledge \( k \) composed of a group of axioms such that for all defined \( q_i \) and its associated answer \( a_j \), this holds

\[
\exists k : [\forall q_i (k \vdash (q_i = a_j))]
\]

**Proof.** It is insightful when we can compress \( k \) in such a way that very few axioms prove all associations. For the proof of this theorem however, it suffices to say that each \( q_i \) to \( a_j \) association could itself be an axiom of \( k \).

**Definition 1.64** (Q2A-sentence). A Q2A-sentence is a translation sentence in binary that maps each \( q_i \) to an \( a_j \). Every binary string can be interpreted as a theory of knowledge.
A Q2A-sentence contains all the knowledge otherwise obtainable from the axiomatic representation of the theory. Taking the analogy where Q represents the question and A is the answer, the Q2A-sentence encodes the answer to every question of the language.

**Remark 1.65.** Each Q2A-sentence is a binary sentence. Since we identify sentences by assigning them a natural number (alphabetical position), we can only identify countably infinitely many sentences. However, there are uncountably infinitely many theories of knowledge. Neglecting to include these theories will however not be a problem. Indeed, being able to express countably infinitely many theories of knowledge is enough to recover all of those which can be expressed with recursively enumerable axioms or better. The ones which are left out are “random junk”.

**Remark 1.66.** If a specific $q_i$ would be mapped to an infinitely long $a_j$, we would claim that $q_i$ is undefined rather than map to it. This ensures that each unitary encoded segment of the Q2A-sentence is finite and well behaved.

**Remark 1.67.** We reserve the first unitary segment $0_2$ to correspond to the undefined state such that any $q_i$ mapped by it is undefined.

As an example, consider the following binary sentence

\[1101001001110110\ldots\]

Its unitary encoded segments are

\[110, 10, 0, 10, 0, 0, 1110, 110,\ldots\]

From this sentence, Q is mapped to A as

\[
q_1 := a_2 \\
q_2 := a_1 \\
q_3 := \not a \\
q_4 := a_1 \\
q_5 := \not a \\
q_6 := a_3 \\
q_7 := a_2 \\
\ldots
\]

**Definition 1.68 (UTK).** A universal theory of knowledge (UTK) is a theory of knowledge $k$ which embeds every recursively enumerable theory of knowledge. By embedding, we mean that the following first order sentence holds,
∀k((k ⊢ t) → (∃k′((k′ ∈ L) ∧ (k′ ⊢ t))))

In plain English, it means that for all theories of knowledge \( k \), if \( k \) proves a theorem \( t \), then there must exist an alternative formulation of \( k \) within the language \( L \) of the UTK such that \( k' \) proves \( t \).

**Primitive Theorem 1.69.** Unrestricted semantics is a UTK.

**Proof.** This is a trivial consequence of the existence of arbitrary Q2A-sentences.

**Theorem 1.70.** First order arithmetic is a UTK.

**Proof.** Consistent with the Gödel numbering method and for each well formed formula of arithmetic, we associate to it a natural number. Then, for each recursively enumerable theory of knowledge \( k \) we pose a series of equations to be used on the natural numbers. The equations are posed such that they transform the natural number to another natural number in a manner consistent with the rules of inference of \( k \).

This can be repeated for all possible recursively enumerable axiomatic theories of knowledge. As a result, arithmetic is a UTK.

2 The Theory of Everything

Primitive theorems are provable only by direct construction of the object the existence of which we want to prove or by invoking other primitive theorems. As a result and in a certain sense, the pen and the paper we use to write becomes a laboratory where that which we can construct can be used as evidence of its existence. Of course writing about pink elephants does not prove their existence. But writing very carefully about language using language does. Indeed and as shown, in a properly formulated pure theory of language (e.g. the laws of unrestricted semantics) this "exception" is sufficient to prove all of its axiom-like laws as primitive theorems.

One might ask, "but are you not assuming the existence of language. Therefore the whole theory rests on this assumption making it axiomatic?". To which I would answer that since a language is required to formulate that question, the question is self-defeating. In fact any written, spoken or sign-based counter argument to the existence of language would have to use language and would be self-defeating. Hence, the existence of language is immune against all language-based counter-arguments.

As a result a theory which only concerns itself with language (and no other abstract entities) is a special case of logic for which an axiomless derivation is possible.
2.1 Elimination of physically impossible theories

In this work, our goal is to find the theory of everything in physics. So far, we have obtained a way to list all possible theories of knowledge that can be formulated in a finite language. Since they are all in the list, then necessarily, one of them is the theory of everything describing our universe. What we need now is an argument to eliminate all non-ToEs from consideration.

Since this work is done within the universe we are trying to explain, it must be the case that the ToE explaining the universe cannot contradict the existence of that which we primitively prove in it. Stated differently, any primitive theorem of unrestricted semantics must be theorems of the theory of everything.

For example, if we prove the existence of a symbol by writing 1, then it cannot be the case that $\neg \exists x (isSymbol(x) \land (x = 1))$ is a theorem of the theory of everything. As a result we can eliminate all of the theories which would produce such a theorem from being ToE candidates on the grounds that they contradict the primitive knowledge herein derived. The more ToE-candidate we filter out using primitive knowledge, the closer we get to the real ToE.

What if we use absolutely every available primitive theorems, of which there are infinitely many, to filter out as many ToE-candidates as possible? We will see that we exactly recover a condition which is both necessary and sufficient for a ToE. It is in fact enough by itself to recover the physics, all of the physics and nothing but the physics.

**Primitive Theorem 2.1.** The Theory of Everything must be a universal theory of knowledge.

*Proof.* If the ToE is not a UTK, then a mathematician cannot do his job. Indeed, using a pen and a paper a mathematician is free to pose any axiomatic theory $k$ and take it to its logical conclusions. Therefore, all theorems of the form

$$\forall k[(k \vdash t) \rightarrow (\exists k'[k' \in L) \land (k' \vdash t))]$$

must be theorems of the theory of everything, or the ToE would contradict that which can provably be done with a pen and paper. Since mathematicians presumably reside in the universe governed by this ToE, and mathematicians can in principle recover any theorem provable within a UTK, then the ToE must be a UTK. □

All we have to do now is show that the laws of physics are a necessary consequence of any UTK. Then we have an axiomless derivation of the ToE.
2.2 The universal theory of knowledge

To make the laws of physics come out of the UTK, we will reformulate the UTK within the framework of a universal Turing machine (UTM). Doing this will unlock the formalism of algorithmic information theory and other UTM-related theorems to help us out.

To achieve this, recall definition 1.64 on the representation of a theory of knowledge from a $Q_A$-sentence of the binary language. In this scenario, the $Q_A$-sentence is interpreted as a translation sentence from language $Q$ to language $A$ such that each translated sentence poses a series of definitions, each giving an answer to a question (or leaving it undefined).

The $Q_A$-sentence representing an axiomatic theory does in fact contain all the information of the theory. If by analogy we suppose that $Q$ stands for the question and $A$ stands for the answer, we can find the answer to question $n$ by looking up the $n^{th}$ translation segment. Knowledge of the specific axiomatic theory $T$ is not required as a compatible formulation of it is necessarily recoverable.

If, however, we do have knowledge of $T$, we can simplify the $Q_A$-sentence greatly to a new sentence. This new sentence will be called the $\Omega$-sentence of the theory. This sentence will be how we will recover a link to the UTM formulation.

**Definition 2.2 ($\Omega$-sentence).** An $\Omega$-sentence is a binary sentence where $b(i)$, the $i^{th}$ bit of the sentence, corresponds to $q_i$ such that:

$$b(i) = \begin{cases} 0 & q_i \text{ is undefined} \\ 1 & \text{otherwise} \end{cases}$$

**Theorem 2.3.** The $\Omega$-sentence together with the axiomatic formulation of the theory $T$ are sufficient to correctly recover the $Q_A$-sentence in finite time.

**Proof.** To see how, first consider that we obtain the definition of a UTM by performing the following replacements:

- $T$ is the program.
- $q_i$ is the input.
- $a_j$ is the output.

Then for all identifiable $T$, there exists an encoding such that a UTM can read $T$ and $q_i$ as the input and produce $a_j$ as the output. Furthermore, knowing the $\Omega$-sentence, the UTM can skip over non-halting problems. This allows it to recover any number ($< \infty$) of
leading bits of the Q2A-sentence in finite time. As a result, the Q2A-sentence is computable from a UTM having knowledge of the Ω-sentence.

Remark 2.4. It is much faster to find answers with the Q2A-sentence than it is from the Ω-sentence. Indeed,

- Finding the answer to $q_i$ using the Q2A-sentence is a simple lookup of the $i$th unitary encoded segment of the sentence. This segment points to the answer $a_j$. This is a relatively fast search algorithm.

- Using the Ω-sentence involves considerably more effort. First we find the bit corresponding to the question we are interested in. If it is 0, we know there is no answer and we can stop immediately. If instead it is 1, we begin the search. To find the answer, it is necessary to first try to find a proof, in dovetail, that the answer is $a_1$, then $a_2$, then $a_3$ and so on until we find it. We know that eventually we will find an answer because its Ω-sentence bit is 1 hence the answer exists.

Here we assume that speed is not relevant. If we are interested only in logical completeness then both sentences offer that. We will use the Ω-sentence formulation as it is more convenient for the theorems we will later prove.

2.3 The halting probability

We will now select first-order arithmetic as our UTK of choice in order to derive the laws of physics from. To facilitate the mathematical treatment, we will introduce a formalism.

Definition 2.5. We define $S(i)$ as the $i$th sentence of the alphabetical enumeration of language $L$.

We know that some sentences are provable, but we do not know which ones. To represent this formally, we will define a two-state identification function $T(i)$ as

Definition 2.6. For each sentence $S(i)$

$$T(i) = \begin{cases} 0 & S(i) \text{ is a theorem} \\ \infty & S(i) \text{ is not a theorem} \end{cases}$$

The reason for the choice of $\infty$ or 0 as the states, as opposed to say 0 and 1, will become clear when we start to write equations in the form of a sum.

