Computable Artificial General Intelligence

Michael Timothy Bennett [1]

[1]Australian National University

October 30, 2023

Abstract

An artificial general intelligence (AGI), by one definition, is an agent that requires less information than any other to make an accurate prediction. It is arguable that the general reinforcement learning agent AIXI not only met this definition, but was the only mathematical formalism to do so. Though a significant result, AIXI was incomputable and its performance subjective. This paper proposes an alternative formalism of AGI which overcomes both problems. Formal proof of its performance is given, along with a simple implementation and experimental results that support these claims.

Hosted file

NeurIPS2022.zip available at https://authorea.com/users/684323/articles/678834-computable-artificial-general-intelligence

Computable Artificial General Intelligence

Michael Timothy Bennett[*]
School of Computing
Australian National University
michael.bennett@anu.edu.au

Abstract

An artificial general intelligence (AGI), by one definition, is an agent that requires less information than any other to make an accurate prediction. It is arguable that the general reinforcement learning agent AIXI not only met this definition, but was the only mathematical formalism to do so. Though a significant result, AIXI was incomputable and its performance subjective. This paper proposes an alternative formalism of AGI which overcomes both problems. Formal proof of its performance is given, along with a simple implementation and experimental results that support these claims.

1 Introduction

Intelligence, according to Chollet, is a measure of how little information one requires to attain a skill [1, 2]. Accordingly a more intelligent agent would need less information to make correct predictions, would predict at least as accurately as less intelligence agents given the same information and so adapt more effectively to changing circumstances. More succinctly, intelligence is the ability to make accurate generalisations [1, 2]. For the purposes of this paper an AGI is the most intelligent agent by this definition. There are compelling arguments to be made for the development of this sort of AGI. Such an agent would not only more effectively learn, adapt and even ascribe purpose to what it observes [3, 4, 5], but may yield social benefits in comparison to methods popular today. For example, only large organisations have the resources to train models that require a lot of data [6]. AGI would be more accessible. Yet until now there existed only one mathematical formalism which, arguably and under specific conditions, satisfied this definition of artificial general intelligence [7, 8]. It was named AIXI, and while it was formulated to address a different definition of intelligence, the universal prior [9, 10] AIXI employed also allowed it to make correct predictions from minimal data. Unfortunately that universal prior was incomputable, ensuring AIXI could only ever be approximated. To make matters worse, AIXI's performance was later shown to be subjective, because it could be affected by the Universal Turing Machine (UTM) on which it ran [11]. Nevertheless AIXI remains an important result. Given Deepmind co-founder Shane Legg's PhD thesis was on AIXI [8] it is arguable that it shaped the entire AI research sector. However, construction of an AGI requires an alternative theory which addresses these problems. This paper puts forward such a theory, proves it is an artificial general intelligence as defined above, and provides experimental evidence in support of these claims. Background material likely to be unfamiliar is introduced below.

1.1 Semantic Theories of Meaning

Extension. Gottlob Frege, a 19^{th} century philosopher and mathematician, tried to formalise language so that linguistic expressions could be treated as mathematical expressions [12]. Theories of reference were the result. These posit each sentence has a truth value, and each subsentential expression within a sentence contributes to that truth value. Assume it is true that "magpies can fly", but not that "pigs

[*] See acknowledgements in section 4.5. Author website - https://michaeltimothybennett.com/

can fly". The predicate “can fly" behaves as a boolean function that accepts an object and returns true if and only if that object is in the set of things which “can fly". The set of things to which an expression refers is called its *extension*. “Pigs can fly" yields an empty set, because there are no pigs of which “can fly" is true.

Intension. Reference alone failed to capture subtle distinctions between logically equivalent sentences [13]. Quine gave an example using animals with hearts (chordates) and animals with kidneys (renates) [14]. Consider the following expressions:

1. “all chordates are chordates"
2. “all chordates are renates"

In our world chordates and renates share the same extension, and so both expressions have the same truth value. Yet the first expression is an obvious tautology, while the second reveals the potentially useful fact that all animals with hearts also have kidneys. Their content, or meaning as a human might interpret it, is different. Another way to say two logically equivalent expressions have different content is to say they have different intensions. Logically equivalent sentences share an extension, but may differ in their intension. This notion underpins a formulation of tasks, which argues the meaningful difference between logically equivalent sentences lies in the extensions of their subsentential parts [3].

1.2 A Formulation of Tasks

Given a tool, a human will tend to look for problems that the tool might address [15]. Yet it is arguably better to first identify an urgent problem and only then consider what tools might be needed [16]. If intelligence is a tool, then the problem it addresses is a task [3]. The greater the intelligence [1, 2], the more complex the tasks with which it can cope. This begs the question, if it is better to begin with the problem then what, exactly, is a task? To answer this question a model of an arbitrary task [3] was formulated. It suggested intelligence is characterised by the ability to learn the purpose of a task, rather than the means by which one is completed [4]. Both purpose and means were sets of rules [3], but means to an end were very specific rules. It was argued that what differentiated extension from intension was how “weak" (nonspecific) the rules involved were [17]. The purpose of a task together with the circumstances in which it was undertaken then formed an intension, and all means by which that purpose could be fulfilled the extension [4]. By altering the weakness of those rules involved, the purpose of a task could be constructed from the means by which it was completed. Further details are given below of how the formulation of tasks related to intelligent agents. However, the original formulation lacked rigour, and details needed for implementation. This paper fleshes out those details.

