The Optimal Choice of Hypothesis Is the Weakest, Not the Shortest

Michael Timothy Bennett [1]

[1]The Australian National University

October 30, 2023

Abstract

If A and B are sets such that A is a subset of B, generalisation may be understood as the inference from A of a hypothesis sufficient to construct B. One might infer any number of hypotheses from A, yet only some of those may generalise to B. How can one know which are likely to generalise? One strategy is to choose the shortest, equating the ability to compress information with the ability to generalise (a "proxy for intelligence"). We examine this in the context of a mathematical formalism of enactive cognition. We show that compression is neither necessary nor sufficient to maximise performance (measured in terms of the probability of a hypothesis generalising). We formulate a proxy unrelated to length or simplicity, called weakness. We show that if tasks are uniformly distributed, then there is no choice of proxy that performs at least as well as weakness maximisation in all tasks while performing strictly better in at least one. In other words, weakness is the pareto optimal choice of proxy. In experiments comparing maximum weakness and minimum description length in the context of binary arithmetic, the former generalised at between *1.1* and *5* times the rate of the latter. We argue this demonstrates that weakness is a far better proxy, and explains why Deepmind's Apperception Engine is able to generalise effectively.

The Optimal Choice of Hypothesis Is the Weakest, Not the Shortest

Michael Timothy Bennett[1][0000−0001−6895−8782]

The Australian National University michael.bennett@anu.edu.au
http://www.michaeltimothybennett.com/

Abstract. If A and B are sets such that $A \subset B$, generalisation may be understood as the inference from A of a hypothesis sufficient to construct B. One might infer any number of hypotheses from A, yet only some of those may generalise to B. How can one know which are likely to generalise? One strategy is to choose the shortest, equating the ability to compress information with the ability to generalise (a "proxy for intelligence"). We examine this in the context of a mathematical formalism of enactive cognition. We show that compression is neither necessary nor sufficient to maximise performance (measured in terms of the probability of a hypothesis generalising). We formulate a proxy unrelated to length or simplicity, called weakness. We show that if tasks are uniformly distributed, then there is no choice of proxy that performs at least as well as weakness maximisation in all tasks while performing strictly better in at least one. In other words, weakness is the pareto optimal choice of proxy. In experiments comparing maximum weakness and minimum description length in the context of binary arithmetic, the former generalised at between 1.1 and 5 times the rate of the latter. We argue this demonstrates that weakness is a far better proxy, and explains why Deepmind's Apperception Engine is able to generalise effectively.

Keywords: simplicity · inference · general intelligence.

1 Introduction

If A and B are sets such that $A \subset B$, generalisation may be understood as the inference from A of a hypothesis sufficient to construct B. One might infer any number of hypotheses from A, yet only some of those may generalise to B. How can one know which are likely to generalise? According to Ockham's Razor, the simpler of two explanations is the more likely [1]. Simplicity is not itself a measurable property, so the minimum description length principle [2] relates simplicity to length. Shorter representations are considered to be simpler, and do tend to generalise more effectively. This is often applied in the context of logical inference by measuring the length of a declarative program that explains what is observed. The ability to identify shorter representations is compression, and the ability to generalise is arguably intelligence [3]. Hence the ability to compress information is often portrayed as a proxy for intelligence [4], even serving as the foundation [5, 6, 7] of the theoretical super-intelligence AIXI [8]. That the ability to compress information is a proxy for intelligence has gone largely unchallenged. The optimal choice of hypothesis is widely considered to be the shortest. We prove that it is not[1]. We present an alternative, unrelated to description length, called weakness. We prove that to maximise the probability of one's hypothesis generalising to unforeseen situations, it is necessary and sufficient to choose the weakest[2]. This serves to explain why The Apperception Engine [9] is able to form hypotheses that generalise.

2 Background definitions

To do so, we employ a formalism [10, 11, 12] of enactive cognition [13] in which sets of declarative programs are related to one another in such a way as to form a lattice. This unusual representation is necessary to ensure that both the weakness and description length of a hypothesis are well defined[3]. This formalism can be understood in three steps.