Theorem 2.7. If we take the number $i$ corresponding to each sentence and weigh it according to a power probability distribution in base 2, we can
write
\[ 1 = \sum_{i=1}^{\infty} 2^{-i} \] (2.8)

**Proof.** Expanding the sum in binary, we get
\[
\sum_{i=1}^{\infty} 2^{-i} = 2^{-1} + 2^{-2} + 2^{-3} + ...
\]
\[ = 0.1_2 + 0.01_2 + 0.001_2 + 0.0001_2 + ...
\]
\[ = 0.1_2
\]
\[ = 1_2
\]
\[ \square \]

**Theorem 2.9.** Further refining this equation we add the term \( T(i) \) to capture only the theorems while excluding all other sentences from the sum,
\[ \Omega = \sum_{i=1}^{\infty} 2^{-T(i)} 2^{-i} \]

**Proof.** Again expanding the sum into binary, we get
\[
\Omega = \sum_{i=1}^{\infty} 2^{-T(i)} 2^{-i}
\]
\[ = 2^{-\infty}2^{-1} + 2^{-0}2^{-2} + 2^{-0}2^{-3} + 2^{-0}2^{-4} + 2^{-\infty}2^{-5} + ...
\]
\[ = 0_2 + 0.01_2 + 0.001_2 + 0.0001_2 + 0_2 + ...
\]
\[ = 0.01110..._2
\]

We obtain a number \( \Omega \) where its bits are in a one-to-one correspondence with the sentences given by \( S(i) \). If \( S(i) \) is a theorem of \( L \), the \( i \)th bit of \( \Omega \) is 1, otherwise it is 0. The reason why \( T(i) \) uses 0 and \( \infty \) as its two states is now clear. It is to remove the terms that are not theorems from the sum by making them vanish to 0. \[ \square \]

This definition of \( \Omega \) is equivalent to the Gregory Chaitin’s \( \Omega \) number construction of algorithmic information theory associated with the halting probability of a prefix-free universal Turing machine.

**Remark 2.10 (The Universe).** The number \( \Omega \) represents all the knowledge obtainable for a language \( L \) according to some rules of inference. If we define the universe by its knowledge content, there is a \( \Omega_U \) corresponding to our universe. This definition will be made explicit in the next sections as we consider that it is more convenient to represent the universe by its knowledge content, rather than it is by its rules of inference.
2.4. *The universal Turing machine*

Switching to the universal Turing machine at this stage gives us the benefit of unlocking the formalism of algorithmic information theory. Since we are now working with Turing machines it is more appropriate to use the word ‘program’ instead of ‘sentence’, and to use the word ‘halts’ instead of ‘is a theorem’. For example, instead of saying ‘this sentence is a theorem’, we will say ‘this program halts’. It is equivalent but more appropriate when referring to Turing machines.

**Theorem 2.11.** $\Omega$, if calculated from a UTM, is the halting probability of a prefix-free UTM.

*Proof.* The proof will be divided into 2 lemmas.

**Lemma 2.12.** $\Omega$ is a probability therefore $0 \leq \Omega \leq 1$.

*Proof.* If all programs halt, $\Omega = 0.1 = 1$. And if no program halts, $\Omega = 0$. Since the possible values are a real number between 0 and 1, it meets the definition of a probability.

**Lemma 2.13.** The program encoding is prefix free

*Proof.* In the case of unitary encoding, it is easy to see that no program is the prefix of another. A UTM reading unitary encoding will end at the first 0 it encounters and will assume this is the full program. Since the encoding enforces a single 0 bit per program at the very end, this guarantees that the UTM cannot mistake one program for another.

**Definition 2.14.** $\Omega$ is a Chaitin omega number (or halting probability). Each halting probability is a normal and transcendental real number that is not computable, which means that there is no algorithm to compute its digits. Indeed, each halting probability is Martin-Löf random, meaning there is not even any algorithm which can reliably guess its digits.

Why is $\Omega$ not computable? Because of the halting problem. Many sentences exist that do not contradict the rules of inference, but cannot be proven by them. When attempting to prove these sentences, the UTM will run forever without halting.

2.5 *The halting partition*

I have chosen to sum the sentences as an exponential distribution of base 2 in equation 2.8 because it is the most conceptually simple way to do it. It is not however the only way as other choices which preserve the information are available.

In this final subsection of this section, we generalize the halting probability to the halting partition. We note that key properties of the halting probability are preserved under the following generalization.

**Theorem 2.15.** A multiplication factor \( \beta \in [0, \infty] \) can be added to \( T(i) \) without changing the result of the sum

\[
Z = \sum_{i=1}^{\infty} 2^{-\beta T(i)} 2^{-i}
\]

*Proof.* expanding the sum into binary, we get

\[
Z = \sum_{i=1}^{\infty} 2^{-\beta T(i)} 2^{-i} \\
= 2^{-\beta \infty} 2^{-1} + 2^{-\beta 0} 2^{-2} + 2^{-\beta 0} 2^{-3} + 2^{-\beta 0} 2^{-4} + 2^{-\beta \infty} 2^{-5} + ...
\]

\[
= 0 + 0.01 + 0.001 + 0.0001 + 0 + ...
\]

\[
= 0.01110...
\]

The constant \( Z \) does not change because \( \beta \) is always multiplied by \( \infty \) or 0 in the sum, which erases its effects. \( \Box \)

**Theorem 2.16.** A multiplication factor, \( F \geq 1 \) can be added to \( i \) while preserving the non-computable nature of the sum, as well as the halting information content of each sentence

\[
Z = \sum_{i=1}^{\infty} 2^{-\beta[T(i)+Fi]}
\]

*Proof.* Tadaki has shown\(^3\) that Gregory Chaitin’s constant can be extended to include a compression term \( F \) on \( 2^{-i} \implies 2^{-Fi} \) such that the Tadaki constant \( \Omega^F \) remains non-computable \( F \)-random. Furthermore, he goes to show that \( \Omega^F \)'s first \( n \) bits contain \( |n-F| \) halting bit. For example, take the case where \( F = 2 \), then expanding the sum into binary, we get

\[
Z = \sum_{i=1}^{\infty} 2^{-Fi} \\
= 2^{-2 \times 1} + 2^{-2 \times 2} + 2^{-2 \times 3} + 2^{-2 \times 4} + 2^{-2 \times 5} + ... \\
= 2^{-2} + 2^{-4} + 2^{-6} + 2^{-8} + 2^{-10} ... \\
= 0.01 + 0.0001 + 0.000001 + 0.00000001 + 0.0000000001 + ... \\
= 0.0101010101...
\]

The compression factor \( F \) ”decompresses” the information by inserting some 0 in between the bits. It does not erase data. For the full proof, refer to Tadaki’s paper. \( \Box \)

---

Part II

The Theory of Everything

We complete the halting partition by giving it an interpretation as the Theory of Everything in physics from the perspective of algorithmic information theory. The exponential terms are given the roles of macroscopic observables and are program output, program length, program runtime, program memory usage, etc. We complete the halting partition with other possible observables taking the form of a sum of conjugate variable to macroscopic observable pair as $\sum k \sigma_k C_k(i)$.

**Theorem 2.17.** The theory of everything in physics is given by the following equation;

$$Z = \sum_{i=1}^{\infty} e^{-\beta[T(i) + F + \sum \sigma_k C_k(i)]}$$

**Remark 2.18.** In the special case where $\beta = 1$, $F = 1$ and $-\sum \sigma_k C_k(i) = 0$, we recover the halting probability and $Z = \Omega$.

**Proof.** To show that this equation is the ToE, in part III we will derive from it the major equations of physics including but not limited to statistical physics, general relativity, quantum mechanics, the holographic principle, the speed of light as a maximal speed, and more. These derivations will be done from this single equation and with no appeal to physical observation - e.g. from pure reason. $\square$
Part III

Proof of the Theory of Everything

3 Thermodynamics

It has been said that thermodynamics is the most general of all the disciplines of physics. Hence it is expected to be the first derived from a ToE.

A theory is the more impressive the greater the simplicity of its premises, the more different kinds of things it relates, and the more extended its area of applicability. Therefore the deep impression that classical thermodynamics made upon me. It is the only physical theory of universal content which I am convinced will never be overthrown, within the framework of applicability of its basic concepts.

–Albert Einstein

Introduction. In statistical physics, we are interested in the distribution that maximizes entropy

\[ S = - \sum_{x \in X} p(x) \ln p(x) \]  

subject to the fixed macroscopic observables. The solution is the Gibbs ensemble. As an example we take Table 4 as the observables.

<table>
<thead>
<tr>
<th>Observable</th>
<th>Conjugate variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy ( E )</td>
<td>Temperature ( \beta = 1/(k_B T) )</td>
</tr>
<tr>
<td>Volume ( V )</td>
<td>Pressure ( \gamma = p/(k_B T) )</td>
</tr>
<tr>
<td>Number of particles ( N )</td>
<td>Chemical potential ( \delta = -\mu/(k_B T) )</td>
</tr>
</tbody>
</table>

Table 4: Typical observables of statistical mechanics.

then the partition function becomes

\[ Z = \sum_{x \in X} e^{-\beta E(x) - \gamma V(x) - \delta N(x)} \]  

The probability of occupation of a micro-state is

\[ p(x) = \frac{1}{Z} e^{-\beta E(x) - \gamma V(x) - \delta N(x)} \]

the average values and their variance for the observables are
The laws of thermodynamics can be recovered from the partition function by taking the derivatives

\[ \frac{\partial S}{\partial E} \bigg|_{V,N} = \frac{1}{T}, \quad \frac{\partial S}{\partial V} \bigg|_{E,N} = p \frac{T}{T}, \quad \frac{\partial S}{\partial N} \bigg|_{E,V} = -\frac{\mu}{T} \]  \hspace{1cm} (3.7)

We summarize these equations to

\[ dE = TdS - pdV + \mu dN \]  \hspace{1cm} (3.8)

Related work on algorithmic thermodynamics. In their paper 4, John C. Baez and Mike Stay, suggest an interpretation of algorithmic information theory based on thermodynamics, where the characteristics of programs as considered to be observables. Starting from Gregory Chaitin’s \( \Omega \) number, the halting probability

\[ \Omega = \sum_{p \text{ halts}} 2^{-|p|} \]  \hspace{1cm} (3.9)

is extended with algorithmic observables to obtain

\[ \Omega' = \sum_{x \in X} e^{-\beta E(x) - \gamma V(x) - \delta N(x)} \]  \hspace{1cm} (3.10)

Noting the similarity between equation 3.2 and 3.10, they suggest an interpretation where \( E \) is the expected value of the logarithm of the program’s runtime, \( V \) is the expected value of the length of the program and \( N \) is the expected value of the program’s output. Furthermore, they interpret the conjugate variables as (quoted verbatim from their paper);

1. \( T = 1/\beta \) is the algorithmic temperature (analogous to temperature). Roughly speaking, this counts how many times you must double the runtime in order to double the number of programs in the ensemble while holding their mean length and output fixed.

