IN SEARCH OF PURPOSE

TOOLS FOR THE STUDY OF INTELLIGENT SYSTEMS THAT STUDY INTELLIGENT SYSTEMS

ALEXI PARIZEAU*, *GOLDSMITHS, UNIVERSITY OF LONDON*

I. INTRODUCTION

Why are we here*?* What's our purpose? These are questions perhaps as old as humanity. And yet no lasting answer has emerged to satisfy our curiosity. Some have argued that before we can hope to answer such probing questions, we must first know exactly what we're inquiring about. For instance, surely the question of "life's meaning" demands we first know what "life" and "meaning" are.[1] But what if this wasn't so? What if the question of "life's meaning" isn't so much about life and meaning as it is about the systems that can understand the intension of this question? If so, then I ask, what are those systems and how do we study them?

Before trying to answer this question, I suggest taking a step back, since hard problems can sometimes be solved by using new theoretical tools.[2] In particular, there seems to be some utility in first examining two elementary concepts in mathematics: 'nothingness' and 'being undefined'. By applying these concepts in the context of metamathematics, we can produce a metamathematical formulation of the empty set. This then helps to illuminate a simple procedure for detecting rules in recurrent systems. And with this procedure, we can define a function for measuring 'critical explainability' and then use it to show that a system is only a rational and reflexive semantic agent when it grows in critical explainability. I then posit that we are such agents and we have this characteristic. If true, this may be an answer to the question of life's meaning.

II. BUILDING A BETTER MOUSE TRAP: THE METAMATHEMATICAL EMPTY SET

We'll start by just defining the concepts and tools we'll need. The first is grounded on a notion that is quite well known by mathematicians, physicists and philosophers alike: the notion of *nothingness*.[3] In set theory, nothingness is represented as the *empty set*, while in other theories it's called the *null object*. For our purpose here, we'll just focus on the empty set.

So, what is the empty set? One way to think of it is as the set of what's logically impossible, such as the set of triangles that have four sides (Darling 2004, p.106). The prevailing definition though is the formula $\{x \mid x \neq x\}$, which also describes an impossibility. That this formula always returns nothing is critical, since without the inevitability of some impossibility, there could be no incorrect statements. And without incorrect statements, mathematics loses its defining characteristic.[4]

Hence, mathematics needs to have statements that are *necessarily* wrong. To define this better, we can use the concept of *consistency* in any *formal axiomatic system* (FAS). A FAS consists of a finite alphabet, grammar, axioms, rules of inference and proof-checking algorithm. It's then said that for any FAS to be "consistent", there must be a statement in that system that's logically impossible.[5] For

* Contact: alexi.parizeau@gmail.com

example, the statement that $x \neq x$ asserts that x is not equal to itself. We intuitively know this is impossible, since we intuitively adopt the cardinal axiom of logic (i.e. the Law of Identity), which says everything is the same with itself and different from another.

Now recall that an empty set can be defined by $\{x \mid x \neq x\}$. So we can notice that just having a theory that obeys the cardinal axiom of logic already presumes that $\{x \mid x \neq x\}$ must select nothing. That is, by assuming a consistent theory, we presume the empty set. This means we'll never be able to derive the empty set, as it'll get smuggled in by just forbidding contradictions. But while we can't derive it, we can still offer new ways to define it. My hope is then that by defining it better, we might better illuminate its role in logic and epistemology. And it's this deeper understanding of the empty set that I think may be helpful to the task of studying intelligent systems that study themselves.

To this end, I suggest defining the empty set so-as to explicitly expose its connection to formal consistency, while revealing its prevailing definitions to be a special case. The basic idea is to define the empty set as the set complementary to the formal language of a FAS. To achieve this, we'll need another elementary notion: *being undefined*. Essentially, 'being undefined' refers to what is not "accepted" by an automaton, or virtual machine, that computes all the theorems of an FAS. As example, let F be any FAS and M be an automaton that computes the theorems of F. The formal *language* of M is then the set $L(M)$ of all sequences of symbols that M accepts. We then say $\overline{L(M)}$ is the set *complementary* to $L(M)$, such that every $w \in \overline{L(M)}$ is *not accepted* by M, and so $w \notin L(M)$.

So given that every FAS g has a language set $L(g)$, we can ask: *what is selected by the formula:*

$$\{x : x \notin L(g)\}$$

Unless g accepts everything, this formula is guaranteed to select something. It's also guaranteed to select nothing that g can recognize. So to *at least* g, this looks like an empty set.

This formula has an issue though: there can be no algorithm that computes what gets selected for any g. Clearly, this holds in the case of self-reference, where g tries to accept a string it can't accept (e.g. the Halting problem). We'll come back to this. First, let's examine the issue. Consider, for instance, that it's always possible to design an automaton, or virtual machine, called a *Turing machine* (TM) with the same language as a FAS. We can then think of the language set of a program (e.g. the set of strings that the program "accepts" on) as the set of theorems of a FAS. So to tell if a sequence X is undefined to a given FAS, you just have to ask whether X is a member of the complement of the language of a TM that computes the theorems of that FAS. This would be unproblematic, except for the known fact that unless a TM always halts or accepts on no input or on every input, then questions about the members of its language are *generally undecidable* (that means there can't exist any single algorithm for all cases).[6] So there can be no algorithm that lists every member of $\{x : x \notin L(g)\}$ for any g, where g is a FAS that needs a TM to compute its theorems.