1. Reality is represented as a set of declarative programs.
2. A finite subset of reality is used to define a language with which to write statements that behave as logical formulae.
3. Finally, induction is formalised in terms of tasks made up of these statements.

Definition 1 (states of reality). *A set H, where:*

- *We assume a set Φ whose elements we call* ***states****, one of which we single out as the* ***present state*** *of reality*[4]*).*
- *A* ***declarative program*** *is a function $f : \Phi \rightarrow \{true, false\}$, and we write P for the set of all programs. By* ***objective truth*** *about a state ϕ, we mean a declarative program f such that $f(\phi) = true$.*
- *Given a state $\phi \in \Phi$, the* ***objective totality*** *of ϕ is the set of all objective truths $h_\phi = \{f \in P : f(\phi) = true\}$.*
- *$H = \{h_\phi : \phi \in \Phi\}$*

Definition 2 (implementable language). *A triple $\mathcal{L} = \langle H, V, L \rangle$, where:*

- *H is reality, the set containing all* ***objective totalities****.*
- *$V \subset \bigcup_{h \in H} h$ is a finite set, named the* ***vocabulary****.*

[1] This proof is conditional upon certain assumptions regarding the nature of cognition as enactive, and a formalism thereof.

[2] Assuming tasks are uniformly distributed, and weakness is well defined.

[3] An example of how one might translate propositional logic into this representation is given at the end of this paper. It is worth noting that this representation of logical formulae addresses the symbol grounding problem [14], and was specifically constructed to address subjective performance claims in the context of AIXI [15].

[4] Each state is just reality from the perspective of a point along one or more dimensions. States of reality must be separated by something, or there would be only one state of reality. For example two different states of reality may be reality from the perspective of two different points in time, or in space and so on.

- $L = \{l \in 2^V : \exists h \in H \ (l \subseteq h)\}$, *the elements of which are* ***statements***[5].

(Extensions) The **extension of a statement** $a \in L$ is $Z_a = \{b \in L : a \subseteq b\}$, while the **extension of a set of statements** $A \subseteq L$ is $Z_A = \bigcup_{a \in A} Z_a$.

(Notation) Lower case letters s, d, m, z, c represent statements, and upper case S, D, M, Z represent sets of statements. The capital letter Z with a subscript indicates the extension of whatever is in the subscript. For example the extension of a statement a is Z_a, and the extension of a set of statements A is Z_A.

Definition 3 (task). *Given a language $\langle H, V, L \rangle$, a task*[6] *is a triple $T = \langle S, D, M \rangle$ where:*

- *$S \subset L$ is a set of statements called* ***situations****, where the extension Z_S of S is the set of all* ***possible decisions*** *which can be made in those situations.*
- *$D \subset Z_S$ is the set of* ***correct decisions*** *for this task.* [7]
- *$M \subset L$ is the set of all valid* ***models*** *for the task, where*

$$M = \{m \in L : Z_S \cap Z_m \equiv D, \forall z \in Z_m \ (z \subseteq \bigcup_{d \in D} d)\}$$

(How a task is completed) *Assume we have a hypothesis $\boldsymbol{h} \in L$:*

1. *we are then presented with a situation $s \in S$, and*
2. *we must select a decision $z \in Z_s \cap Z_{\boldsymbol{h}}$.*
3. *If $z \in D$, then the decisions is correct and the task will be completed. This will occur if $\boldsymbol{h} \in M$.*

Note that $\forall m \in M : D \equiv Z_S \cap Z_m$, which means any $m \in M$ can be used to obtain D from S, because $D = \{z \in Z_m : \exists s \in S \ (s \subset z)\}$.

3 Formalising induction

Definition 4 (probability of a task). *Let Γ be the set of all tasks given an implementable language $\mathfrak{L}$. There exists a uniform distribution over Γ.*

Definition 5 (generalisation). *Given two tasks $\alpha = \langle S_\alpha, D_\alpha, M_\alpha \rangle$ and $\omega = \langle S_\omega, D_\omega, M_\omega \rangle$, a model $m \in M_\alpha$ generalises to task ω if $m \in M_\omega$.*

[5] Statements are the logical formulae about which we will reason. If the totality of the present state of reality is $h \in H$, then a statement l is **true** iff $l \subset h$. 2^V is the powerset of V.