2. \( p = \gamma/\beta \) is the algorithmic pressure (analogous to pressure). This measures the tradeoff between runtime and length. Roughly speaking, it counts how much you need to decrease the mean
length to increase the mean log runtime by a specified amount, while holding the number of programs in the ensemble and their mean output fixed.

3. \( \mu = -\delta / \beta \) is the algorithmic potential (analogous to chemical potential). Roughly speaking, this counts how much the mean log runtime increases when you increase the mean output while holding the number of programs in the ensemble and their mean length fixed.

From equation 3.10, they derive analogues of Maxwell’s relations and they consider thermodynamic cycles such as the Carnot cycle or Stoddard cycle. For this they introduce the concepts of algorithmic heat and algorithmic work.

The authors then claim that the choice of correspondence between thermodynamic observables and algorithmic observables is somewhat arbitrary and reference other authors 5 who have used completely different correspondences.

In this work, we study algorithm thermodynamics for the purpose of explaining thermodynamics. We will claim that there does exist a preferred correspondence between algorithmic observables and statistical mechanical observables. This preferred choice is set by the units of the observables. In a physical partition function, each observable must have the units of energy and is divided by \( \beta = 1/(k_bT) \) the units of which are also energy, leaving no units in the exponential. Therefore, in our replacements, the conjugate variables used must convert the units of its associated observable to an energy. Examples of allowed conjugate-observable pairs are listed in table 5.

<table>
<thead>
<tr>
<th>Observable</th>
<th>Variable</th>
<th>Units</th>
<th>Conjugate</th>
<th>Variable</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Halting event</td>
<td>( E )</td>
<td>( J )</td>
<td>Temperature</td>
<td>( \beta = 1/(k_bT) )</td>
<td>( 1/(J/K \times K) )</td>
</tr>
<tr>
<td>Program length</td>
<td>( x )</td>
<td>( m )</td>
<td>Force</td>
<td>( \gamma = F/(k_bT) )</td>
<td>( N/(J/K \times K) )</td>
</tr>
<tr>
<td>Running frequency</td>
<td>( \tau )</td>
<td>( 1/s )</td>
<td>Action</td>
<td>( \alpha = S/(k_bT) )</td>
<td>( J \times s/(J/K \times K) )</td>
</tr>
<tr>
<td>Runtime</td>
<td>( t )</td>
<td>( s )</td>
<td>Power</td>
<td>( \kappa = P/(k_bT) )</td>
<td>( W/(J/K \times K) )</td>
</tr>
</tbody>
</table>

**Theorem 3.11.** The general halting partition is the Boltzmann distribution used in statistical physics.

**Proof.** We try to explain thermodynamics and physics starting from algorithmic thermodynamics. First, we recall theorem 2.17. We recover thermodynamics by introducing an energy associated with each halting event, denoted by \( E \). Adding the rest of the conjugate-observable pairs, we obtain

\[ Z = \sum_{x} e^{-\beta(E + Fx + S\tau + \ldots)} \]

where the triple dots represent other possible observables. We interpret the program \( x \) as a micro-state of the set of all prefix-free programs that are run on the UTM. It is easy to see that the function for \( Z \) is the partition function of the Gibbs ensemble of thermodynamics.

Since both the running frequency and the runtime are associated with time, we only need to select one of them as our conjugate-observable pair. In this section, \( S\tau \) leads to conceptually simpler results and so is used here, whereas in the chapter on spacetime, \( Pt \) will be preferred.

3.1 Energy

**Theorem 3.12.** The law of conservation of energy.

*Proof.* Since we have recovered a thermodynamic partition function, we can define a conserved energy quantity. This is a direct consequence of taking the thermodynamic state equation of the partition function.

\[ dE = TdS - Fdx - Sd\tau - \ldots \]  

(3.13)

**Theorem 3.14.** When the values of the halting bits of \( Z \) are not known to an observer, each bit of information has the following energy

\[ E = k_B T \ln 2 \]

*Proof.* Consider an observer not aware of the bit values of \( Z \). To the observer, \( Z \) looks like

\[ Z_N = 0.\omega_1\omega_2\omega_3\ldots\omega_N \]  

(3.15)

There are \( W = 2^N \) different possibilities, or micro-states. Since each bit has two possible values, the entropy of the system is \( S = k_B \ln 2^N \). Adding or removing a bit changes the entropy and the energy by
\[ \Delta S = S_{N+1} - S_N = k_B \ln 2^{N+1} - k_B \ln 2^N = k_B \ln 2 \]
\[ \Delta E = T \Delta S = T (k_B \ln 2) \]

This result agrees with the well known Landauer limit \(^6\).

3.2 Time

Posing \( dE = 0 \), we can look at a simplified subset of the general halting partition involving only the frequency and the length observable.

\[ Z = \sum_x e^{-(\ln 2)\beta(F_x + S \tau)} \]  \hspace{1cm} (3.16)

where \( \tau = 1/t \).

**Theorem 3.17.** At the limit of \( t \to \infty \), we recover \( \Omega \)

**Proof.** A program \( p \) can have any value of \( S \) within \([0, \infty[\). If the program halts immediately, \( S = 0 \). If it never halts, \( S = \infty \). If it halts after a certain time, \( S = t_p \). A program that never halts will not be part of the halting partition. This will be the case if \( S = \infty \). As a result we obtain,

\[ \lim_{t \to \infty} \frac{S_x}{t} = \begin{cases} 
0 & \text{x halts} \\
\infty & \text{otherwise}
\end{cases} \]

This is the definition 2.6 of \( T(i) \). Hence

\[ \lim_{t \to \infty} \frac{S_x}{t} = T(x) \]

Therefore,

\[ \lim_{t \to \infty} Z = \lim_{t \to \infty} \left( \sum_x e^{-(\ln 2)\beta(F_x + S \tau)} \right) = \sum_x e^{-(\ln 2)\beta(F_x + T(x))} = \Omega \]

At \( t \to \infty \), the halting programs, i.e. \( S \in [0, \infty[\), have all halted, whereas the programs where \( S = \infty \) have not. \( \Box \)
Theorem 3.18. At the limit of $t \to 0^+$, we obtain initial conditions.

Proof. We study the limit of $t \to 0^+$. We obtain

$$
\lim_{t \to 0^+} \frac{S_x}{t} = \begin{cases} 
0 & S_x = 0 \\
\infty & \text{otherwise}
\end{cases}
$$

At that limit, the only programs that contribute to $Z$ are those that halt immediately. These are the initial conditions. \qed

Theorem 3.19. For $0 < t < \infty$, the partition function $Z$ is

$$Z(t) = \Omega - 2^{-k(t)}$$

where $2^{-k(t)}$ is an error rate that is monotonically decreasing to 0 as $t \to \infty$.

Proof. \footnote{Here, we reproduce the definition of $k(t)$ and the proof provided by John C. Baez and Mike Stay in their paper on algorithmic thermodynamics.}

Definition 3.20. For any $k \geq 0$ and time $t \geq 0$, let $k(t)$ be the location of the first zero bit after position $k$ in the estimation of $\Omega$.

Then because $-\frac{S_x}{t}$ is a monotonically decreasing function of the running frequency and decreases faster than $k(t)$, there will be a time step where the total contribution of all the programs that have not halted yet is less than $2^{-k(t)}$.

\qed

For example, say

$$\Omega = 0.0111100 \ldots$$

To keep it simple we consider, in isolation, a single program and assume that all other programs have long halted (at $t \to 0^+$). Let us take the values $x = 5$ and $S_x = 50$ for this program. We obtain,

$$Z_x(t) = 2^{-x}2^{-\frac{S_x}{t}}$$

$$Z_5(t) = 2^{-52}2^{-\frac{50}{t}}$$

$$= 0.00001 \times 2^{-50}$$

The halting probability $\Omega$ is,

$$\Omega = 0.011100 \cdots + Z_5(t)$$

Let us look at what happens as we vary $t$.

1. If $t \to 0^+$, then $Z_5(0^+) = 0$. $Z$ differs from $\Omega$ by the maximum uncertainty of $2^{-5}$. Therefore $\Omega - Z_5(0^+)$ is accurate only in its first 5 bits.
2. As $t \to \infty$, then $Z_5(\infty) = 0.00001$.

3. Between 0 and $\infty$, $Z_5(t)$ varies from $2^{-5}$ at $t = 0$ to 0 at $t \to \infty$. Since $-\langle S_5/t \rangle$ is monotonically decreasing, the uncertainty $2^{-k(t)}$ must decrease monotonically to 0 as $t$ increases.

4. At distances further than $2^{-k(t)}$, the partition function contains bits of programs that have yet to halt. So, in a sort, a reversal of time occurs where halting information is available before the time $t$ is long enough for the program to have halted.\(^8\)

3.3 Hard arrow of time

We now have the tools required to define a hard arrow of time.

**Theorem 3.21.** The far future ($t \to \infty$) is non-computable.

**Proof.** As proven in theorem 3.17,

$$\lim_{t \to \infty} Z(t) = \Omega$$

$\Omega$, as the halting probability of a random program read by a UTM, was proven to be non-computable by Gregory Chaitin. Likewise, the halting problem guarantees that the final days of the universe cannot be precisely calculated. We must wait to see it unfold. \(\square\)

**Theorem 3.22.** The future ($t > t_0$) is not pre-computable.

**Proof.** Without loss of generality suppose that,

$$Z(t_0) = 0.001...$$

$$Z(t_1) = 0.101...$$

$Z(t_0)$ differs from $Z(t_1)$ at digit $d_1$. Here we ask if and how can an observer at $t_0$ calculate the future at $t_1$. To calculate the future, an observer knowing $Z(t_0)$ must prove at time $t_0$ that $d_1 = 1$. However the observer can never succeed and here is why. If the observer does prove that $d_1 = 1$, then because he is part of the universe, the universe becomes $Z(t_1)$. Hence he has not pre-calculated the future, but simply travelled in time to a time where this result is known. Pre-calculating the future is equivalent to traveling forward in it to a time when the pre-calculation is true.\(^9\) \(\square\)

**Theorem 3.23.** The past ($t < t_0$) and the present ($t = t_0$) are immediately accessible.

\(^8\)In the chapter on spacetime we will see that this is sufficient to derive special relativity.

