# *A one page derivation of the Theory of Everything*

*Alexandre Harvey-Tremblay*

*aht@protonmail.ch*

*Published: August 16, 2017*

In a previous work I have derived the theory of everything (ToE) in a 74 pages paper[1]. To make the theory more accessible, in this work, I derive the equation for the ToE on one page. I then follow the derivation with a few pages of discussion.

[1] Alexandre Harvey-Tremblay. An axiomless derivation of the theory of everything. https://www.academia.edu/33079029/An_axiomless_derivation_of_the_Theory_of_Everything, 2017a

## *Derivation*

I use first order logic and first order arithmetic as my logical framework. I pose the following assumption;

**Assumption 1.** *When a mathematician proves a theorem in first order logic, he does so in the universe.*

We continue the derivation under the thesis that it follows from (1) that first order logic must be recoverable from the ToE in some way, or the mathematician could not use it. In fact this goes for every set of axioms that might be investigated by the mathematician. Using a sentence of first order logic, we would say that for any ToE this sentence must hold;

$$\forall k \forall t [k \vdash t \implies ToE \vdash (k \vdash t)] \tag{2}$$

*where k stands for set of axioms and t stands for theorem.*

| n | sentence |
|---|---|
| 1 | $k_1 \vdash t_1$ |
| 2 | $k_1 \vdash t_2$ |
| 3 | $k_2 \vdash t_1$ |
| 4 | $k_2 \vdash t_2$ |
| 5 | $k_1 \vdash t_3$ |
| ⋮ | ⋮ |
| $n$ | $[k \vdash t]_n$ |

Table 1: Since there are countably many sentences, they can be associated to a natural number in a one to one correspondance.

Is this restriction enough to constructively produce a ToE and recover the laws of physics? It turns that yes it is enough. How? First I consider that each $k \vdash t$ pair forms a sentence that I can map to a natural number $n$ (see Table 1). I then create a binary real number $\Omega$ defined as follows;

$$\Omega = \sum_{n=1}^{\infty} 2^{-E(n)-n} \qquad \textit{where, } E(n) = \begin{cases} 0 & [k \vdash t]_n \\ \infty & \text{otherwise} \end{cases} \tag{3}$$

This definition connects (2) to the halting probability of a prefix-free universal Turing machine[2] (UTM). We are almost there. To construct the ToE, it suffices to make the sum monotonically converge towards $\Omega$ as the calculation progresses. This can be done by adding the program-action to program-frequency observable $\mathcal{S}\tau$ to the sum.

[2] Gregory J. Chaitin. An algebraic equation for the halting probability. https://www.cs.auckland.ac.nz/~chaitin/berlin.pdf, 1988; and Ming Li and Paul Vitányi. An introduction to kolmogorov complexity and its applications. Springer, 1997

$$Z_\Omega = \sum_{n=1}^{\infty} 2^{-\beta[E(n)+Fn+\mathcal{S}\tau]} \tag{4}$$

This is it. I obtain an "entropic" UTM, which is enough to recover the laws of physics[3].

[3] Alexandre Harvey-Tremblay. On an entropic universal turing machine isomorphic to physics. https://www.academia.edu/34218736/On_an_entropic_universal_Turing_machine_isomorphic_to_physics, 2017b

## Discussion

I will now unpack the derivation and discuss it point by point.

### Universal language

Why pose assumption (1)? We consider all set of axioms and their corresponding theorems and we call that : *reason*. If *reason* is to exist in the universe, then it must be the case that *reason* is recoverable from the ToE, as the ToE explains everything in the universe.

The first order sentence (2) states that for all sets of axioms $k$ and all theorems $t$, if $k$ proves $t$, then it implies that the theory of everything proves that $k$ proves $t$. Sentence (2) a necessary consequence that *reason* is done in the universe.

As a result, any universe which contains *reason* must be bound by (2). This is the only statement from pure reason which bounds what the universe can be. Since this statement is enough to construct a ToE and recover the laws of physics, it follows that the laws of physics are logically implied by the existence of *reason*.