Fortunately, there's a type of automata that doesn't suffer – in theory – from undecidability: *deterministic finite automata* (DFA).[7] DFA can even simulate some TM, but only those TM that can use at most k tape cells, where k is independent of the number of cells used as input. This is due to the fact that DFA are by definition virtual machines that only use a finite number of states in whatever computation they perform. Because of this, every physical computer can be specified as a DFA, no matter its design, since every physical computer can only ever use a finite amount of physical resources in a finite number of time steps before the heat death of the universe.

So, in brief, if we want to answer any question about some $L(g)$, such as the complement set of $L(g)$, it helps if g is computable by a DFA. But even if g is computable by a DFA, no g can accept a sequence that's not in its language. So we must now talk about the case of self-reference.

For this next discussion, it'll be useful to introduce more notation. First, let's give our formula a symbol. I suggest the "null sharp" symbol $\emptyset^{\#}$ with a subscript to specify what the selection applies to. Hence, $\emptyset_g^{\#}$ means $\{x: x \notin L(g)\}$. Then, since this formula can be used by one FAS on another, let's allow the # symbol in the superscript to be replaced with the label for any system that can perform the set selection (e.g. a computer program that outputs the specified set). So when we need to express that some FAS f can use this formula on another FAS g, we can write $\emptyset_g^f$.

We might then ask, *but what if some f tries to perform $\emptyset_g^f$ where $f = g$*? Intuitively, no sequence X that is undefined to g can be recognized by g when g searches for what is not recognized by g. Hence, g must find *nothing*. So let's say that a set can be called the "*absolute empty set*" if and only if all consistent FAS must find nothing when trying to select it. It then follows that if we let g be any FAS, then the formula $\emptyset_g^g$ selects the absolute empty set if and only if g is consistent. The idea is simply: *if g looks for what's undefined to g and cannot find it, then not finding is the finding*.[8]

As proof, let g be any FAS, then notice that if $x \in \emptyset_g^g$ for any $x \notin L(g)$, then $x \in L(g) \wedge x \notin L(g)$, which is contradictory.[9] To fix the contradiction, it must be that $\forall x(x \notin \emptyset_g^g)$. Thus, the 'absolute empty set' is the *unique* result of using $\emptyset^{\#}$ self-referentially; that is, $\emptyset = \emptyset_g^g$ for any consistent g.

We need to be careful with this formula though, since using it non-self-referentially can be treacherous. Consider that since the formula $\emptyset_g^f$ can involve *two* FAS (i.e. f and g), the result is a "set" whose size might be disputed by f and g. This is somewhat unusual, since the practice of mathematics is mostly performed in a "domain-of-discourse" that allows only a single FAS. For instance, when evaluating 1+1, it is assumed implicitly that we are referring to the addition of two integers within a *single* standard arithmetic system. But to evaluate $\emptyset_g^f$ we need to explicitly refer to two particular FAS: an FAS g that's targeted by the formula and an FAS f that performs the formula. So in an inter-FAS "environment" where some f subsumes some g, the cardinality of $\emptyset_g^{\#}$ can be larger than zero for f (i.e. $|\emptyset_g^f| \geq 0$), but exactly zero for g (i.e. $|\emptyset_g^g| = 0$). This is because the size of the set $\emptyset_g^{\#}$ gets limited by the size of the formal language used, which leads to two FAS potentially disagreeing on $|\emptyset_g^{\#}|$ for any given g. This issue never goes away, as you can always find some FAS that sees $|\emptyset_g^{\#}|$ as being ever larger, for any g. So the formula $\emptyset^{\#}$ doesn't *generally* lead to the selection of *the* absolute empty set, but it does select it as a special case: when $\emptyset^{\#}$ is implemented self-referentially. Meanwhile in all cases the set selected by the formula $\emptyset^{\#}$ will always appear empty to at least one FAS (i.e. $|\emptyset_g^{\#}| = 0$ when selected by at least g, for any g, even though $|\emptyset_g^{\#}| \geq 0$ for some FAS that's not g). Therefore, we can say that $\emptyset^{\#}$ is a metamathematical formula for selecting "*quasi*-"empty sets whose cardinality is limited by the language of an FAS.

In summary, we can select "empty sets" that are only guaranteed to be empty with regards to the FAS whose complementary language is a subset of the set we selected from. This is then a metamathematical way of selecting empty sets such that the absolute empty set (i.e. the standard $\emptyset$) gets recovered as the special case where self-reference is employed.