\(^9\)We will significantly strengthen this results in the subsection ‘Experiments & observers’
Proof. Without loss of generality suppose that,

\[ Z(t-1) = 0.001 \ldots \]
\[ Z(t_0) = 0.101 \ldots \]

As time goes forward more bits of \( Z \) are turned from 0 to 1. Since 0 bits represent an unknown answer as well as a negative answer, there is no loss of information when a bit-flip occurs. An observer at \( t_0 \) can read \( Z(t_0) \) and since \( Z(t_0) \) does not erase halting information with respect to \( Z(t-1) \), he will have read all the past and present knowledge of the universe. \( \square \)

3.4 Quantum information

In this section we attempt to describe a situation when the observer is lacking full knowledge of \( Z \). We ask, how do we modify \( Z \) to account for this lack of knowledge? We will see that there will be two ways to modify the halting partition such that it preserves its halting information. The first way involves adding a thermodynamic observable associated to each halting bit and the second way involves taking a degenerate halting partition over all possible values of \( \Omega \) (for all UTMs). With the first way, we will derive the existence of the quantum mechanical spin and with the second way (section 4.7), we will derive Schrödinger’s equation.

First, we define what we mean by a lack of knowledge by an observer with respect to \( Z \). An observer might be aware of some of the bits of \( Z \), but since the universe is very big presumably he cannot know them all. Therefore it must be reflected in his partition function that those bits are not fixed to a definite value.

We consider the case at \( t \to \infty \) where \( \Omega \) is available. Since the halting bit at every position is either a 0 or 1 and is non-computable, we will denote each bit as the matrix \([0, 1]\) rather than fix it at a specific value. Naively, we first try the following sum

\[
Z \equiv 0.1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0.01 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0.001 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0.0001 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \ldots
\]

However a problem immediately follows. Indeed, when the sum is executed, the matrices erase halting information and we would obtain the wrong result of

\[
Z \equiv 0.11111\ldots \begin{bmatrix} 1 \\ 0 \end{bmatrix}
\]

The bits are all correlated as they are either all 1, or all 0. To avoid this unfortunate consequence, we must break the "information de-
stroying” additivity of the matrices. This can be done by “flipping” some of the matrix such that a halting program is represented by a matrix $[1, 0]$ and a non-halting program is given by $[0, 1]$. As an example, assume that only the first and fourth program halts, we now obtain,

$$Z = 0.1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0.01 \begin{bmatrix} 0 \\ 1 \end{bmatrix} + 0.001 \begin{bmatrix} 0 \\ 1 \end{bmatrix} + 0.0001 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \ldots$$

$$= \begin{bmatrix} 1001 \ldots \\ 0110 \ldots \end{bmatrix}$$

$$= \begin{bmatrix} \Omega \\ 1 - \Omega \end{bmatrix}$$

This is enough to recover the bits of $\Omega$, but $Z$ still does not represent what the observer sees. Knowing $\Omega$ tells the observer which bit has halted and which has not. There is no room left for bits the values of which are unknown to him. To account for that, we associate, as a linear combination, both possible outcomes to each halting bit, such that

$$Z = 0.1 \left( c_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + d_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) + 0.01 \left( c_2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + d_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) + \ldots$$

where $c_i$ and $d_i$ are complex. It can be rewritten to the following form

$$Z = 0.1 \begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0.01 \begin{bmatrix} a_2 & b_2 \\ c_2 & d_2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \ldots$$

Adding a $2 \times 2$ matrix term in front of every halting bit might seem like an ad hoc solution, but all it means is that there will be a thermodynamic observable in the halting partition corresponding to each term. As long as we are willing to pay the “price” of this additional observable, then we are allowed to add the $2 \times 2$ matrix. No restrictions have thus far been placed on the elements of the $2 \times 2$ matrices. This creates a problem if it’s determinant is not equal to 1. Indeed in such a case the contribution of each halting bit to the probability would be either scaled up or down. To fix this problem, we pose that $\det \{ M_i \} = 1$ and we immediately obtain the $SU(2)$ representation group,

$$Z = 0.1 \begin{bmatrix} a_1 & -\bar{b}_1 \\ \beta_1 & \bar{\alpha}_1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0.01 \begin{bmatrix} a_2 & -\bar{b}_2 \\ \beta_2 & \bar{\alpha}_2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \ldots$$

where
$SU(2) = \left\{ \begin{pmatrix} \alpha & -\beta \\ \beta & \alpha \end{pmatrix} : \alpha, \beta \in \mathbb{C}, |\alpha|^2 + |\beta|^2 = 1 \right\}$

These modifications ensure that, because of the existence of an exponential map between a Lie group and a Lie algebra, we can rewrite $Z$ in its Gibb’s ensemble exponential form,

$$Z = \sum_x e^{-\beta (F x + \alpha_i i + \alpha_j j + \alpha_k k)}$$

where the unit quaternion $\alpha_i i + \alpha_j j + \alpha_k k$ is the Lie algebra correspondence to the Lie group $SU(2)$ and $\alpha_i, \alpha_j, \alpha_k$ are real numbers. The presence of the real number $\alpha$ imposed by the linear-algebraic correspondence conveniently allows us to recover the statistical mechanical form of an observable multiplied by its conjugate variable. Since the exponential terms represent an energy, it follows that it must be a real number. The existence of a double cover from $SU(2)$ to $SO(3)$ allows us to transform the unit quaternion to real numbers exclusively.

In this scenario, the thermodynamic observable is the three dimensional invariance over rotations and the effect on the partition function is that each halting bit becomes a spin. We have recovered the properties of the elementary two-state quantum system but with the added benefit that we obtain a rigorous definition of the quantum measurement. The measurement happens when the halting bit is read to a 1 or a 0 by an observer from $Z$. Since the halting probability is a normal number, it proves that the quantum measurement of a spin is algorithmically random.

The derived $Z$ becomes much more complex for the observer. He must now keep track of 4 independent real numbers (or 2 complex numbers) to describe the $Z$ that he sees. In other words, the observer now needs 4 classical bits per halting bit. We will see in the section on space that this 4 to 1 ratio is instrumental in deriving the exact equation of the Bekenstein-Hawking entropy of black holes.

Remark 3.24. We will explicitly construct an algorithm that meets these requirements in theorem 4.1. It encodes all available and unmeasured halting bits using 4 bits per halting bit. They are encoded such that, even unmeasured, they are uniquely identifiable by the observer.

Remark 3.25. There is an alternative derivation of this result that we could have used. We could have simply asked what happens to $Z$ when we introduce the three dimensional rotations as an observable? Reversing the arguments above, we would have seen that each halting bit becomes a spin. The derivation is somewhat shorter but it would not have been immediately obvious why we specifically use three dimensions and not two or fifty (unless we were to appeal to scientific observations). Introducing the $SU(2)$
representation as a solution to an observer with incomplete knowledge of halting information directly enforces 3D rotations of the two-state systems obtained.

3.5 Experiments & observers

A UTM executing a program to termination before starting working on the next one will hang at the first non-halting program. To avoid this, one might be tempted to start each program one by one, then run each program for one iteration, then each program for a second iteration and so on. Progress will be made on every program regardless of the existence of non-halting programs. However, if there are infinitely many program each potentially running for an infinite amount of time, the solution is to dovetail the programs.

**Definition 3.26 (Dovetailing programs).** Dovetailing is a method of computation where program execution in interweaved so that non-halting programs do not cause other programs to hang their execution. The first program is started then it is run for one iteration. Then, the second program is started. Then, the first followed by the second program are each ran for one iteration. Then, the third program is started. And so on.

The general halting partition clearly executes its programs in a dovetail-like fashion - see theorem 3.19. At any time, shorter programs will have had a longer running period than longer programs. This raises a question. The bits of $\Omega$ obtained via dovetailing are compressible to a very short algorithm. Can an observer recover the halting bits by running this program, and therefore correctly predict the measurement outcome of a spin before it is measured? The answer is no, because an observer can only run experiments.

**Definition 3.27 (An experiment).** A finite state machine executing a script. The experiment will be completed in a finite time and will use a finite amount of memory\(^{10}\).

If the experiment has less internal state than the script defining the UTM running the universe, the experiment will see an algorithmically random halting partition for all future times $t\, ^{11}$. Hence, its future will be non-calculable. The experiment’s arrow of time is enforced by the unsolvability of the halting problem using finite-state machines. For observers that can only run experiments, the same applies to them.

3.6 The universe

The bits encoding halting information are the source of entropy of the general halting partition. The other thermodynamic observables are

\(^{10}\) It may also provably never terminate, but this is not the halting problem.

\(^{11}\) As we will see in the subsection on the universe, this implies a quantum mechanical universe.
non-information bearing. Since this represents $\Omega$ and because of the conservation of energy, it follows that all information present in the universe is reducible to the halting probability of its algorithmic formulation.

A halting probability of a UTM is the most compressed form possible for this information, and can be used to prove any sentence of the language. This can be seen in the following way. Suppose that we want to prove the Riemann hypothesis - a millenium problem with a reward of 1,000,000$. Knowing $\Omega$, it is very easy. We simply write a program that will halt if the proof is found or run forever. Specifically, this program could alphabetically try proofs in dovetail fashion until one is found, then halt. Now, this program $p$ will have a well defined length $l_p$. If we know $\Omega$, we can prove the Riemann hypothesis like this. We run all programs from shortest to longest in dovetail fashion on the UTM. Every once in while we look at the estimation of $\Omega$ obtained and compare it to the real value of $\Omega$. If the uncertainty is less than $2^{-l_p}$ and $p$ has not halted, we know the proof does not exist and we claim the reward.

A universe at $t \to \infty$ is a universe that contains all the answers to all the questions that can be asked in it. Essentially, it has solved the halting problem. To not solve the halting problem, the universe must exist at a time $t \in [0, \infty[$. We can now define both time and the universe.

**Definition 3.28 (Universe).** A universe comprises

1. a language $L$,
2. its alphabetical enumeration (the list of all sentences in $L$ in alphabetical order),
3. a halting probability $\Omega$ for the unitary coding of the alphabetical enumeration of $L$ where the bit at position $i$ corresponds to the halting status of the program of length $i$,
4. a halting probability given by:

$$\Omega = \sum_x e^{-(\ln 2)\beta(E+Fx)}$$

This defines the universe as the sum of all the knowledge that exists within it for its entire history. From this definition and supplemented with the running frequency to action observable $S\tau$, we also obtain a definition of the universe at a time $t$.

**Definition 3.29 (Universe at time $t$).** The universe as defined by 3.28, at time $t$ has a halting partition given by

---

12 If a macroscopic observer builds a computer out of the non-information bearing observables, would the computer not store information? Answer - yes, but this information would also be encoded in the entropy of the system. It is no more than duplicated information.
\[ Z = \sum_{x} e^{-(\ln 2)\beta(E+Fx+S\tau)} \]

At time \( t \in [0, \infty) \), \( Z \) replaces \( \Omega \).