[6] For example, this could represent chess as a supervised learning problem where $s \in S$ is the state of a chessboard, $z \in Z_s$ is a sequence of moves by two players that begins in s, and $d \in D \cap Z_s$ is such a sequence of moves that resulted in victory for one player in particular (the one undertaking the task).

[7] Note that each $d \in D$ is a superset of a member of S. S may be understood as a set of inputs, and D as the set of all unions of input and output which are correct.

Definition 6 (child-task and parent-task). *A task $\alpha = \langle S_\alpha, D_\alpha, M_\alpha \rangle$ is a child-task of $\omega = \langle S_\omega, D_\omega, M_\omega \rangle$ if $S_\alpha \subset S_\omega$ and $D_\alpha \subseteq D_\omega$. This is written as $\alpha \sqsubset \omega$. If $\alpha \sqsubset \omega$ then ω is then a parent of α, and α is a child of ω.*

A proxy is meant to estimate one thing by measuring another. In this case, if intelligence is the ability to generalise [10, 12, 3], then a greater proxy value is meant to indicate that a statement is more likely to generalise. Not all proxies are effective (most will be useless). We focus on two in particular.

Definition 7 (proxy for intelligence). *A proxy is a function $q : L \rightarrow \mathbb{N}$. The set of all proxies is Q.*

(Weakness) *The weakness of a statement m is the cardinality of its extension $|Z_m|$. There exists $q \in Q$ such that $q(m) = |Z_m|$.*

(Description Length) *The description length of a statement m is its cardinality $|m|$. Longer logical formulae are considered less likely to generalise [2], and a proxy is something to be maximised, so description length as a proxy is $q \in Q$ such that $q(m) = \frac{1}{|m|}$.*

A child task may serve as an ostensive definition of its parent, meaning one can generalise from child to parent.

Definition 8 (induction). *$\alpha = \langle S_\alpha, D_\alpha, M_\alpha \rangle$ and $\omega = \langle S_\omega, D_\omega, M_\omega \rangle$ are tasks such that $\alpha \sqsubset \omega$. Assume we are given a proxy $q \in Q$, the complete definition of α and the knowledge that $\alpha \sqsubset \omega$. We are not given the definition of ω. The process of induction would proceed as follows:*

1. *Obtain a hypothesis by computing a model $\mathbf{h} \in \underset{m \in M_\alpha}{\arg\max}\, q(m)$.*
2. *If $\mathbf{h} \in M_\omega$, then we have generalised from α to ω.*

4 Results

Proposition 1 (sufficiency). *Weakness is a proxy sufficient to maximise the probability that induction results in generalisation from α to ω.*

Proof: You're given the definition of $\alpha = \langle S_\alpha, D_\alpha, M_\alpha \rangle$ and a hypothesis $\mathbf{h} \in M_\alpha$. Let $\omega = \langle S_\omega, D_\omega, M_\omega \rangle$ be the parent to which we wish to generalise:

1. The set of statements which *might* be decisions addressing situations in S_ω and not S_α, is $\overline{Z_{S_\alpha}} = \{l \in L : l \notin Z_{S_\alpha}\}$.
2. For any given $\mathbf{h} \in M_\alpha$, the set of decisions $\mathbf{h}$ implies which fall outside the scope of what is required for the known task α is $\overline{Z_{S_\alpha}} \cap Z_{\mathbf{h}}$.
3. $|\overline{Z_{S_\alpha}} \cap Z_{\mathbf{h}}|$ increases monotonically with $|Z_{\mathbf{h}}|$, because $\forall z \in Z_m : z \notin \overline{Z_{S_\alpha}} \rightarrow z \in Z_{S_\alpha}$.
4. $2^{|\overline{Z_{S_\alpha}}|}$ is the number of tasks which fall outside of what it is necessary for a model of α to generalise to, and $2^{|\overline{Z_{S_\alpha}} \cap Z_{\mathbf{h}}|}$ is the number of those tasks to which a given $\mathbf{h} \in M_\alpha$ does generalise.