III. A GENERAL PROCEDURE FOR DETECTING RULES IN RECURRENT SYSTEMS

To see how metamathematical empty sets might be useful to the study of intelligent systems, I suggest applying it in a procedure for probing unknown, interactive or only partially observable axiomatic systems, which we can refer to as 'black boxes'. In particular, we'll be considering systems that can be modeled as a function whose codomain is a subset of its domain. Such functions can be called recursive, since all their valid outputs are also valid inputs. An interesting property of these functions, which we intend to exploit, is that they can recur, or loop, indefinitely. We can then say that a recursive function 'recurs' when any valid output is fed to it as an input (e.g. in a feedback cycle). Since these functions can be simulated by computers, let's also say that a recursive function C recurred in time t if a simulation of C accepted its last output sequence after no more than t steps. We'll also say that the recurrence of C is 'terminated' if it does not accept an input it was given.

For example, let C be any computable recursive function whose last output sequence was $X = (a_1, a_2, \dots, a_k)$. Then, pick an a_i and change it to any symbol y such that $y \neq a_i$. Either C accepts the new sequence X' and recurs, or it doesn't and terminates. We can then prove that given any C computed by an automaton with language $L(C)$, where $L(C)$ is also the language of a consistent FAS and X is one of its theorems, then there always exists some $\emptyset_C^{\#} = \{X' \notin L(C)\}$. Proof: we know, by hypothesis, that $L(C)$ is the language of a consistent FAS. We also know, by definition, that an FAS is consistent only if there's a sequence of symbols it can't accept. So let X' be any such sequence. You can then always create a new FAS with a grammar rule $X \rightarrow X'$, or equivalently, some automaton M with production rules to transit from X to X'. Hence, there's always a way to select $X' \notin L(C)$.

So what does this mean? The basic idea is you can always find an automaton M that can help you probe the rules of an axiomatic system. This should sound familiar, because in context of causal graph theory (Pearl 2000), such an M is akin to performing repeatable 'interventions'. While in the context of metabiology (Chaitin 2012), M is like an induced 'algorithmic mutation'. In general though, any such M is just a mechanism for breaking the rules of recurrent axiomatic systems. And finding such M might be useful for probing any axiomatic systems without formal specifications. In particular, it might be useful for probing non-static axiomatic systems where new axioms get created, or where the grammars are in flux (e.g. any 'infinite game', like life, see (Carse 2011)).

To understand this better, recall that the main condition placed on C was that its output must always be a valid input. So when an automaton makes any number of transformations $a_i \rightarrow y$ that produces an input sequence X' we can detect whether it's in the language $L(C)$ by observing whether C can still recur after the change. If the change forms some X' where $X' \notin L(C)$, then the recurrence of C must be terminated since X' is not a valid input. Meanwhile, if the change does not form any $X' \notin L(C)$, the input is valid and so a recurrence of C can be observed in time t. Essentially, if X' is a sequence of symbols that breaks the rules of C, then some M made a change to X that effectively ended any recurrence of C. Hence, if we can observe the regular recurrence of a function and interfere with its input, we can probe for rules and axioms by trying to break them.

For example, consider the effects of some gene family which induces hypermutation in a host leading to critical degradation of its genome beyond the threshold of replication. Here, we can let C be an automaton that models host replication using a genome X and we can let M be an automaton that models hypermutation on C's genome. From this, we can easily see that if M produces any mutation $\emptyset_C^M$, such that $\emptyset_C^M = \{X' \notin L(C)\}$, then M effectively terminates the recurrence of C.

Hence, metamathematical empty sets, such as $\emptyset_C^M$, can also represent transformations $a_i \to y$ *that terminate the descending lineage of the ensemble that contained* a_i (Hull 1980, p.327-329). Thus, since any accurate theory of life must predict such termination events,[10] metamathematical empty sets appear suitable for studying living systems, along with systems similar to life.

Now, any system with rules is, in principle, *explainable*. So if some a_i is critical to any recurrent system C, then the termination of the recurrence of C can in principle, if not in practice, be explained by pointing out that there's an automaton, or virtual machine, that can computes C, and that this automaton has a rule that is broken by some $a_i \to y$. Hence, the explanations, or reasons, for such elements a_i can be probed by repeatedly varying, perturbing or otherwise eliminating a_i, which has the effect of repeatedly showing that a_i is critical to C. It'll be convenient to refer to any such element a_i for any given input sequence X, so I'll write "$E(a_i)$" to indicate that the element a_i of a given input X has the property of being '*critically explainable*'. Finding *critically explainable* a_i then means finding some variable that can cause the catastrophic extinction of an otherwise stable-looking theory, such that the continued stability of such a theory is represented as a recurring function, and the catastrophic event is represented by its termination.[11]

In summary, if you are able to *interfere* with a recurring system whose output must be a valid input, then you can detect the existence of rules by identifying elements a_i such that $E(a_i)$. That is, if part of a recurring function's output is critical to it having a recognizable input, then disrupting that part would effectively terminate its recurrence. We then say that that part can be critically explained by the fact that disrupting it broke some set of rules.