Describing a time forward or backward is respectively the same as adding or removing the appropriate 1-valued bits to or from \( Z \). Analogous to \( \Omega \), \( Z \) represents the probability that a program has halted within a certain time frame.

### 3.7 Discussion

From the perspective of an observer running finite-state experiments, access to the full listing of \( \Omega \) is prohibited due to memory and time constraints. As a result, the observer must account for the uncertainty (e.g. superposition) of each bit of \( \Omega \) that are unknown to him. From these considerations, quantum mechanics follows as we have previously seen with the spin and, in the next section, we will derive Schrödinger’s equation from the same considerations.

The quantum measurement and simultaneous wave function collapse occurs when the halting bit becomes known to ‘all possible observers’ \(^{13}\). If the halting bit is known only to one observer, he becomes entangled with the system. This entangled system finally collapses when the information propagates to other observers.

The halting probability experienced by an observer who lack complete knowledge of \( Z \), is obtained when we take each possible halting probability and combine them in a degenerate partition function. Successful quantum measurement and wave function collapse reduces the degeneracy of the partition function experienced by the observer.

This is easily formalized from statistical mechanics by considering that \( Z \) is a canonical ensemble. A micro-canonical ensemble can be constructed by taking \( 2^N \) canonical ensembles. The maximally entangled ensemble seen by an observer with no halting knowledge is then given by

\[ Z' = \frac{1}{A} \sum_{i} \left( \sum_{\xi} e^{-(\ln 2)\beta(E+Fx)} \right) \]  
(3.30)

where the inner sum is iterated over the set of all possible \( \Omega \) (for all UTMs), and \( A \) is a normalization constant. Taking a very long yet finite-length prefix of \( \Omega \), we simplify the problem by assuming that each bit arrangement is equally likely. This simplification is why we were able to derive the energy per bit of \( k_B T \ln 2 \) in theorem 3.14.

Why is the future not pre-calculable? It is possible to approximate the future to a high degree of certainty. For example, we know that

\(^{13}\) By ‘all possible observers’ we mean here that the information related to the halting bit value propagates macroscopically in the system such that it loses its entanglement and quantum character.
the Sun will shine for about 4 billion years more. However, this is not guaranteed. A mini black hole could swallow it in say 1 billion years. To calculate, with certainty, if the Sun will still shine in 4 billion years, we must rule out all of these edge cases. This can only be done if we solve the full universe.

In a relaxed sense we equate the philosophical problem of a frustrator to the halting problem of computer science.

**Definition 3.31 (Frustrator in philosophy).** A frustrator in philosophy is an agent having free will who, knowing his future, makes the choice which guarantees the known future will not occur.

For example, a lottery player who knows he will win the next draw might simply decide not to play. This will frustrate the prediction. Since a free agent can be a frustrator to any future event, the inescapable conclusion is that no free agent can know their future. The frustrator problem in philosophy is usually taken as a strong suggestion that the future cannot be predicted with certainty.

In our derivation the future is presented as the solution to the halting problem. Since the observer cannot solve the halting problem, he must wait for the future to unfold in order to see what it is.
4 Space

Why does space have three dimensions? We have derived a rotational invariance as the thermodynamic observable associated with the spin of the bits of a halting probability. In this chapter, we will extend this result to define a linear position away from an origin. This result will be strong enough to derive black holes, the force of gravity and Schrödinger’s equation.

4.1 Maximum entropy

**Theorem 4.1.** The maximal entropy of a section of a halting probability represented in space by spins is

\[ S = k_B \frac{c^3 A}{4G} \]

where \( A \) is the area of the sphere enclosing the volume used for the measurements.

**Proof.** In the previous section, we have seen that the spin of a halting bit leads to the statistical mechanic observable of a rotation in 3D. It is therefore natural to relate the spin to the sphere. We start with an empty sphere and add spins one by one, until we reach a maximum. We ask; how many can we fit?\(^*\)

**Lemma 4.2.** The spin requires four degrees of freedom to be fully represented.

**Proof.** Two degrees are taken up by the yawn and pitch angle of the wave-function pointing to a point on the top-half surface of the Bloch sphere. These are respectively angle \( \phi \) and \( \theta \) shown on Figure 1. The Bloch sphere represents a pure quantum state system with wave-function

\[ |\psi\rangle = \alpha |0\rangle + \beta |1\rangle, \text{ where} \]

\[ \alpha = \cos \frac{\theta}{2} \]

\[ \beta = e^{i\phi} \sin \frac{\theta}{2} \]

The third degree is associated with the result for the measurement value of the spin along the \( z \) axis. Finally, the fourth degree is associated to the measurement value along the \( xy \) plane.\(^{15}\) This completes the full description of a spin, including all observables and measurement outcomes.

---

\(^*\)We take all spins to be centred at the origin. Introducing a position away from the origin will require bits to describe it. As we introduce a proper definition of position in the next subsection, we will see that this balances the entropy to eliminate any gains.

\(^{15}\)The third and the fourth degrees are not available from the wave-function but are instead read by a quantum measurement.
Remark 4.3. Why must the measurement outcome of orthogonal quantum states both be described pre-measurement, if only one of them can be measured? After measurement and wave-function collapse, the system has one degree of freedom. The pure-state is destroyed and is replaced with a classical bit. However, the system does not know which axis will be measured first. Therefore they must both be described. The perturbation of the measurement resets the bit values associated with the spin.

The two degrees of freedom of the yaw and pitch angle of the wave-function are real numbers and require up to infinitely many classical bits to be expressed to the desired precision. It would be a bad idea to encode the orientation of spins using real numbers because there is no limit to their precision. A single spin could consume arbitrarily many bits to describe just one of its real numbered degrees of freedom.

However, we consider the Pauli exclusion principle, and note that a non-integer spin cannot occupy the same quantum state. We also note that the precision required to define a spin is wasted beyond the precision of the instrument measuring it. This instrument cannot be precisely rotated with an angle smaller than the Planck angle. As a result, the spins need not be expressed more accurately than the Planck angle.

Lemma 4.4. Since the distinguishable rotation angles are finite, there exists an efficient way to encode spins in a sphere, where one classical bit maps to one degree of freedom.

Proof. We associate classical bits to a position on the surface of the enclosing sphere. It is natural to pose that the number of bits that can fit on its surface is given by

\[ N = \frac{A}{L_p^2} = \frac{c^3 A}{Gh} \]

where \( L_p^2 \) is the Planck length. We can maximize the number of spins in a volume, and therefore maximize the entropy, if we minimize the number of classical bits required to express each spin. Figure 2 shows a spherical geodesic divided in equal areas, each occupied by a classical bit. Each bit on the surface encodes the presence or absence of a pure state wave-function pointing to the bit and centered at the origin.

From the Bloch representation, we easily see that only half of the bits on the surface are actually needed to express all permissible spin orientations in the volume. For each second bit, a one-valued bit

![Figure 2: The bits on the surface encode the degrees of freedom of the spin.](http://www.scholarpedia.org/article/File:BHentropyF1.jpeg)
on the surface of the sphere corresponds to the presence of a wave-function $|\psi\rangle$ pointing to it, while a zero-valued bit indicates a vacant spot. Figure 3 illustrates the encoding of the orientation in 2D.

The remaining half of bits are used for the third and fourth degrees of freedom. This encoding is maximally efficient and each classical bit encodes a single degree of freedom.

We now want to determine the maximum entropy of the spin system describable by those bits. Since the general halting partition has a factor $\ln 2$ on the exponential term, the derivative $dS/dE$ gives us a pseudo-entropy related to the real entropy by a factor $1/\ln 2$.

$$
\frac{dS'}{dE} = \frac{\ln 2}{k_B T}
$$

$$
dE = k_B T \frac{dS'}{\ln 2}
$$

$$
= k_B T dS
$$

(where $S \ln 2 = S'$)

The factor $\ln 2$ is obtained because the statistical mechanics partition is derived from the base-2 sum of the halting probability. This implies that,

$$
S = \frac{1}{\ln 2} S'
$$

$$
= \frac{1}{\ln 2} k_B \ln W
$$

where $W$ is the number of micro-states for the system. Since each micro-state (e.i. spin) requires four classical bits to be described, a factor of $\frac{1}{4}$ is added, such that

$$
W = 2^{\frac{3}{4}N}
$$

$$
= 2^{\frac{3}{4} \frac{c^3 A}{4Gh}}
$$

$$
\ln W = \frac{c^3 A}{4Gh} \ln 2
$$

$$
S = \frac{1}{\ln 2} k_B \frac{c^3 A}{4Gh} \ln 2
$$

Simplifying the constants, we get

$$
S = k_B \frac{c^3 A}{4Gh}
$$

which is the Bekenstein-Hawking entropy.
Theorem 4.5. At most, half of the permissible orientations are occupied by spins.

Proof. First, note that 4 classical bits are required to describe a spin. Second, note that 2 of these classical bits are used to mark occupied orientations. This means that half of the orientations must be vacant.

The halting probability of a UTM is a normal real number that is algorithmically random. This implies that, statistically, it contains as many zeros as ones in its binary digit representation. Conveniently, the maximal entropy of the sphere occurs when the bits on its surface are composed of equally many zeros and ones.

4.2 Position

We recall the general halting partition,

\[ Z = \sum_x e^{-\beta(E + Fx + S\tau + \ldots)} \]

and we specifically focus our attention on the second term in the sum of the exponential \( Fx \), where \( F \) is a force with units Joules per meter and \( x \) is the length of the program with units meter. This term has the same form and units as an entropic force of the form \( Fdx = TdS \). For example, entropic forces also describe polymer tension, osmotic force, etc.

Since \( x \) represents the program’s length in meters, this result strongly suggests that it is encoding spatial lengths. To encode length using bits we will consider an efficient algorithm with the following characteristics;

1. It must have the observer as its origin.
2. It must be able to express positions arbitrarily far away from the origin.
3. It must be able to express positions with arbitrary precision.

Naively, we might be tempted to use a sequence of bits to represent the quantity of discreet steps required to reach a point in space from the present location. For example, suppose the plank’s length is the smallest unit of space. Then an object of 1m away from the origin would have its position encoded with the number \( 6.25 \times 10^{34} \) in binary. A rather large bit requirement for something that is very simple to describe in layman’s terms.

We can make significant better use of our bits by specifying a scale before listing the bits. For example, let’s say we only care about
specifying the position within a centimetre. We could initially specify a scale, then list the number of repetition of the scale unit to reach the location of interest.

The price to pay for such algorithmic compression is that the position of an object is definable up to a certain precision, and it must have a characteristic scale. I will conjecture that this is the most efficient position encoding algorithm that meets the requirements above.