5. Therefore the probability that a given model $\mathbf{h} \in M_\alpha$ generalises to the unknown parent task ω is

$$p(\mathbf{h} \in M_\omega \mid \mathbf{h} \in M_\alpha, \alpha \sqsubset \omega) = \frac{2^{|\overline{Z_{S_\alpha}} \cap Z_{\mathbf{h}}|}}{2^{|\overline{Z_{S_\alpha}}|}}$$

$p(\mathbf{h} \in M_\omega \mid \mathbf{h} \in M_\alpha, \alpha \sqsubset \omega)$ is maximised when $|Z_{\mathbf{h}}|$ is maximised.

Proposition 2 (necessity). *To maximise the probability that induction results in generalisation from α to ω, it is necessary to weakness as a proxy, or a function thereof*[8]*.*

Proof: Let α and ω be defined exactly as they were in the proof of prop. 1.

1. If $\mathbf{h} \in M_\alpha$ and $Z_{S_\omega} \cap Z_{\mathbf{h}} = D_\omega$, then it must be he case that $D_\omega \subseteq Z_{\mathbf{h}}$.
2. If $|Z_{\mathbf{h}}| < |D_\omega|$ then generalisation cannot occur, because that would mean that $D_\omega \nsubseteq Z_{\mathbf{h}}$.
3. Therefore generalisation is only possible if $|Z_m| \geq |D_\omega|$, meaning a sufficiently weak hypothesis is necessary to generalise from child to parent.
4. The probability that $|Z_m| \geq |D_\omega|$ is maximised when $|Z_m|$ is maximised. Therefore to maximise the probability induction results in generalisation, it is necessary to select the weakest hypothesis.

To select the weaknest hypothesis, it is necessary to use weakness (or a function thereof) as a proxy.

Remark 1 (prior). The above describes inference from a child to a parent. However, it follows trivially increasing the weakness of a statement increases the probability that it will generalise to any task. As tasks are uniformly distributed, every statement in L is a model to one or more tasks, and the number of tasks to which each statement $l \in L$ generalises is $2^{|Z_l|}$. Hence the probability of generalisation[9] to $\langle S, D, M \rangle$ is $p(\mathbf{h} \in M \mid \mathbf{h} \in L) = \frac{2^{|Z_{\mathbf{h}}|}}{2^{|L|}}$. This assigns a probability to every statement $l \in L$ given an implementable language. It is a probability distribution in the sense that the probability of mutually exclusive statements sums to one[10]. This prior may be considered universal in the very limited sense that it assigns a probability to every conceivable hypothesis (where what is conceivable depends upon the implementable language) absent any parameters or specific assumptions about the task as with AIXI's intelligence order relation [8, def. 5.14 pp. 147][11]. As the vocabulary of the implementable language V is finite, L must also be finite, and so p is computable.

[8] For example we might use weakness multiplied by a constant to the same effect.

[9] $\frac{2^{|Z_{\mathbf{h}}|}}{2^{|L|}}$ is maximised when $\mathbf{h} = \emptyset$, because the optimal hypothesis given no information is to assume nothing (you've no sequence to predict, so why make assertions that might contradict reality?).

[10] Two statements a and b are mutually exclusive if $a \notin Z_b$ and $b \notin Z_a$, which we'll write as $\mu(a,b)$. Given $x \in L$, the set of all mutually exclusive statements is a set $K_x \subset L$ such that $x \in K_x$ and $\forall a, b \in K_x : \mu(a,b)$. It follows that $\forall x \in L, \sum\limits_{b \in K_x} p(b) = 1$.

[11] We acknowledge that some may object to the use of the term universal, because V is finite.