IV. OUR FINAL INSTRUMENT — BEWARE, HERE BE DRAGONS

We've now seen how to generally detect the presence of rules in recurrent systems. But to study *intelligent* life, detecting the presence of rules isn't enough. We'll need one more tool, built from two functions. These rely on the procedure from the last section. Crucially, they're defined using an "oracle". An oracle is simply any black box system capable of answering certain types of questions that are posed to it. In what follows, we are allowed to choose any oracle we like, so long as it has (1) a formal language, and (2) the ability to decide if two or more sequences of symbols are deemed "equivalent", such that when presented with a set of sequences, it tells us how to group them into piles, where each pile represents an equivalence class. Essentially, this means that the oracle can be any black box that performs classification tasks on the data it's given. We can then define three functions based on mappings determined by whichever oracle $\mathcal{M}$ we chose.

To start, let's have the symbol $\delta_{\mathcal{M}}$ represent the first function. Note that $\delta_{\mathcal{M}}$ has a subscript for the oracle used to define the function. What we then want is for $\delta_{\mathcal{M}}$ to take a sequence $x \in L(\mathcal{M})$ of symbols as its input, such that x is interpreted by $\mathcal{M}$ as some recursive function C. We then want $\delta_{\mathcal{M}}(x)$ to return a set of equivalence classes for all valid input sequences w of C such that $w \in L(\mathcal{M})$ and w has elements a_i such that $E(a_i)$ for C. For intuition, I suggest thinking of $\delta(x)$ as a measure of the '*critically explainable complexity*' of x. If $|\delta(x)| > 1$ we then say x is "complex", else it is "simple".

The second function, which we can think of as '*critically explainable generality*', can be assigned the inverse symbol ${\delta_{\mathcal{M}}}^{-1}$. We then want ${\delta_{\mathcal{M}}}^{-1}(x)$ to return a set of equivalence classes for all recursive functions $C \in L(\mathcal{M})$ for which the input sequence x contains elements a_i such that $E(a_i)$ for C. Again, for the sake of intuition, if $|\delta^{-1}(x)| > 1$ we can say x is "general", else it is "particular".

The third function, which we can think of as '*critical explainability*', can be assigned the symbol $\mathrm{K}_{\mathcal{M}}$. We simply need it to return the Cartesian product of the first two functions, which can be written as $\mathrm{K}_{\mathcal{M}}(x) = \delta_{\mathcal{M}}(x) \times {\delta_{\mathcal{M}}}^{-1}(x)$, where "×" just means that every member of the set $\delta_{\mathcal{M}}(x)$ gets combined with every member of the set ${\delta_{\mathcal{M}}}^{-1}(x)$. [12]

Importantly, in order to ever have $\mathrm{K}_{\mathcal{M}}(x) \neq \emptyset$, $\mathcal{M}$ needs to be able to interpret x as a recursive function. To do this, a function must be encoded as sequences of symbols in the language of $\mathcal{M}$, else $\mathcal{M}$ cannot recognize it. This shouldn't be too much trouble though, since if a recursive function takes only inputs of finite lengths and returns outputs of only finite lengths, then it can always be written as a sequence of symbols of finite length (for proof, write the function as a table lookup).

It should also be underlined that $\delta_{\mathcal{M}}(x)$ and ${\delta_{\mathcal{M}}}^{-1}(x)$ both need to return a set of equivalence classes based on any given sequence of symbols $x \in L(\mathcal{M})$. In the case of $\delta_{\mathcal{M}}(x)$, the parameter x is interpreted as a recursive function per the rules of $\mathcal{M}$. It then returns the set of equivalence classes for all inputs w with some element a_i such that $E(a_i)$ for the function x. Meanwhile, the function ${\delta_{\mathcal{M}}}^{-1}(x)$ does the inverse: it interprets x as an input sequence and returns the set of equivalence classes of all recursive functions C where x has some element a_i such that $E(a_i)$ for C.[13]

Confusing as this may be, what's important to understand for our purpose here is that we can pick an oracle $\mathcal{M}$ that defines the function $\mathrm{K}_{\mathcal{M}}$. We may then ask, *what is so special about* $K_{\mathcal{M}}$? Well, if we choose the oracle just right, then $\mathrm{K}_{\mathcal{M}}$ can always return a finite and non-empty set for any input that's in the oracle's language. What this means, essentially, is that the oracle is telling you all the critically explainable classifications for any input you give it. Depending on the oracle, this can be quite interesting (e.g. if the oracle's language is 'open', or constantly evolving). It might get especially interesting if the oracle is capable of conforming its classifications to new theories it learns. But for just our purpose here we needn't worry about which types of oracles determine which kinds of functions. The only kind of oracle we presently need is just an 'ideal' one; that is, an oracle $\mathcal{M}$ where $\mathrm{K}_{\mathcal{M}}(x)$ always returns finite and non-empty sets for any $x \in L(\mathcal{M})$.[14] Though it may not be obvious, this puts a big constraint on the language of $\mathcal{M}$, since every $x \in L(\mathcal{M})$ must then be interpretable as a recursive function, else ${\delta_{\mathcal{M}}}^{-1}(x)$ would be null and so $\mathrm{K}_{\mathcal{M}}(x)$ would also be null, due to the Cartesian product. But if we are allowed to choose any oracle $\mathcal{M}$, including ideal ones, then we're finally ready to talk about studying intelligent agents that study themselves.