### Unpacking the sum

To see into more detail what happens in the sum, let us unpack (3) with an example. We get

$$\Omega = \sum_{n=1}^{\infty} 2^{-E(n)-n} \tag{5}$$

We recall that if the $n^{th}$ sentence is provable, then $E(n) = 0$, otherwise $E(n) = \infty$. Using example values for $E(n)$, we get

$$= 2^{-\infty}2^{-1} + 2^{-0}2^{-2} + 2^{-0}2^{-3} + 2^{-0}2^{-4} + 2^{-\infty}2^{-5} + ... \tag{6}$$

The presence of the negative infinity in the term of the exponential causes some terms to vanish to zero.

$$= 0_b + 0.01_b + 0.001_b + 0.0001_b + 0_b + ... \tag{7}$$

$$= 0.01110..._b \tag{8}$$

The end result is that we obtain a real number $\Omega$ where the value of its bits corresponds to the provability of its corresponding sentence. Indeed, if the $n^{th}$ bit of $\Omega$ is 1, then the $n^{th}$ sentence is provable, otherwise it is not. This connects $\Omega$ to the halting probability of a prefix-free universal Turing machine (UTM).

As the bits of the sum are related to the provability of theorems, then it follows that the bits are non-computable and in fact $\Omega$ is known to be an algorithmically random number. A UTM can attempt to calculate $\Omega$ by starting each program, and as they halt, add their contribution to $\Omega$. After an infinite amount of time, $\Omega$ will indeed be recovered.

However, the calculation does not converges towards $\Omega$ as it progresses and discontinuously yields $\Omega$ only at infinity. To see why, consider the case where the first zero-valued bit of $\Omega$ is at position $i$. Since the non-halting problem is unsolvable, at most the calculation of $\Omega$ differs from the real value of $\Omega$ by $2^{-i}$. The error rate only vanishes at infinity when all halting programs are known. To make the laws of physics come out, we must adjust the calculation so that it converges towards $\Omega$ during the calculation. In other words, the error rate must monotonically decrease during the calculation. As shown in the following section, this can be done with entropic dovetailing.

## *Universal Turing Machine*

How can a UTM calculate $\Omega$ without having its progress hang? Indeed, attempting to run programs on a UTM for the purposes of calculating $\Omega$ will have two hanging traps. First, if we start one program and wait for it to terminate before starting another one, the UTM will hang at the first non-halting program. Second, if we start each program in parallel, since there are infinitely many such programs, the UTM will never return to work on the first one. The solution is to dovetail programs.

**Definition 9** (Dovetailing)**.** *Dovetailing is a program execution strategy for a Turing machine to guarantee that progress will be made on arbitrarily-many programs even in the presence of non-halting programs.*

**Definition 10** (Standard dovetailing)**.** *Consider the case of standard dovetailing. First, we start the shortest program and perform one iteration. Then, we start the second program and perform one iteration on the first and second program. Then, we start the third program and perform one iteration on the first, second and third program. And so on. Using dovetailing, progress will eventually be made on every program and no program will cause the TM to hang.*

To convert (3) into a dovetailing calculation of $\Omega$, it suffices to add the program-action conjugate to program-frequency observable $\mathcal{S}\tau$ to the sum. We get equation (4).

What are these program observables and why are we allowed to add them? $\Omega$ is similar to a Gibb's ensemble of statistical physics. In fact, this similarity has been noted by other authors before [4]. Simple

[4] K. Tadaki. A statistical mechanical interpretation of algorithmic information theory. https://arxiv.org/pdf/0801.4194.pdf, 2008; John C. Baez and Mike Stay. Algorithmic thermodynamics. arXiv:1010.2067 [math-ph], 2010; and Ming Li and Paul Vitányi. An introduction to kolmogorov complexity and its applications. Springer, 1997

replacements (changing the name of the variables) are enough to switch back and forth between the two representations. The Gibb's ensemble compares to the halting probability as;

$$\begin{array}{cc} \text{Gibb's ensemble} & \text{Halting probability} \\ Z = \sum_x e^{-\beta(E+pV+Fx)} & \Omega = \sum_{n=1}^{\infty} 2^{-E(n)-n} \end{array} \tag{11}$$

To recover the Gibb's ensemble, a multiplication constant must to added to to the term of the exponential of the halting probability. This multiplication constant takes the role of an "algorithmic temperature". The equation becomes,

$$Z'_{\Omega} = \sum_{n=1}^{\infty} 2^{-\beta[E(n)+Fn]} \tag{12}$$

$E(n)$, as it is either 0 or $\infty$ will absorb $\beta$, so its contribution to $\Omega$ remains the same. For $\beta Fn$, the effect is to "compress" or "decompress" the bits of $\Omega$. If $\beta F > 1$, no bit erasure take places. Let us unpack the sum with $\beta F = 2$ as an example.

$$= 2^{-2\times1} + 2^{-2\times2} + 2^{-2\times3} + 2^{-2\times4} + 2^{-2\times5} + ... \tag{13}$$

$$= 2^{-2} + 2^{-4} + 2^{-6} + 2^{-8} + 2^{-10}... \tag{14}$$

$$= 0.01 + 0.0001 + 0.000001 + 0.000000001 + ... \tag{15}$$

$$= 0.0101010101... \tag{16}$$

As we can see, some zero-valued bits have been injected between the bits of $\Omega$. To recover $\Omega$, it suffices to eliminate the extra bits.