We have shown that, if tasks are uniformly distributed, then weakness is a necessary and sufficient proxy to maximise the probability that induction results in generalisation. It is important to note that another proxy may perform better given cherry-picked combinations of child and parent task for which that proxy is suitable. However, such a proxy would necessarily perform worse given the uniform distribution of all tasks (because weakness is necessary and sufficient to maximise the probability of generalisation in that case). Can the same be said of description length?

Proposition 3. *Description length is neither a necessary nor sufficient proxy for the purposes of maximising the probability that induction results in generalisation.*[12].

Proof: In propositions 1 and 2 we proved that weakness is a necessary and sufficient proxy. It follows that either maximising $\frac{1}{|m|}$ (minimising description length) maximises $|Z_m|$ (weakness), or minimisation of description length is unnecessary to maximise the probability of generalisation. Assume the former, and we'll construct a counterexample with an implementable language $\langle H, V, L \rangle$ where $L = \{\{a, b, c, d, j, k, z\}, \{e, b, c, d, k\}, \{a, f, c, d, j\}, \{e, b, g, d, j, k, z\}, \{a, f, c, h, j, k\}, \{e, f, g, h, j, k\}\}$ and a task $\langle S, D, M \rangle$ where

- $S = \{\{a, b\}, \{e, b\}\}$
- $D = \{\{a, b, c, d, j, k, z\}, \{e, b, g, d, j, k, z\}\}$
- $M = \{\{z\}, \{j, k\}\}$

Weakness as a proxy selects $\{j, k\}$, while description length as a proxy selects $\{z\}$. This demonstrates the minimising description length does not necessarily maximise weakness, and maximising weakness does not minimise description length. As weakness is necessary and sufficient to maximise the probability of generalisation, it follows that minimising description length is neither[13].

4.1 Experiments

Included with this paper is a Python script to perform experiments using PyTorch with CUDA, SymPy and A^* [16, 17, 18, 19] (see commented code and appendix for details). In these experiments, a toy program computes models to 8-bit string prediction tasks. The purpose of this experiment was to compare the performance of weakness and description length as proxies.

Implementable language: To specify tasks with which the experiments would be conducted, we needed an implementable language to describe simple 8-bit string prediction problems. Hence there were 256 states, one for every possible 8-bit string. The possible statements were then all the expressions regarding those 8 bits that could be written in propositional logic (the simple connectives $\neg$, $\wedge$

[12] In plain English, we are saying description length is a worse proxy than weakness.

[13] Hence weakness is a better proxy than description length.

and $\vee$ needed to perform binary arithmetic – a written example of how propositional logic can be used in an implementable language is also included in the appendix). To re-iterate, the implementable language used was a representation of propositional logic as it pertained to these 8 bits, meaning for each statement there exists an equivalent expression in propositional logic. For efficiency, these statements were implemented as either PyTorch tensors or SymPy expressions in different parts of the program, and converted back and forth depending on what was convenient (basic set and logical operations on these propositional tensor representations were implemented for the same reason).

Task: A task was specified by choosing $D \subset L$ such that all $d \in D$ conformed to the rules of either binary addition or multiplication with 4-bits of input, followed by 4-bits of output. The experiments were made up of trials. The parameters of each trial were "operation" (a function), and an even integer "number_of_trials" between 4 and 14 which determined the cardinality of the set D_k (defined below). Each trial was divided into training and testing phases. The training phase proceeded as follows:

1. A task T_n was generated:
   (a) First, every possible 4-bit input for the chosen binary operation was used to generate an 8-bit string. These 16 strings then formed D_n.
   (b) A bit between 0 and 7 was then chosen, and S_n created by cloning D_n and deleting the chosen bit from every string (meaning S_n was composed of 16 different 7-bit strings, each of which could be found in an 8-bit string in D_n).
2. A child-task $T_k = \langle S_k, D_k, M_k \rangle$ was sampled from the parent task T_n. Recall, $|D_k|$ was determined as a parameter of the trial.
3. From T_k two models (rulesets) were then generated; a weakest c_w, and a MDL c_{mdl}.