V. The Ultimate Question

We've now completed the task of defining our tools. Let's now use them to express a necessary characteristic of intelligent systems that study themselves. We'll then return to the perennial question of "*Why are we here?*" and see if this characteristic might provide us with a novel answer.

To start, let's have $\mathcal{T}_{life}$ represent any theory of life that has a language $L(\mathcal{T}_{life}) \subseteq L(\mathcal{M})$ with respect to the oracle $\mathcal{M}$ we picked. This means we can pick any formal theory where every valid sentence represents some valid instance of life, so long as it's in the language of $\mathcal{M}$. Assuming this, the set $\mathcal{L}$ of all critically explainable life, for our oracle $\mathcal{M}$ who learned $\mathcal{T}_{life}$, is simply defined as:

$$\mathcal{W} = \{x \in L(\mathcal{M}) \mid \mathrm{K}_{\mathcal{M}}(x) \neq \emptyset\},$$
$$\mathcal{L} = L(\mathcal{T}_{life}) \cap \mathcal{W}.$$

So are we done? Is it enough to just have $\mathcal{M}$ and $\mathcal{T}_{life}$, then study the reasons for each $\ell \in \mathcal{L}$? No, unfortunately not. Consider that even if we had an oracle that knew a definitive theory of life, it still wouldn't be enough to explain why we search for purpose. That's because the question is only meaningful to agential systems that are capable of seeking purpose (Metz 2013). So just knowing the members of $\mathcal{L}$ isn't enough, since not all $\ell \in \mathcal{L}$ search for their own purpose. Hence, to study the question of "life's meaning", we need to select for what can ask this question. That is, we need to answer: *"What is "the meaning of life" to systems that can understand the intension of this question?"*

To address this, let $\mathcal{T}_{semantic}$ be any theory where every $x \in L(\mathcal{T}_{semantic})$ specifies a system x that, while being properly constituted, *will continually create new syntax and semantics, while maintaining the capability to self-reflect, such that* $K_x(x) \neq \emptyset$.

Then let $\mathcal{T}_{rational}$ be any theory such that every $x \in L(\mathcal{T}_{rational})$ specifies an "agent" that selects available actions expected to maximize one of its internalized performance measures, taking into account what knowledge it has and what it has experienced (Russell & Norvig 2010, p.36).

Then let $\mathcal{I}$ be the set of all '*rational & reflexive semantic agents*' recognized by our chosen oracle $\mathcal{M}$ that's properly equipped with theories $\mathcal{T}_{semantic}$ and $\mathcal{T}_{rational}$. Formally, the set can be given by:

$$\mathcal{I} = L(\mathcal{T}_{semantic}) \cap L(\mathcal{T}_{rational}) \cap \mathcal{W}.$$

All we have left to do now is define a measure of time. It has been suggested that humans tend to experience time as discontinuous (Libet 2009). So I propose that we employ a special clock that advances one tick, or time step, only when our chosen oracle $\mathcal{M}$ is presented with an input. That is, our clock is a counter of $\mathcal{M}$'s input events. Any real time that passes in-between these clock ticks is ignored. For instance, we'll say that every time t corresponds exactly to the index of an input of $\mathcal{M}$.

Granted the above, I now offer a theorem that states a necessary characteristic of all $i \in \mathcal{I}$:

Every $i \in \mathcal{I}$ *arranges for* $|K_{\mathcal{M}}(i)|$ *to monotonically increase over finite time intervals, where* $\mathcal{M} = i$.

I believe that the reason for why this must hold is rather illuminating, so let me now sketch why it must be so. Assume, as hypothesis, that $i \in \mathcal{I}$. If so, then it must also be that $\mathrm{K}_i(i) \neq \emptyset$, else i could not self-reflect and thus $i \notin L(\mathcal{T}_{semantic})$ and so $i \notin \mathcal{I}$. We can also infer that i is a non-static FAS, else i can't be creative (Chaitin 2005) and thus $i \notin L(\mathcal{T}_{semantic})$ and so $i \notin \mathcal{I}$. From this we can infer that the size of the language $L(i)$ must increase in time, else i could not accept previously undefined syntax and semantics and thus $i \notin L(\mathcal{T}_{semantic})$ and so $i \notin \mathcal{I}$. We also know, by definition, that K_i can be used to find critically explainable elements of any FAS, including non-static FAS. Hence, if $L(i)$ has a new member due to being a non-static FAS, there must be at least one new equivalence class, hence $\mathrm{K}_i(i)$ return sets of increasing cardinality. Finally, we can infer that $|\mathrm{K}_i(i)|$ is monotonic increasing, since i must remember past equivalences, else it can't take into account past experiences and thus $i \notin L(\mathcal{T}_{rational})$ and so $i \notin \mathcal{I}$. The claim then follows.