The last step to recover the ToE is to add observable(s) to the partition function so as to obtain a dovetailing algorithm able to calculate $\Omega$ with an ever improving approximation.

To do so, we introduce the program-action conjugate $\mathcal{S}$ and the program-frequency observable $\tau$. We obtain,

$$Z_{\Omega} = \sum_{n=1}^{\infty} 2^{-\beta[E(n)+Fn+\mathcal{S}\tau]} \tag{17}$$

As $\Omega$ is non-computable, then $\mathcal{S}$ must also be non-computable. Since the dovetailing algorithm is constructed exclusively via a Gibb's ensemble, the algorithm maximizes the entropy of $\Omega$ during the calculation. This will be key to recover the laws of physics.

Let us now prove the two requirements for this algorithm; 1) It is a dovetailing algorithm and 2) it recovers $\Omega$ when $t \to \infty$.

**Theorem 18.** *At the limit of $t \to \infty$, we can compute $\Omega$ from $Z_\Omega$.*

*Proof.* A program $n$ can have any value of $\mathcal{S}_n$ within $[0, \infty]$. If the program halts immediately, $\mathcal{S}_n = 0$. If it never halts, $\mathcal{S}_n = \infty$. If it halts after a certain time, $\mathcal{S}_n \in \mathbb{N}$. A program that never halts will not contribute to the halting partition. This will be the case if $\mathcal{S}_n = \infty$. As a result we obtain,

$$\lim_{\tau \to 0^+} \tau S_n = \lim_{t \to \infty} \frac{\mathcal{S}_n}{t} = \begin{cases} 0 & \text{n halts} \\ \infty & \text{otherwise} \end{cases} \tag{19}$$

As this is the definition of $E(n)$ (see 3), we obtain

$$\lim_{t \to \infty} \frac{\mathcal{S}_n}{t} = E(n) \tag{20}$$

**Lemma 21.** $E(n) + E(n) = E(n)$

*Proof.* $E(n)$ is either 0 or $\infty$. Since $0 + 0 = 0$ and $\infty + \infty = \infty$, the lemma holds. □

Therefore,

$$\lim_{t \to \infty} Z_\Omega = \lim_{t \to \infty} \left( \sum_{x=1}^{\infty} 2^{-\beta[E(x) + \mathcal{S}\tau + Fx]} \right) \tag{22}$$

$$= \sum_{x=1}^{\infty} 2^{-\beta[E(x) + E(x) + Fx]} \tag{23}$$

$$= \sum_{x=1}^{\infty} 2^{-\beta[E(x) + Fx]} \qquad \text{(Lemma 21)}$$

$$= Z'_\Omega \tag{24}$$

Can we compute $\Omega$ knowing $Z'_\Omega$? The answer is yes as we just need to remove the zero-valued bits inserted in between the bits of $\Omega$. □

**Theorem 25.** *To show that equation (4) dovetails programs, it suffices to show the following. For $0 < t < \infty$, the partition function $Z_\Omega$ is*

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

*where $2^{-k(t)}$ is an error rate that is monotonically decreasing to 0 as $t \to \infty$. As a result of increasing time, the calculation of $Z_\Omega$ produces an ever more precise estimation of $\Omega$.*

*Proof.* [5].

**Definition 26.** *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 $\Omega$.*

[5] Here, we have reproduced the definition of $k(t)$ and the proof provided by John C. Baez and Mike Stay in their paper on *algorithmic thermodynamics*.

John C. Baez and Mike Stay argues as follows:

The term $2^{-\mathcal{S}\tau}$ exponentially suppresses long program runtimes. Then because $-\frac{\mathcal{S}_x}{t}$ 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 $2^{-k(t)}$.