For each model $c \in \{c_w, c_{mdl}\}$, the testing phase was as follows:

1. The extension Z_c of c was then generated.
2. A prediction D_{recon} was then constructed s.t. $D_{recon} = \{z \in Z_c : \exists s \in S_n \ (s \subset z)\}$.
3. D_{recon} was then compared to the ground truth D_n, and results recorded.

Between 75 and 256 trials were run for each value of the parameter $|D_k|$. Fewer trials were run for larger values of $|D_k|$ as these took longer to process. The results of these trails were then averaged for each value of $|D_k|$.

Rate at which models generalised completely: Generalisation was deemed to have occurred where $D_{recon} = D_n$. The number of trials in which generalisation occurred was measured, and divided by n to obtain the rate of generalisation for c_w and c_{mdl}. Error was computed as a Wald 95% confidence interval.

Average extent to which models generalised: Even where $D_{recon} \neq D_n$, the extent to which models generalised could be ascertained. $\frac{|D_{recon} \cap D_n|}{|D_n|}$ was measured and averaged for each value of $|D_k|$, and the standard error computed.

Experimental results: These results (displayed in tables 1 and 2) demonstrate that weakness is a significantly better proxy for intelligence than compression (meaning the minimisation of description length). The generalisation rate for c_w was between $110 - 500\%$ of c_{mdl}, and the extent of generalisation between $103 - 156\%$. The difference varied with the problem type (multiplication or addition) and the value of $|D_k|$.

Table 1. Results for Binary Addition

	c_w				c_{mdl}			
$\lvert D_k \rvert$	Rate	±95%	AvgExt	StdErr	Rate	±95%	AvgExt	StdErr
6	.11	.039	.75	.008	.10	.037	.48	.012
10	.27	.064	.91	.006	.13	.048	.69	.009
14	.68	.106	.98	.005	.24	.097	.91	.006

Table 2. Results for Binary Multiplication

	c_w				c_{mdl}			
$\lvert D_k \rvert$	Rate	±95%	AvgExt	StdErr	Rate	±95%	AvgExt	StdErr
6	.05	.026	.74	.009	.01	.011	.58	.011
10	.16	.045	.86	.006	.08	.034	.78	.008
14	.46	.061	.96	.003	.21	.050	.93	.003

5 Concluding remarks

We have shown that, if tasks are uniformly distributed, then weakness maximisation is necessary and sufficient to maximise the probability that induction will produce a hypothesis that generalises. It follows that there is no choice of proxy that performs at least as well as weakness maximisation across all possible combinations of child and parent task while performing strictly better in at least one. We've also shown that the minimisation of description length is neither necessary nor sufficient. This calls into question the supposed relationship between compression and intelligence [4, 20, 21]. This is supported by our experimental results, which demonstrate that weakness is a far better predictor of whether a hypothesis will generalise, than description length. Weakness should not be conflated with simplicity or Ockham's Razor. A simple statement need not be weak, for example "all things are blue crabs". Likewise, complex nonsense can assert nothing in particular. If this result is to be understood as an epistemological razor, it is this:

Explanations should be no more specific than necessary.[14]

The Apperception Engine: The Apperception Engine [9, 22, 23] (Evans et. al. of Deepmind) is an inference engine that generates hypotheses that generalise often. To achieve this, Evans formalised Kant's philosophy to give the engine a "strong inductive bias". The engine forms hypotheses from only very general assertions, meaning logical formulae which are universally quantified. That is possible because the engine uses language specifically tailored to efficiently represent the sort of sequences to which it is applied. Our results suggest a simpler and more general explanation of why the engine's hypotheses generalise so well. The tailoring of logical formulae to represent certain sequences amounts to a choice of implementable language $\langle H, V, L \rangle$, and the use of only universally quantified logical formulae maximises the weakness of the resulting hypothesis. To apply this approach to induction from child task α to parent ω would mean we only entertain a model $m \in M_\alpha$ if $p(m \in M_\omega \mid m \in M_\alpha) = 1$. Obviously this can work well, but only for the subset of possible tasks that the vocabulary is able to describe in this way (anything else will not be able to be represented as a universally quantified rule, and so will not be represented at all [24]). This serves to illustrate how future research [25, 26] may investigate implementable languages to facilitate more efficient induction in particular categories of task.