**Definition 4.6 (Positioning via program length, or PVPL for short).** Suppose a Cartesian coordinate system with orthogonal axis \( x \ y \ z \) and an origin at \((0,0,0)\). The position of a point will be encoded via four digits. One scaling constant \( \lambda \), and 3 scale-repetition values \( l_x, l_y \) and \( l_z \). Its position is noted as \( p = (\lambda, l_x, l_y, l_z) \). The point can be identified within an error margin of \( \pm (1/2) \lambda \). The scale \( \lambda \) can be made as small as we want it by increasing the bit count of \( l_x, l_y \) and \( l_z \).

**Theorem 4.7 (Conversion formulas).** PVPL encoding can be converted back to cartesian coordinates by multiplying the scaling factor \( \lambda \) with the repetition value for each coordinate and keeping the uncertainties.

\[
\begin{align*}
x &= l_x \lambda \pm \Delta \lambda \\
y &= l_y \lambda \pm \Delta \lambda \\
z &= l_z \lambda \pm \Delta \lambda
\end{align*}
\]

where \( \Delta \lambda = (1/2) \lambda \).

For example, a point at cartesian coordinate \((0,0,1)\) with an error margin \( \pm 0.1 \) is encodable in PVPL with the following 4-digits: \( p_1 = (0.1, 0, 0, 1010_8) \). To double the precision without changing the position, we must add an extra bit to the repetition value in \( z \) and halve the scaling constant such that \( p_2 = (0.05, 0, 0, 10100_8) \). Using PVPL to describe near objects of similar scales has the advantage of requiring a very low bit count.

**Theorem 4.8 (Positional entropy).** Using the positional encoding of definition 4.6, we can pose a relation between \( \Delta S \) and \( \Delta x \) of

\[
\Delta S = 2\pi k_B \frac{1}{\lambda} \Delta x
\]

Proof. Naively, we might be tempted to allow the PVPL values \( l_x, l_y \) and \( l_z \) to take any bit sequence. If we do so, doubling \( \Delta x \) from 10 meters to 20 meters (or in binary; \( 1010_8 \) meters to \( 10100_8 \) meters) increases the bit count by 1. Since this does not double the entropy, it contradicts theorem 4.8.

However, it would be a mistake. PVPL values are not the bit sequence themselves, but the length of the program. What is the entropy of a program of length \( L \)? A physical observer will measure the
entropy of the program to be equal to its length. A physical observer
does not and cannot know the precise prefix-free encoding because
he sees a degenerate general halting partition. Hence, he must accept
that all possible bit sequences could represent a program of length $L$.
Since there are $2^L$ such sequences, $S = k_B L = k_B \Delta x$.

Why multiply $\Delta x$ with $1/\lambda$? For example, suppose we have $\Delta x = 5$ meters. If we encode the position using the meter scale, we will
require a program of length 5. However, if we use the centimetre
scale, the required program length will now be 500. This increases
the entropy by a factor of 100. The smaller the scale the higher theentropy.

Why multiply $\Delta x$ with a factor of $2\pi$? Our previous result from
theorem 4.1 shows that the number of bits depends on the area of the
sphere. This results suggests that the bits rest of the surface of the
sphere defining a volume in space.

Suppose a circle of radius $r$ with $N$ points uniformly distributed
on its perimeter. The points on the perimeter maps to a segment on
a line of length $L$. Keeping the scale intact, the length of the segment
will be $L = 2\pi r$ and each point will be separated by a distance
d = $2\pi r / N$. The factor $2\pi$ multiplying $\Delta x$ is a consequence of the
scale preserving mapping of linear distances encoded by equidistant
bits on a circle.

Why go to such lengths to avoid rescaling the line coordinates
and to keep the factor $2\pi$? We are not allowed to rescale the axis as
a rescale would increase the precision of all positions and therefore
consume more bits.

$\square$

4.3 Schwarzschild radius

**Theorem 4.9.** Using all the classical bits on the surface of a sphere encoding a halting probability, we encode, using entropic positioning, the longest
distance that can be expressed for the black hole. We find that this distance is
equal to the Schwarzschild radius,

$$r_s = \frac{2GM}{c^2}$$

**Proof.** According to theorem 4.1, the maximum entropy in a volume
of space is

$$S = k_B \frac{A c^3}{4Gh} = k_B \frac{\pi r^2 c^3}{hG}$$

We will convert all of these bits to entropic positioning so that we
can express the point furthest away from the center of the sphere. We
recall equation 4.14.
\[ \Delta S = 2\pi k_B \frac{1}{\lambda} \Delta x \]

Equating the two entropies (\( \Delta S = S \)), we get

\[ k_B \frac{\pi r^2 c^3}{\hbar G} = 2\pi k_B \frac{1}{\lambda} \Delta x \]

Since we are dealing with a black hole, it should not be able to express programs of length longer than its horizon, or information might leak out. Therefore we pose \( \Delta x = r \), and we obtain,

\[ k_B \frac{\pi r^2 c^3}{\hbar G} = 2\pi k_B \frac{1}{\lambda} (r) \]

Solving for \( r \), we obtain

\[ r = \frac{1}{k_B} \frac{\hbar G}{\pi c^3} 2\pi k_B \frac{1}{\lambda} \]

(4.10)

Then, we take the algorithmic scaling factor \( \lambda \) to be, in fact, the Compton wavelength. We obtain

\[ r = \frac{1}{k_B} \frac{\hbar G}{\pi c^3} 2\pi k_B \left( \frac{Mc}{\hbar} \right) \]

Reducing the constants, we obtain

\[ r = \frac{2GM}{c^2} \]

Which is the Schwarzschild radius.

\[ \square \]

4.4 Hawking radiation

**Theorem 4.11.** A sphere encoding bits of a halting probability will radiate at a temperature inversely proportional to its mass. The temperature will be consistent with Hawking radiation;

\[ T = \frac{\hbar c^3}{8\pi GMk_B} \]

**Proof.** According to theorem 4.1, the maximum entropy in a volume of space is

\[ S = k_B \frac{Ac^3}{4G\hbar} = k_B \frac{\pi r^2 c^3}{\hbar G} \]

Using the thermodynamic relation \( dE =TdS \), we want to obtain the energy, then the mass via \( E = mc^2 \).

\[ dS = 2k_B \frac{\pi r^3 c^3}{\hbar G} dr \]
multiplying by $T$, we obtain the derivative of energy
\[ dE = TdS = 2k_B T \frac{\pi r c^3}{\hbar G} dr \]

dividing by $c^2$, we obtain the mass
\[ dM = \frac{1}{c^2} dE = 2k_B T \frac{c}{\hbar G} r dr \]

posing $r$ to be the Schwarzschild radius calculated in theorem 4.9 and $dr$ to be its derivative with respect to the mass, we obtain
\[ dM = 2k_B \pi T \frac{1}{\hbar G} c T \frac{c}{c^2} dM \]

Solving for $T$ and reducing the constants, we obtain
\[ T = \frac{\hbar c^3}{8\pi GMk_B} \]

, the temperature of Hawking radiation.

\[ \Box \]

4.5 Newton’s law of inertia

**Theorem 4.13.** The positional entropy leads to Newton’s law of inertia and to the proof of the existence of inertia.

\[ F = ma \]

**Proof.** The laws of statistical physics state that we can take the derivative of the entropy with respect to $x$ and obtain $F/T$. For an entropic force, the derivative is
\[ \frac{\partial S}{\partial x} = \frac{F}{T} \]

In the quasi-static approximation, the derivative becomes
\[ F\Delta x = T\Delta S \]

Let us replace $\Delta S$ with theorem 4.8. We obtain
\[ F\Delta x = T \left( 2\pi k_B \frac{1}{\hbar} \Delta x \right) \]

\[ F = 2\pi k_B T \frac{1}{\hbar} \]

(4.14)

Inspired by Erik Verlinde’s paper on entropic gravity \(^{16}\), we take $T$
to be Unruh’s temperature and we obtain

$$F = 2\pi k_B \left( \frac{1}{2\pi k_B} \frac{\hbar a}{c} \right) \frac{1}{\lambda}$$

Then, we take the algorithmic scaling factor $\lambda$ to be, in fact, the Compton wavelength ($\lambda = \frac{\hbar}{mc}$). We obtain

$$F = 2\pi k_B \frac{1}{2\pi k_B} \frac{\hbar a}{c} \left( \frac{mc}{\hbar} \right)$$

Finally, it reduces to

$$F = ma$$

\[ \square \]

4.6 Newton’s law of gravity

**Theorem 4.15.** Bits spread across the surface of a sphere will produce an entropic force governed by Newton’s law of gravity. Namely, a force with the following formula.

$$F = \frac{GmM}{r^2}$$

**Proof.** Here we abandon the volume entropy and consider that gravity is an entropic force created from the very bits that encode the universe at its most fundamental level. The number of classical bits on the surface is

$$N = \frac{c^3 4\pi r^2}{G\hbar}$$

From this equation, we can obtain the energy via the equipartition theorem which maps to the energy of a system to its number of degrees of freedom.\(^{18}\)

$$E = \frac{1}{2} k_B T N \quad \text{(equipartition theorem)}$$

$$= \frac{1}{2} k_B T \left( \frac{c^3 4\pi r^2}{G\hbar} \right)$$

We can obtain the mass by dividing by $c^2$

$$M = \frac{1}{c^2} E = k_B T \frac{2\pi r^2 c}{hG}$$

solving for $T$, we obtain

$$T = \frac{\hbar GM}{2\pi k_B c r^2}$$
We recall equation 4.14 and insert the $T$ we just found in it and we obtain

$$ F = \frac{2\pi k_B T}{\lambda} $$

$$ F = \frac{2\pi k_B \left( \frac{\hbar GM}{2\pi k_B c r^2} \right)}{\lambda} $$

Again, posing that the algorithmic scale factor $\lambda$ is in fact the Compton wavelength, we obtain

$$ F = \frac{2\pi k_B \hbar GM}{2\pi k_B c r^2} \left( \frac{mc}{\hbar} \right) $$

Note here that we introduce $m$ as the mass of the Compton wavelength as opposed to $M$ because we want to quantify the force felt between two objects of different masses. Finally, it reduces to

$$ F = \frac{G m M}{r^2} $$

Erik Verlinde further generalizes entropic gravity to account for arbitrary matter distributions.

### 4.7 Schrödinger’s equation

**Theorem 4.16.** A position described by entropic positioning (theorem 4.8) will evolve in time according to Schrödinger’s equation.

$$ \frac{i\hbar}{\partial t} \psi(x,t) = \left[ -\frac{\hbar^2}{2m} \nabla^2 + V(x,t) \right] \psi(x,t) $$

The proof is slightly more involved than the preceding theorems. First, here is a sketch

1. We will show that entropic position encoding using the bits produced by the general halting partition leaves holes in space where a position cannot be expressed.