$\square$

For example, say

$$\Omega = 0.0111100\ldots \tag{27}$$

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 $\mathcal{S}_x = 50$ for this program. We obtain,

$$Z_x(t) = 2^{-x}2^{-\frac{\mathcal{S}_x}{t}} \tag{28}$$

$$Z_5(t) = 2^{-5}2^{-\frac{50}{t}} \tag{29}$$

$$= 0.00001 \times 2^{-\frac{50}{t}} \tag{30}$$

The halting probability $\Omega$ is,

$$\Omega = 0.0111000\cdots + Z_5(t) \tag{31}$$

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 $-(\mathcal{S}_5/t)$ is monotonically decreasing, the uncertainty $2^{-k(t)}$ must decrease monotonically to 0 as $t$ increases.

## *The laws of physics*

How then do we recover the laws of physics from (4)? To recover the laws of physics, we make use of the properties of the Gibb's ensemble notably by studying its state equation. Although not necessary, it helps to give (4) a physical interpretation of its observables. We will map the program-observables to physical-observables as follows.

- The program-runtime is the number of *Iterations* a UTM needs to perform until a program halts. It is therefore natural to associate it with the physical *Time* in *seconds*. Indeed, a program requiring more iterations to halt will also require more time to terminate. If a system performs iterations at a faster or slower rate, the conjugate variable to time, the *Power* in *Watts*, can be adjusted to account for this variation.

- Its inverse, the algorithmic-frequency, is associated with the reverse of the second, $s^{-1}$, and its conjugate variable is the *Action* in *Joules-seconds*.
- The program-size is expressed in number of *bits*. Writing the bits one after the other on any medium (paper, disk drive, etc.) will require a certain physical size for each bit. As the line is the lowest dimensional geometry to spread bits, the program-size is naturally associated with the physical *length* as its simplest case. Furthermore, if an encoding medium would allow greater or lesser "packing-tightness" of the bits, it can be modelled with its conjugate variable the *Force* in *Newtons* pushing the bits together or pulling them apart. If one wishes instead to investigate geometries of higher dimensions, one can use different units. For the $3D$ case, the program-size can be mapped to a *Volume* in $m^3$ and its conjugate variable will be the *Pressure* in $N/m^2$. For the $2D$ case, it can be mapped to an *Area* in $m^2$ and its conjugate variable will be the *Surface tension* in $N/m$.
- Only the halting event remains. As it is the only quantity with *no units*, it is natural to map it to the *Energy* in *Joules*. Indeed, in the Gibb's ensemble, the energy is the only observable not multiplied by a conjugate variable. Adding extra units to the halting event only to have them cancelled out by a conjugate variable would be futile.

Summarizing the points above, we obtain Table 2 as our mapping of choice between *algorithmic thermodynamics* and *physical thermodynamics*.

| Observable | Variable | Units | Conjugate | Variable | Units |
|---|---|---|---|---|---|
| Halting event | $E$ | $J$ | Temperature | $T$ | $K$ |
| Program-size (length) | $x$ | $m$ | Force | $F$ | $N$ |
| Program-size (area) | $A$ | $m^2$ | Stiffness | $\gamma$ | $N/m$ |
| Program-size (volume) | $V$ | $m^3$ | Pressure | $p$ | $N/m^2$ |
| Program-frequency | $\tau$ | $1/s$ | Action | $\mathcal{S}$ | $J \times s$ |
| Program-runtime | $t$ | $s$ | Power | $P$ | $W$ |

Table 2: The preferred correspondence between *algorithmic thermodynamics* and *statistical physics*.

### *State equation*

The state equation for the partition function (4) is,

$$dE = TdS - \mathcal{S}d\tau - Fdn \tag{32}$$

This is analogous to the law of conservation of energy, interpreted as a law of conservation of information. In the physical interpretation, the state equation is,

$$dE = TdS - \mathcal{S}d\tau - Fdx - \gamma dA - pdV \tag{33}$$

In a previous paper[6], I show how this state equation proves many of the laws of physics including,

1. A maximum speed of light.
2. Light-cone.
3. Lorentz's transformation.
4. Law of inertia.
5. General relativity.
6. A possible explanation for dark energy.
7. Spin.
8. Polarization.
9. Schrödinger's equation.
10. Dirac equation.
11. The arrow of time.
12. The quantum measurement.