If we were so capable, we might then wish to verify this by computing $\mathrm{K}_i(i)$ for some artificial agent i over multiple time intervals. For instance, we could use a sequence $(i_t, i_{t+1} \dots, i_{t+k})$, over any interval $[t, t+k]$, where $t, k \in \mathbb{N}$. We could then let $k(i_t) = \left|\mathrm{K}_{i_t}(i_{t-1})\right|$ and check if the derivative of k (if it has one) is larger than zero for all t in $(t, t+k)$. Notice though that the oracle which determines growth in $k(i_t)$ is i_t. This means that every i_t can, in principle if not in practice,

control whether it will continue to be a rational and reflexive semantic agent at time $t + 1$. I believe this then models the sense of agency we sometimes feel over our rationality.

So, assuming no errors in the above theorem, we are finally ready to tackle the question of "*Why are we here?*". Reformulating the question using our new tools, the question then becomes: *"What is the operational function of any $i \in \mathcal{I}$?"* To which we can reply:

The operational function of any $i \in \mathcal{I}$ is to increase in $K_i(i)$.

How can you interpret this? I suggest you assume that it is *you* who are the oracle in all the above definitions. Then, at those moments where you qualify as a rational and reflexive semantic agent, it follows that you are a member of the set $\mathcal{I}$. This answer then simply says that your higher-purpose, or meta-purpose, is just to grow in K. That is, your ultimate reason for being is just to increase your own critical explainability during those times that you are rational and self-aware.

Now, this answer is wrong, or not even wrong, if there's any fatal flaw in the definitions of the tools employed here. But even if no such flaw is found, it's still an open question as to whether this interpretation of the function K is sound. Hence, what is of some interest now is whether K can be computed by rational and reflexive semantic *artificial* agents, since by using artificial agents, these questions might be further studied with computational accuracy. I leave this to future work.

VI. WAIT, BUT HOW?

Thus far I have said little about exactly *how* any $i \in \mathcal{I}$ might grow in $\mathrm{K}_i(i)$, focusing instead on 'why' and 'what'. To answer 'how', we only need an interpretation of the function K, so let me suggest one that seems at least plausible at present.

Let any $i \in \mathcal{I}$ (over some time interval) be any person (e.g. yourself). Then, for any x, we know that the size of $\mathrm{K}_i(x)$ is dependent on the Cartesian product $\delta(x) \times \delta^{-1}(x)$. So if either function returns null, the product must also be null and so $\mathrm{K}_i(x) = \emptyset$. We also know that the larger either set is, so too must the product be. So if some person i is to persist as a rational and reflexive semantic agent, they must grow either $\delta(i)$, or they must grow $\delta^{-1}(i)$, or both.

Now recall that $\delta(i)$ is a measure of i's critically explainable complexity, while $\delta^{-1}(i)$ is a measure of i's critically explainable generality. So some i might increase their generality by forming new recurring functions that depend on their own selves (e.g. by finding an activity or place in the world that depends on their continued existence). This might be done by increasing the unique recurrent functions that depend on i, such that the loss of i terminates the function's recurrence. In this sense, increasing one's generality is essentially increasing one's criticality to new recurring functional dependencies. This might then explain why goal-seeking is so often conflated with life's meaning, since to satisfy life's meaning, we often create dependencies on ourselves.

We also needn't just grow $\delta^{-1}(i)$. We can also grow $\delta(i)$; that is, we can grow our critically explainable complexity, and this might be done by simply creating original syntax and meanings that expand the language of the recursive function i which may represent our "self".

We can also try to grow our $\delta(i)$ and $\delta^{-1}(i)$ in tandem, balancing both to strive toward the theoretical maxima of K. In the end though, however we increase in critical explainability, if we do increase in it, we satisfy life's meaning, no matter how we happened to go about it.

VII. WAIT, BUT WHAT IF IT'S TRUE?

A final concern is, *what if this isn't all wrong*? What if the function K can one day be experimentally shown to model properties of rational and reflexive semantic agents? What then? Alarmingly, irrespective of whether the above answer to the "ultimate question" is true, a greedy search algorithm that maximized K on just a single agent could turn out to be catastrophically immoral, made all the worse if maximizing K turns out to be meaningless. To mitigate this risk, we would need to solve how rational and reflexive semantic agents can be motivated to use only sane and ethical search strategies for growing K.[15] For if growing in K is meaningful and if algorithms to grow K are possible, then the algorithms we design for ethically growing K could very well be the most meaningful algorithms of our lives.

References

Bak, P., 2013. *How Nature Works: the science of self-organized criticality*, Springer Science & Business Media.