2. We will show that these holes are causing a Brownian motion of the encoded position.

3. We will derive its diffusion coefficient to be $\hbar/(2m)$.

4. We will consider that the presence of any external field is experienced as acceleration via $F = ma$.

5. Using the well known Brownian motion equations of Langevin, we show that the above reproduces Schrödinger’s equation exactly.
Lemma 4.17. Holes.

Proof. We recall the general halting partition

\[ Z = \sum_x e^{-(\ln 2)\beta(E+Fx)} \]

We have previously seen how positions can be encoded with PVPL. We have also seen that the observable \( x \) denotes program lengths. However, not all programs halt hence some lengths are missing from the sum. These missing programs are holes in space the position of which cannot be expressed by the general halting partition. Since \( \Omega \) is a normal number, we can predict certain randomness related properties of these holes. \( \square \)

Lemma 4.18. A particle in space will experience Brownian motion due to the holes.

Proof. We will calculate the average displacement \( \bar{\Delta x} \) of a particle subjected to entropic positioning and space holes. Since \( Z \) is a normal number, we conclude that half of the program’s lengths are available to describe position and half are not. Therefore, to describe a particle at position \( x \), there is a 50% chance there is a halting program available to express it. And in the case where there is no program at exactly \( x \), then there is a 50% chance that there will be one at position \( x + 1 \), and so on. In other words, a particle at \( x \) has 50% chance of being at \( x \), 25% chance of being at \( x + 1 \), 12.5% chance of being at \( x + 2 \), etc. Expressed as a sum, we obtain

\[
\bar{\Delta x} = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \cdots
\]

\[
= \sum_{i=0}^{\infty} \frac{i}{2^{i+1}}
\]

\[
= 1
\]

On average, as it moves through space, a position will shift by \( \bar{\Delta x} = 1 \) at each iteration of the Brownian motion. But will a stationary point also experience Brownian motion? The answer is yes. A stationary point will experience Brownian motion because of the degeneracy of the general halting partition. The general halting partition is degenerate to an observer because he does not know the specific \( \Omega \)-value used by the universe. Therefore he must assume that all \( \Omega \) are possible and that each forms a degenerate state. As the micro-states switch around the possible degeneracies, the holes are moved around. The probabilities are the same as the sum above and \( \bar{\Delta x} = 1 \) for a stationary point. \( \square \)
Lemma 4.19. The diffusion coefficient of the described Brownian motion is

\[ D = \frac{\hbar}{2m} \]

Proof. It is well known that in general the diffusion coefficient of Brownian motion is given by

\[ D = \frac{l^2}{2\tau} \]

where \( l \) is the length of the random step and \( \tau \) is the frequency of the occurrence of the steps. Entropic position uses the scale factor \( \lambda \) for each unit of length. When \( \lambda \) is the Compton wavelength, we get a scaling factor of

\[ \lambda = \frac{\hbar}{mc} \]

Since entropic positioning can only express position as multiples of \( \lambda \), we take it as the Brownian step of length \( l \). The diffusion coefficient becomes

\[ D = \left( \frac{\hbar}{mc} \right)^2 \frac{1}{2\tau} \]

This leaves us with the need to define \( \tau \). For \( \tau \), we take the characteristic frequency of the wave \( E = \hbar \omega \). Solving for \( \tau = 1/\omega \), we obtain

\[ \omega = \frac{E}{\hbar} \]
\[ \tau = \omega^{-1} = \frac{\hbar}{E} \]

Replacing \( \tau \) in the equation for \( D \), we obtain

\[ D = \frac{\hbar^2}{m^2 c^2} \left( \frac{E}{2\hbar} \right) \]

Using \( E = mc^2 \), and reducing the constants, we obtain our final expression of \( D \),

\[ D = \frac{\hbar^2}{m^2 c^2} \left( \frac{mc^2}{2\hbar} \right) = \frac{\hbar}{2m} \]

\[ \square \]
Lemma 4.20. The Langevin equations for Brownian motion with a diffusion coefficient of $\bar{h}/(2m)$ and an external field $F = ma$ reproduces Schrödinger’s equation.

Proof. We recall the well known Langevin equation,

$$d[x(t)] = v(t)dt$$  \hspace{1cm} (4.21)

$$d[v(t)] = -\gamma m v(t)dt + \frac{1}{m} W(t)dt$$  \hspace{1cm} (4.22)

where $W(t)$ is a random force and a stochastic variable giving the effect of a background noise to the motion of the particle.

From $F = ma$ and replacing the acceleration $d[v(t)]/dt$ with $F/m$, Edward Nelson \(^{19}\) is able to show that the Langevin equation becomes,

$$\frac{1}{2} \nabla u^2 + D \nabla^2 u = \frac{1}{m} \nabla V$$  \hspace{1cm} (4.23)

where $D$ is the diffusion coefficient of $\bar{h}/(2m)$ obtained in lemma 4.19, where $F = -\nabla V$, where $u = v \nabla \ln \rho$ and $\rho$ is the probability density of $x(t)$. For brevity, the proof of 4.23 is omitted here but can be reviewed in his paper. Eliminating the gradients on each side and simplifying the constants, we obtain

$$\frac{m}{2} u^2 + \frac{\bar{h}}{2} \nabla u = V - E$$  \hspace{1cm} (4.24)

where $E$ is the arbitrary integration constant. This equation in nonlinear because of the term $u^2$ but it can be made linear by a change of dependant variable. To make it linear, let us pose

$$u = \frac{\bar{h}}{m \psi} \nabla \psi$$

and replace it into equation 4.24, we obtain

$$\frac{m}{2} \left( \frac{\bar{h}}{m \psi} \nabla \psi \right)^2 + \frac{\bar{h}}{2} \nabla \left( \frac{\bar{h}}{m \psi} \nabla \psi \right) = V - E$$

taking the gradients and the exponents, we obtain

$$\frac{\bar{h}^2}{2m \psi^2} \nabla^2 \psi + \frac{\bar{h}^2}{2m} \left[ -\frac{1}{\psi^2} \nabla^2 \psi + \frac{1}{\psi} \nabla^2 \psi \right] = V - E$$

The first two terms cancel each other.

$$\frac{\bar{h}^2}{2m \psi} \nabla^2 \psi = V - E$$
Finally, it simplifies to
\[
\left[-\frac{\hbar^2}{2m} \nabla^2 + V - E\right] \psi = 0 \quad (4.25)
\]
which is the time independent Schrödinger’s equation.

We are now ready to derive the time dependent Schrödinger’s
equation and prove theorem 4.16.

**Proof.** We use the same proof used by Edward Nelson in the same
paper. Starting from the time dependent Schrödinger’s equation and
show that a replacement of \( \psi = e^{R+iS} \) leads to the Langevin equa-
tion of Brownian motion. We write the time dependent Schrödinger’s
equation, perform the replacement and obtain the Langevin equations
of Brownian motion.

\[
\frac{\partial \psi}{\partial t} = i \frac{\hbar}{2m} \nabla^2 \psi - i \frac{1}{\hbar} V \psi \quad (4.26)
\]

Replacing \( \psi \) with \( e^{R+iS} \), we obtain
\[
\begin{align*}
\frac{\partial}{\partial t} (e^{R+iS}) &= i \frac{\hbar}{2m} \nabla^2 (e^{R+iS}) - i \frac{1}{\hbar} V (e^{R+iS}) \\
\end{align*}
\]

Taking the derivatives and the gradients, we obtain
\[
\left[ \frac{\partial R}{\partial t} + i \frac{\partial S}{\partial t} \right] (e^{R+iS}) = i \frac{\hbar}{2m} \left[ \nabla^2 R + i \nabla^2 S + (\nabla (R + iS))^2 \right] (e^{R+iS}) - i \frac{1}{\hbar} V (e^{R+iS})
\]

Eliminating \( e^{R+iS} \) from each side and simplifying, we obtain

\[
\begin{align*}
\frac{\partial R}{\partial t} + i \frac{\partial S}{\partial t} &= i \frac{\hbar}{2m} \left[ \nabla^2 R + i \nabla^2 S + (\nabla (R + iS))^2 \right] - i \frac{1}{\hbar} V \\
\text{(eliminating } e^{R+iS}) \\
\frac{\partial R}{\partial t} + i \frac{\partial S}{\partial t} &= i \frac{\hbar}{2m} \left[ \nabla^2 R + i \nabla^2 S + (\nabla R)^2 + 2i \nabla R \nabla S - (\nabla S)^2 \right] - i \frac{1}{\hbar} V \\
\text{(taking the product)} \\
\frac{\partial R}{\partial t} + i \frac{\partial S}{\partial t} &= \frac{\hbar}{2m} \left[ \nabla^2 R - \nabla^2 S + i (\nabla R)^2 - 2 \nabla R \nabla S - i (\nabla S)^2 \right] - i \frac{1}{\hbar} V \\
\text{(distributing the } i) \\
\end{align*}
\]

We obtain two equations by separating the real and the imaginary
parts

\[
\begin{align*}
\frac{\partial R}{\partial t} &= \frac{\hbar}{2m} \left[ -\nabla^2 S - 2 \nabla R \nabla S \right] \\
\text{(4.27)} \\
\frac{\partial S}{\partial t} &= \frac{\hbar}{2m} \left[ \nabla^2 R + (\nabla R)^2 - (\nabla S)^2 \right] - \frac{1}{\hbar} V \\
\text{(4.28)}
\end{align*}
\]

This is equivalent to the Langevin equations with some replace-
ments
\[ \frac{\partial u}{\partial t} = -\frac{\hbar}{2m} \nabla^2 v - \nabla (v \cdot u) \quad (4.29) \]
\[ \frac{\partial v}{\partial t} = \frac{\hbar}{2m} \nabla^2 u + \frac{1}{2} \nabla (u^2) - \frac{1}{2} \nabla (v^2) - \frac{1}{m} \nabla V \quad (4.30) \]

**Lemma 4.31.** Equation 4.27 with the replacements \( \nabla R = (m/\hbar) u \) and \( \nabla S = (m/\hbar) v \) produces 4.29.