Neural networks: How might a task be represented in the context of a function? Though we use continuous real values in base 10 to formalise neural networks, all computation still takes place in a discrete, finite and binary system. A finite composition of imperative programs may be represented as a finite number of declarative programs [27]. As such, activations within a network given an input can be represented as a finite set of declarative programs, expressing a decision. The choice of architecture specifies the vocabulary in which this is written, determining what sort of relations can be described according to the Chomsky Hierarchy [28]. The reason LLMs are so prone to fabrication and inconsistency may be because they are optimised only to minimise loss, rather than maximise weakness [10]. Future research should investigate means by which the weakness of a network can be maximised.

References

[1] E. Sober. *Ockham's Razors: A User's Manual.* Cambridge Uni. Press, 2015.

[14] We don't know which possibilities will eventuate. A less specific statement contradicts fewer possibilities. Of all hypotheses sufficient to explain what we perceive, the least specific is most likely.

[2] J. Rissanen. “Modeling By Shortest Data Description*”. In: *Autom.* 14 (1978), pp. 465–471.

[3] F. Chollet. *On the Measure of Intelligence.* 2019.

[4] G. Chaitin. “The Limits of Reason”. In: *Sci. Am.* 294.3 (2006), pp. 74–81.

[5] R. Solomonoff. “A formal theory of inductive inference. Part I”. In: *Information and Control* 7.1 (1964), pp. 1–22.

[6] R. Solomonoff. “A formal theory of inductive inference. Part II”. In: *Information and Control* 7.2 (1964), pp. 224–254.

[7] A. Kolmogorov. “On tables of random numbers”. In: *Sankhya: The Indian Journal of Statistics* A (1963), pp. 369–376.

[8] M. Hutter. *Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability.* Berlin, Heidelberg: Springer-Verlag, 2010.

[9] R. Evans. “Kant’s Cognitive Architecture”. PhD thesis. Imperial College, 2020.

[10] M. T. Bennett. “Symbol Emergence and the Solutions to Any Task”. In: *Artificial General Intelligence.* Ed. by B. Goertzel, M. Iklé, and A. Potapov. Cham: Springer, 2022, pp. 30–40.

[11] M. T. Bennett. “Enactivism & Objectively Optimal Super-Intelligence”. In: *Manuscript* (2023).

[12] M. T. Bennett. “Computable Artificial General Intelligence”. In: *Manuscript* (May 2022).

[13] D. Ward, D. Silverman, and M. Villalobos. “Introduction: The Varieties of Enactivism”. In: *Topoi* 36 (Apr. 2017).

[14] S. Harnad. “The symbol grounding problem”. In: *Physica D: Nonlinear Phenomena* 42.1 (1990), pp. 335–346.

[15] J. Leike and M. Hutter. “Bad Universal Priors and Notions of Optimality”. In: *Proceedings of The 28th Conference on Learning Theory, in Proceedings of Machine Learning Research* (2015), pp. 1244–1259.

[16] A. Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems.* USA: Curran Assoc. Inc., 2019.

[17] D. Kirk. “NVIDIA Cuda Software and Gpu Parallel Computing Architecture”. In: *Proceedings of the 6th International Symposium on Memory Management.* ISMM ’07. Canada: ACM, 2007, pp. 103–104.

[18] A. Meurer et al. “SymPy: Symbolic computing in Python”. In: *PeerJ Computer Science* 3 (Jan. 2017), e103. DOI: `10.7717/peerj-cs.103`.

[19] P. E. Hart, N. J. Nilsson, and B. Raphael. “A Formal Basis for the Heuristic Determination of Minimum Cost Paths”. In: *IEEE Transactions on Systems Science and Cybernetics* 4.2 (1968), pp. 100–107.

[20] J. Hernández-Orallo and D. L. Dowe. “Measuring universal intelligence: Towards an anytime intelligence test”. In: *Artificial Intelligence* 174.18 (2010), pp. 1508–1539.