[6] Alexandre Harvey-Tremblay. On an entropic universal turing machine isomorphic to physics. https://www.academia.edu/34218736/On_an_entropic_universal_Turing_machine_isomorphic_to_physics, 2017b

### *Maximum speed*

As an example, we will derive the maximum speed, the speed of light, from the state equation (33). We consider that the runtime is the reverse of the frequency and we pose $\tau = t^{-1}$ and $d\tau = -t^{-2}dt$.

$$\begin{aligned}
dE &= TdS + \mathcal{S}t^{-2}dt - Fdx - \gamma dA - pdV && \text{(State equation 33)}\\
TdS &= Fdx - \mathcal{S}t^{-2}dt && \text{(Posing } dE, dA \text{ and } dV \text{ to 0)}\\
\frac{T}{F}\frac{dS}{dt} &= \frac{dx}{dt} - \frac{\mathcal{S}}{t^2F} && \text{(Dividing by } Fdt\text{)}\\
\frac{T}{F}\frac{dS}{dt} &= \frac{dx}{dt} - \frac{P}{F} && \text{(Posing } P = \mathcal{S}/t^2\text{)}
\end{aligned}$$

Note that the units for each term are meters per second. The equation therefore relates a speed to a change of entropy.

Let us look at three cases:

1. If $\frac{dx}{dt} - \frac{P}{F} < 0$, then $\frac{dS}{dt} < 0$. The entropy decreases with time.
2. If $\frac{dx}{dt} - \frac{P}{F} > 0$, then $\frac{dS}{dt} > 0$. The entropy increases with time.
3. If $\frac{dx}{dt} - \frac{P}{F} = 0$, then $\frac{dS}{dt} = 0$. The entropy remains constant.

To understand why this implies a speed barrier at $P/F$, we must ask how can a UTM decrease the entropy of $\Omega$. To do so it must erase the value of a bit of $\Omega$, which violates the conservation of information. Hence any system which conserves information will have a characteristic power and a characteristic force which limits the speed of the system.

If the system is the universe, then taking $P$ to be the characteristic Planck power, and $F$ to be the characteristic Planck force, we do in fact recover the speed of light.

$$P\left(\frac{1}{F}\right) = \frac{c^5}{G}\left(\frac{G}{c^4}\right) = c \tag{34}$$

*Exfoliation of spacetime*

At the beginning of the calculation of $\Omega$, all bits are unknown. Hence the entropy is maximal. The entropy of the first $N$ bits of $\Omega$ is

$$S = Nk_B \ln 2 \tag{35}$$

As the calculation progresses and the value of the bits are obtained, the entropy is reduced by $k_B \ln 2$ for each bit known. As a result, we would expect the halting entropy go down with time. This would violate the second law of thermodynamics if it were not for the presence of other observables which can act as entropy sinks. The physical state equation has the $dx$, $dA$ and $dV$ observables which can be used as entropy sinks. As time goes forward and the halting entropy is reduced, the spacetime entropy is increased to compensate.

As shown in my paper on entropic UTM, it is from this exfoliation that we can recover the law of inertia, general relativity and dark energy, the Schrödinger's equation and the Dirac equation. Hence, spacetime is created by the second law of thermodynamics to offset the entropy reduction of irreversible time.

## *Conclusion*

The ToE presented herein provides a unification of the laws of physics. The unifying first order sentence is

$$\forall k \forall t [k \vdash t \implies ToE \vdash (k \vdash t)]$$

which requires that the ToE be universal in the sense that it can verify the theorems of any set of axioms. This is a sufficient condition to constructively create a ToE from pure reason that is able to recover the laws of physics. It does not seem to be possible to recover laws that are not sound (e.g. do not occur in nature). Hence, (2) does not over-predict.

## *References*

John C. Baez and Mike Stay. Algorithmic thermodynamics. arXiv:1010.2067 [math-ph], 2010.

Gregory J. Chaitin. An algebraic equation for the halting probability. https://www.cs.auckland.ac.nz/~chaitin/berlin.pdf, 1988.

Alexandre Harvey-Tremblay. An axiomless derivation of the theory of everything. https://www.academia.edu/33079029/An_axiomless_derivation_of_the_Theory_of_Everything, 2017a.

Alexandre Harvey-Tremblay. On an entropic universal turing machine isomorphic to physics. https://www.academia.edu/34218736/On_an_entropic_universal_Turing_machine_isomorphic_to_physics, 2017b.

Ming Li and Paul Vitányi. An introduction to kolmogorov complexity and its applications. Springer, 1997.

K. Tadaki. A statistical mechanical interpretation of algorithmic information theory. https://arxiv.org/pdf/0801.4194.pdf, 2008.