Proof.
\[
\frac{\partial R}{\partial t} = \frac{\hbar}{2m} \left[ -\nabla^2 S - 2 \nabla R \nabla S \right] \quad \text{(equation 4.27)}
\]
\[
\nabla \frac{\partial R}{\partial t} = \nabla \frac{\hbar}{2m} \left[ -\nabla^2 S - 2 \nabla R \nabla S \right] \quad \text{(multiplying by \( \nabla \))}
\]
\[
\frac{\partial \nabla R}{\partial t} = \nabla \frac{\hbar}{2m} \left[ -\nabla R - 2 \nabla S - 2 \nabla R \nabla S \right] \quad \text{(replacing \( \nabla R \) and \( \nabla S \))}
\]
\[
\frac{m}{\hbar} \frac{\partial u}{\partial t} = \nabla \frac{\hbar}{2m} \left[ -\nabla \left( \frac{m}{\hbar} v \right) - 2 \left( \frac{m}{\hbar} u \right) \left( \frac{m}{\hbar} v \right) \right] \quad \text{(replacing \( \nabla R \) and \( \nabla S \))}
\]
\[
\frac{\partial u}{\partial t} = \nabla \frac{\hbar}{2m} \left[ -\nabla v - 2 \frac{m}{\hbar} u \cdot v \right] \quad \text{(eliminating \( m/\hbar \))}
\]
\[
\frac{\partial u}{\partial t} = -\frac{\hbar}{2m} \nabla^2 v - \nabla (u \cdot v) \quad \text{(equation 4.29)}
\]

\[
\frac{\partial S}{\partial t} = \frac{\hbar}{2m} \left[ \nabla^2 R + (\nabla R)^2 - (\nabla S)^2 \right] - \frac{1}{\hbar} \nabla V \quad \text{(equation 4.28)}
\]
\[
\nabla \frac{\partial S}{\partial t} = \nabla \frac{\hbar}{2m} \left[ \nabla \nabla R + (\nabla R)^2 - (\nabla S)^2 \right] - \frac{1}{\hbar} \nabla V \quad \text{(multiplying by \( \nabla \))}
\]
\[
\frac{m}{\hbar} \frac{\partial v}{\partial t} = \nabla \frac{\hbar}{2m} \left[ \nabla \left( \frac{m}{\hbar} u \right) + \left( \frac{m}{\hbar} u \right)^2 - \left( \frac{m}{\hbar} v \right)^2 \right] - \frac{1}{\hbar} \nabla V \quad \text{(replacing \( \nabla R \) and \( \nabla S \))}
\]
\[
\frac{\partial v}{\partial t} = \nabla \frac{\hbar}{2m} \left[ \nabla u + \frac{m}{\hbar} u^2 - \frac{m}{\hbar} v^2 \right] - \frac{1}{m} \nabla V \quad \text{(eliminating \( m/\hbar \))}
\]
\[
\frac{\partial v}{\partial t} = \frac{\hbar}{2m} \nabla^2 u + \frac{1}{2} \nabla (u^2) - \frac{1}{2} \nabla (v^2) - \frac{1}{m} \nabla V \quad \text{(equation 4.30)}
\]

\[
\frac{\partial v}{\partial t} \quad \square
\]

This completes the proof of theorem 4.16.

\[
\square
\]
5 Spacetime

Why is there a maximum speed in the universe? In this section, we investigate the general halting partition augmented with the power-time observable $P \times t$.

$$Z = \sum e^{-\beta(E-Pt-Fx)} \quad (5.1)$$

The units of the term $P \times t$ are $(J/s) \times s = J$ and the units of $F \times x$ are $(J/m) \times m = J$. Here $J$ is joules, $P$ is a power in Watts, $F$ is a force in newtons, $t$ is a time in seconds and $x$ is a distance in meters.

The fundamental thermodynamics state equation for this partition functions becomes,

$$dE = TdS - Pdt - Fdx \quad (5.2)$$

5.1 Light-cone

We work in the quasi static approximation

$$\Delta E = T \Delta S - P \Delta t - F \Delta x \quad (5.3)$$

We look at the thermodynamic cycle of the system transiting through time and space starting at $P_x$ to $P_0$ to $P_t$ to $P_x$ as illustrated on Figure 4. During the transitions and to keep the energy constant, tradeoffs must be made between time, distance and entropy. This cycle is reminiscent of other thermodynamic cycles such as those involving pressure and volume, etc.

We pose that $\Delta E = 0$ throughout the cycle.

$P_x$ to $P_0$: As we translate $P_x$ closer in space to $P_0$ while keeping the time fixed to $\Delta t = 0$, the entropy must decrease to compensate. Since entropy tends to increase, we conclude that objects have a tendency to resist being returned to the origin and are instead encouraged to expand away from each other.

$$\Delta S = \frac{F}{T} \Delta x \quad (\Delta x \geq 0)$$

$P_0$ to $P_t$: As we translate $P_0$ forward in time to $P_t$ while keeping the distance fixed to $\Delta x = 0$, the entropy must increase to compensate. We conclude that an object evolving forward in time is encouraged by entropic considerations.

$$\Delta S = \frac{P}{T} \Delta t \quad (\Delta t \geq 0)$$

Figure 4: A thermodynamic cycle through space, time and entropy as observables.
As we translate $P_x$ forward both in time and space to $P_t$, we have

$$\Delta S = \frac{P}{T} \Delta t + \frac{F}{T} \Delta x$$

and if $\Delta S = 0$, we have a translation at a constant speed given by

$$\frac{\Delta x}{\Delta t} = -\frac{P}{F}$$

We conclude that an object travelling at speed $-P/F$ is neither encouraged nor discouraged by entropic considerations. We will now look at this in more details.

### 5.2 Speed of light

**Theorem 5.4.** The maximum speed of any object is a unique constant ($c$).

**Proof.** Starting from equation 5.3 and posing $dE = 0$,

$$F dx = T dS - P dt$$

$$\frac{dx}{dt} = \frac{T dS}{F dt} - \frac{P}{F}$$

$$v = \frac{T dS}{F dt} - \frac{P}{F}$$

(5.5)

Note that the units for each term of equation 5.5 are meters per seconds. The equation therefore describes a speed.

Let us look at three cases:

1. If $|v| > |{-P/F}|$, then $dS/dt < 0$ and the entropy decreases with time. This violates the second law of thermodynamics.

2. If $|v| < |{-P/F}|$, then $dS/dt > 0$ and the entropy increases with time. This is fine.

3. If $|v| = |{-P/F}|$, then $dS/dt = 0$ and the entropy remains constant. This is also fine.

Since, according to the second law of thermodynamics, the average entropy cannot decrease with time, it follows that $P/F$ is the fastest speed possible for a given system. Hence,

$$\frac{P}{F} = k$$

Taking the characteristic Planck units, we obtain
the speed of light - here proven to be an upper bound enforced by
the second law of thermodynamics.

5.3 Lorentz transformation

As is well-known, a fixed maximum speed leads to special relativity
and the Lorentz transformations. But what does that tell us about the
entropy?

If we add two speeds together like this

\[ v_1 + v_2 = -c + \frac{T}{F} \frac{dS_1}{dt} - c + \frac{T}{F} \frac{dS_2}{dt} \]

\[ \neq v_f \]

we do not get a faster speed, but instead two particles with two
different speeds \( v_1 \) and \( v_2 \) forming a composite system. For the en-
tropy to be additive, it must be the case that the two systems it de-
scribes must be statistically independent. Since this is not the present
case, the entropies cannot be added linearly. Instead,

\[ v_f = \gamma (v_1, v_2) \]

5.4 General Relativity

Analogous to the Newtonian case, Eric Verlinde shows that, assum-
ing special relativity, the entropy derivation of gravity can be used to
recover general relativity. In the present paper, we have demonstrated
special relativity by proving that the speed of light is maximal. This
augments Eric Verlinde’s derivation to a theorem of the general halt-
ing partition. Here we present a sketch of his proof. He starts with a
gradient of bits over the surface of a sphere

\[ dN = \frac{dA}{Gh} \]

The mass of these bits over a surface is obtainable via integration

\[ M = \frac{1}{2} \int_S TdN \]

\[ = \frac{1}{4\pi G} \int_S e^\phi \nabla \phi dA \]

Where \( e^\phi \) is the red shift that he derives earlier in his paper. This
equation is known to be the natural generalization of Gauss’s law to
general relativity. He then argues that this is enough to recover the full description of general relativity. For brevity the proof will not be included. Instead we will refer to Eric Verlinde’s paper for the rest.

5.5 Quantum Field Theory

Starting from theorem 4.16 (Schrödinger’s equation) and theorem 5.4 (the speed of light as a maximum speed), Quantum field theory will be held to be recoverable using standard literature methods 20.

6 Conclusion

So why does the axiomless derivation work at all?

Sufficient reason, by virtue of which we consider that we can find no true or existent fact, no true assertion, without there being a sufficient reason why it is thus and not otherwise, although most of the time these reasons cannot be known to us.

–Gottfried Wilhelm Leibniz

The principle of sufficient reason of Leibniz suggests that for any fact to be true, there must be a sufficient explanation for it. If the principle holds, then it follows that positing an axiom as a requirement of the ToE violates the principle of sufficient reason. Therefore, if the universe does exist, then there must be an axiomless derivation of the laws of physics.

A summary of the argument presented goes as follows:

1. The axiomless theory (def 1.16) obtained from the primitive existence of language allows us to produce a universal theory of knowledge (def 1.68).

2. Using an antiquated term, a UTK is essentially the "world of reason" envisioned by Plato where one is free to pose any assumption and correctly follow these assumptions to their logical conclusions. This "world" is valid for any and all assumptions posed. It comprises all the theorems that require at least one axiom to prove.

3. This "world of reason" taken as a whole provides the necessary condition to recover the ToE. Indeed since the "world of reason" is provable primitively, it follows that any ToE must be a UTK.

4. Finally, part III of this work is the derivation of the laws of physics from a UTK, as a convenience taken to be first-order arithmetic. The proof holds for any theory sufficiently descriptive to be a UTK.

5. From the restrictions imposed on a ToE by being a UTK, we have recovered the halting partition; an expression of the ToE in its most general form, and we have shown that it is sufficient by itself and with no appeal to physical observations, to derive quantum mechanic, statistical physics, general relativity, a proof of a maximal speed, etc.

This is presented as evidence of the existence of an axiomless derivation of physics producing the ToE.

We iterate the equation corresponding to the ToE.
$$Z = \sum_{i=1}^{\infty} e^{-(\ln 2)\beta[T(i)+F+i_{\sigma}C_{\sigma}(i)]}$$

where each $i$ is a program running on a UTM with the corresponding thermodynamic observables.

Since both quantum mechanics and general relativity are recoverable from this equation and consistent with the axiomless arguments presented, it follows that it must be the ToE.

References