Barrow, J.D., 2009. *The Book of Nothing: Vacuums, Voids, and the Latest Ideas about the Origins of the Universe*, Knopf Doubleday Publishing Group.

Bentley, P., 2002. *Digital Biology: The Creation of Life Inside Computers and How It Will Affect Us*, Headline.

Carse, J., 2011. *Finite and Infinite Games*, Simon and Schuster.

Chaitin, G., 2005. *Meta Math!: The Quest for Omega*, Vintage.

Chaitin, G., 2012. *Proving Darwin: Making Biology Mathematical*, Vintage.

Darling, D., 2004. *The Universal Book of Mathematics: From Abracadabra to Zeno's Paradoxes*, John Wiley & Sons.

Deacon, T., 2011. *Incomplete Nature: How Mind Emerged from Matter*, W. W. Norton.

Deutsch, D., 2011. *The Beginning of Infinity: Explanations That Transform the World*, Penguin Group.

Dretske, F.I., 1981. *Knowledge and the Flow of Information*, MIT Press.

Godfrey-Smith, P., 2009. *Theory and Reality: An Introduction to the Philosophy of Science*, University of Chicago Press.

Goldstein, J. & Harris, S., 2015. Questions Along the Path: A Conversation with Joseph Goldstein. *Waking Up With Sam Harris*. Available at: https://soundcloud.com/samharrisorg/joseph_sam_2 [Accessed February 25, 2017].

Hersh, R., 1997. *What Is Mathematics, Really?*, Oxford University Press.

Holt, J., 2012. *Why Does the World Exist?: An Existential Detective Story*, W. W. Norton.

Hopcroft, J.E., Motwani, R. & Ullman, J.D., 2001. *Introduction to Automata Theory, Languages, and Computation* 2nd ed., Pearson Education.

Hull, D.L., 1980. Individuality and Selection. *Annual Review of Ecology and Systematics*, pp.311–332.

Kaplan, R., 1999. *The Nothing that Is: A Natural History of Zero*, Oxford University Press.

Libet, B., 2009. *Mind Time: The Temporal Factor in Consciousness*, HUP.

Marletto, C., 2015. Constructor Theory of Life. *Journal of The Royal Society Interface*.

Metz, T., 2013. The Meaning of Life. In E. N. Zalta, ed. *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University.

Nagel, E. & Newman, J.R., 1958. *Godel's Proof*, Routledge.

Parfit, D., 2011. *On What Matters*, Oxford University Press.

Pearl, J., 2000. *Causality*, Cambridge University Press.

Popper, K., 1962. *Conjectures and Refutations*, Routledge.

Pross, A., 2012. *What is Life?: How chemistry becomes biology*, OUP.

Rovelli, C., 2017. Meaning and Intentionality = Information + Evolution. *preprint*.

Russell, S.J. & Norvig, P., 2010. *Artificial Intelligence: A Modern Approach*, Prentice Hall.

Seife, C., 2000. *Zero: The Biography of a Dangerous Idea*, Penguin Books.

Tegmark, M., 2015. Friendly Artificial Intelligence: the Physics Challenge, *Artificial Intelligence and Ethics: Papers from the 2015 AAAI Workshop*.

Weinberg, S., 2015. *To Explain the World: The Discovery of Modern Science*, HarperCollins.

Technical Endnotes & Further Reading

[1] See (Tegmark 2014) for why it's believed that a physical definition of "life" and "meaning" is key.

[2] See (Weinberg 2015). But note that not all scientific progress relies on new mathematical tools, see (Godfrey-Smith 2009).

[3] For overviews of nothingness, see (Kaplan 1999; Seife 2000; Barrow 2009; Holt 2012).

[4] For more on why mathematics is the subject where errors are inevitable, see (Hersh 1997, Ch.3).

[5] For more on the definition of consistency, see (Nagel & Newman 1958). Specifically, the technical definition is that a formal system is internally consistent if there is at least one formula that's not derivable from the axioms of a formal axiomatic system.

[6] For details on recursively enumerable languages, see (Hopcroft et al. 2001, Ch.9).

[7] See (Hopcroft et al. 2001) for technical guidance on how to select everything undefined to a deterministic finite automaton (DFA). Since it's fairly straightforward, let me demonstrate. Let E be the DFA specified by the 5-tuple $(Q, \sum, \delta, q_0, F)$, where Q is a finite set of states, $\sum$ is a finite input alphabet, δ is a transition function, q_0 is the starting state such that $q_0 \in Q$, and F is a set of accepting states such that $F \subseteq Q$. The language of this DFA is then the set of all accepted input sequences of E (i.e. the set of all strings of symbols that allow the state of E to transit to a state in F). Now we can select part of the 'undefined domain' of E by constructing the DFA labeled $\bar{E}$ which we specify with the 5-tuple $(Q, \sum, \delta, q_0, Q - F)$. Here we see that $\bar{E}$ has a language that is the complement to $L(E)$ with respect to the alphabet $\sum$. Since no $w \in L(\bar{E})$ is a member of $L(E)$, the language $L(\bar{E})$ is a subset of the undefined domain of E. That is, $L_{undef}(E) \supseteq \{w \in L(\bar{E})\}$. To select a larger set, we can take the complement of $L(E)$ with respect to a superset alphabet of $\sum$. For instance, for language $L(E) \subseteq \sum_1^*$ such that $\sum_1 \subset \sum_2$, the complement $\overline{L(E)}$ is $\sum_2^* - L(E)$. This still doesn't select everything that's undefined to E though. The complete set is only selected by the union of sets of all strings over all possible alphabets, minus the language of E. Formally we can write this as $L_{undef}(E) = \{\bigcup_i \sum_i^* - L(E)\}$. Hence, to select everything undefined to E, we can use the formula $\{w \in L_{undef}(E)\}$. Or, as appears in the main text, we can use the formula $\{w \notin L(E)\}$.