[21] S. Legg and J. Veness. “An Approximation of the Universal Intelligence Measure”. In: *Algorithmic Probability and Friends.* 2011.

[22] R. Evans, M. Sergot, and A. Stephenson. "Formalizing Kant's Rules". In: *J Philos Logic* 49 (2020), pp. 613–680.

[23] R. Evans et al. "Making Sense of Raw Input". In: *Artificial Intelligence* 299 (2021).

[24] M. T. Bennett. "Compression, The Fermi Paradox and Artificial Super-Intelligence". In: *Artificial General Intelligence.* Ed. by B. Goertzel, M. Iklé, and A. Potapov. Cham: Springer, 2022, pp. 41–44.

[25] M. T. Bennett. "Emergent Causality & the Foundation of Consciousness". In: *Manuscript* (2023).

[26] M. T. Bennett. "How to Compute Meaning & Lovecraftian Horrors". In: *Manuscript* (2023).

[27] W. A. Howard. "The Formulae-as-Types Notion of Construction". In: *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism.* Ed. by J. Seldin and J. Hindley. Academic Press, 1980, pp. 479–490.

[28] G. Delétang et al. *Neural Networks and the Chomsky Hierarchy.* 2022.

6 Appendices

6.1 Example of an implementable language

- There exist 4 bits bit_1, bit_2, bit_3 and bit_4, to which each $h \in H$ assigns a value.
- $V = \{a, b, c, d, e, f, g, h, i, j, k, l\}$ is a subset of all logical tests which might be applied to these 4 bits:
  - $a : bit_1 = 1$
  - $b : bit_2 = 1$
  - $c : bit_3 = 1$
  - $d : bit_4 = 1$
  - $e : bit_1 = 0$
  - $f : bit_2 = 0$
  - $g : bit_3 = 0$
  - $h : bit_4 = 0$
  - $i : j \wedge k$
  - $j : bit_1 = bit_3$
  - $k : bit_2 = bit_4$
  - $l : i \vee bit_2 = 1$
- $L = \{\{a, b, c, d, i, j, k, l\}, \{e, b, c, d, k, l\}, \{a, f, c, d, j\}, \{e, f, c, d\}, \{a, b, g, d, k, l\}, \{e, b, g, d, i, j, k, l\}, \{a, f, g, d\}, \{e, f, g, d, j\}, \{a, b, c, h, j, l\}, \{a, b, g, h, l\}, \{e, b, c, h, l\}, \{a, f, c, h, i, j, k, l\}, \{e, f, c, h, k\}, \{e, b, g, h, j\}, \{a, f, g, h, k\}, \{e, f, g, h, i, j, k, l\}\}$

6.2 Example of a task ω

- $S = \{\{a, b\}, \{e, b\}, \{a, f\}, \{e, f\}\}$
- $D = \{\{a, b, c, d, i, j, k, l\}, \{e, b, g, d, i, j, k, l\}, \{a, f, c, h, i, j, k, l\}, \{e, f, g, h, i, j, k, l\}\}$
- $M = \{\{i\}, \{j, k\}, \{i, j, k\}, \{i, l\} ...\}$

6.3 Example of a child-task α of ω

- $S = \{\{a, b\}, \{e, b\}\}$
- $D = \{\{a, b, c, d, i, j, k, l\}, \{e, b, g, d, i, j, k, l\}\}$
- $M = \{\{i, j, k, l\}, \{b, d, j\}, ...\}$
  - Weakest model $\mathbf{m} = \{i, j, k, l\}$
  - Strongest model $\mathbf{e} = \{b, d, j\}$
  - $Z_{\mathbf{m}} = \{\{a, b, c, d, i, j, k, l\}, \{e, b, g, d, i, j, k, l\}, \{a, f, c, h, i, j, k, l\}, \{e, f, g, h, i, j, k, l\}\}$
  - $Z_{\mathbf{e}} = \{\{a, b, c, d, i, j, k, l\}, \{e, b, g, d, i, j, k, l\}\}$