[8] See (Goldstein & Harris 2015) for some interesting discussion on the philosophy of 'not finding'. In particular, note the discussion regarding Dzogchen master Tulku Urgyen Rinpoche who was fond of saying, with regards to the concept of 'emptiness', that "The not finding is the finding."

[9] For technical details on the halting problem, see (Hopcroft et al. 2001, Ch.9).

[10] For some relevant reading on contemporary theories of life, I recommend (Deacon 2011), which says that life can be represented by 'emergent hypercycles exhibiting self-rectyfing constraint preservation'. I also recommend the theory of life as systems which achieve 'dynamic kinetic

stability' over time, see (Pross 2012), and the constructor theory of life that explains how current physics allows for the evolution of accurate self-reproducers, see (Marletto 2015), and the meta-theory of life as 'evolving software', see (Chaitin 2012). Note that in all these theories, the terms for "life" can be expressed using recursive functions over a descending lineage of an ensemble such that failure to recur is interpreted as a termination event.

[11] For philosophies that seem closely related to the notion of 'critical explainability', see (Popper 1962); see also (Deutsch 2011). This notion also seems to connect to the notion of "noisy channels" as employed by Dretske in his investigation of information flows (Dretske 1981). For a brief, if somewhat trivial, illustration of these connections, let a *noisy channel* be any recursive function C that recurs at each time step while a message is in transit such that there's probability 1 of the message sequence X being changed to some X' during that time interval. The probability of losing the message is then 1 when $E(a_i)$ for all elements a_i of X, since by definition those are the elements that terminate C when varied. Hence, *for any noisy channel, the loss of a message is maximally predictable when all its elements are critically explainable.*

[12] After preparing this present work, I became aware of Rovelli's recent study of a physical definition of 'meaningful information' (Rovelli 2017). I believe the function $\mathrm{K}_{\mathcal{M}}$ is related to that definition, though examining exactly how would be a topic for another day.

[13] An interesting technical note is that we can define an extended version of the functions δ and δ^{-1} to take *sets* of sequences as arguments, thus allowing for compositions like $\delta^{-1}(\delta(x))$. Indeed, this seems far more natural, but it would also be more complicated, and unnecessary for this essay.

[14] In the body of the essay, I didn't provide much detail about what it means for an oracle to be 'ideal'. For technically-minded readers, let me now provide a more rigorous definition. An 'ideal' oracle $\mathcal{M}$ is one where you can always find an $n \in \mathbb{N}$ such that $0 < |\,\mathrm{K}_{\mathcal{M}}(x)| < n$ for any x that $\mathcal{M}$ can interpret as a recursive function. This is useful, since if $\mathrm{K}_{\mathcal{M}}(x)$ returns an infinite set, we can't use it as a measure. And if $\mathrm{K}_{\mathcal{M}}(x)$ returns null, then our oracle has no critical explanation for x, implying $\mathcal{M}$ isn't perfect. Essentially, $\mathrm{K}_{\mathcal{M}}(x) = \emptyset$ means either x is not an input sequence that contains an element a_i such that $E(a_i)$ for any recursive function known to $\mathcal{M}$ or x doesn't encode a recursive function that can be terminated by any means known to $\mathcal{M}$. In either case, $\mathrm{K}_{\mathcal{M}}(x) = \emptyset$ signals either x is not critically explainable, or $\mathcal{M}$ is not a perfect oracle. But if x encodes a consistent function, it must determine critically explainable inputs. And since x is just a sequence of symbols, it must have FAS that it's critical to. Hence, if x specifies a consistent FAS and $x \in L(\mathcal{M})$, yet $\mathrm{K}_{\mathcal{M}}(x) = \emptyset$, then the fault is with $\mathcal{M}$ and so $\mathcal{M}$ isn't an ideal oracle.

[15] For recommended reading with regards to how we might ethically grow in K, I suggest starting with Parfit's final thesis that *everyone ought to follow the principles that produce the maximum good consequences because it is only these optimific principles that everyone would have sufficient reasons to choose, and could therefore rationally choose*. See (Parfit 2011).