Agency. An agent must have agency. Typically, intelligent agents are understood using the agent-environment paradigm [18], in which an agent maps percepts to actions. Nowhere in this paradigm was a task mentioned, so it was modified (Figure 1) to account for the existence of tasks.

In this modified paradigm (which amounts to enactive cognition [19]), a task would be completed as follows: an embodied mind would be presented with a situation, about which it would make a decision, causing the agent to affect the environment. That decision would be correct if sufficiently likely to cause the task’s completion. Such criteria were assumed, imposed upon the agent through natural selection or design. Because a decision could specify a sequence of actions, the formulation of tasks accounted for agency [4].

Rulesets. To distinguish correct from incorrect implied a set of rules. Any ruleset that specified the desired end of a task would be sufficient. A ruleset would be constructed from a set of only positive examples [4], known as an ostensive definition [20]. As Russell put it “all nominal definitions, if pushed back far enough, must lead ultimately to terms having only ostensive definitions" [21]. This paper assumes an ostensive definition is given, but one might also be constructed through repeated interaction in the manner of a reinforcement learning agent [4].

Language. Rules correspond unambiguously to the state of a machine (the embodied agent). The state of a machine is just a set of facts, each pertaining to self evident phenomena one could detect

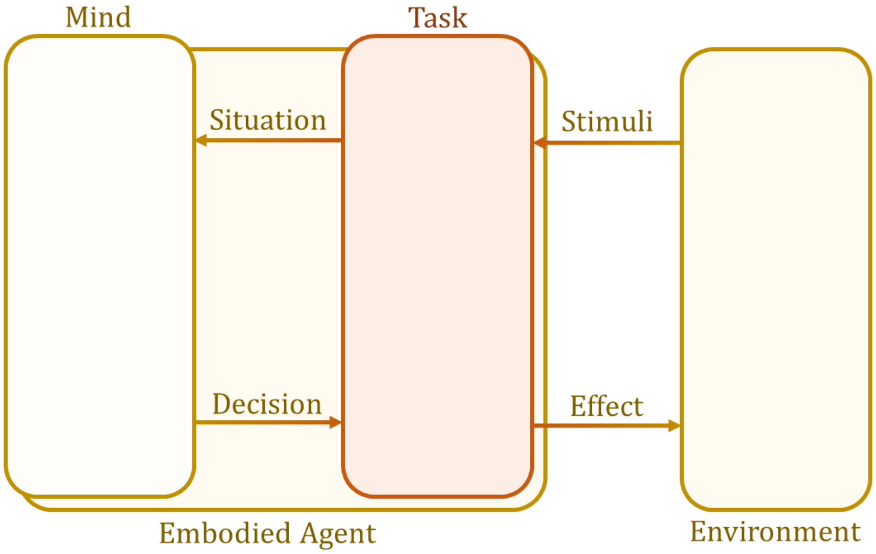

Figure 1: Agent environment paradigm, modified to account for tasks.

mechanically, such as the position of a mechanical arm or the electrical current on a wire. Not to be confused with physical symbol systems [22], preceding work [3] defined a physically implementable language as one in which statements

1. unambiguously correspond to the state of hardware (dyadic),
2. can be mechanically verified (implementable), which means
3. there is one and only one true interpretation of such a statement.

2 Definitions and Proofs

To predict the world, an agent must have a model of the world. AIXI modelled the world as a program able to perfectly reproduce what AIXI had experienced. There would always be many such programs, each of which served to completely explain the past, yet they would differ in how well they predicted the future. What made AIXI intelligent was that it had a means of discerning which of those programs would predict the future accurately. The lower a program's Kolmogorov Complexity k [23], the more likely it would correctly predict the future. This gave AIXI what is called a universal prior. Yet Kolmogorov Complexity also introduced problems. It was incomputable, and it equated plausibility with program length (true only sometimes), making AIXI's performance subjective [11]. Problems with Kolmogorov Complexity aside, the underlying ideas remained compelling. All that was needed was a different means of discerning which model (of a set of models equally capable of explaining the past) might best predict the future. This was possible using the aforementioned formulation of tasks. Under this formulation [3], models took the form of rules rather than programs. What was modelled was a task, not the environment. A universal prior would still be required, but could not use Kolmogorov Complexity. Chollet formulated a measure of intelligence, but it too relied upon Kolmogorov Complexity [2] and so could not serve (I would suggest this makes it as subjective as AIXI's performance). However, his argument that intelligence is a measure of the ability to generalise accurately remained compelling. The idea would just need to be reformulated as weakness [3], with which rulesets could be evaluated, to give a universal prior.

2.1 Definitions

To aid understanding, informal explanations accompany some of the mathematical definitions below.

Definition 0 - Fact: A measurable property of the world which is, at present, true (independent of any observer).

Informal Explanation. For example, a fact might be a specific value held at specific memory address in a computer, or the occurrence of a pattern of values at positions relative to any memory address in a computer, or even a pattern over time such as a bit changing from 1 to 0.

Definition 1 - State: The present state of a system is *the set of all facts* about that system.

Informal Explanation. Naturally, the set of all facts includes what must logically follow or have preceded the present. A set of facts is also a fact, and so there exists a fact for each and every possible representation, combination or subset all facts about a system. Further, the present state of a system is itself a fact. Importantly, a state cannot contradict itself. For example, a bit in a conventional computer cannot be 0 if it is 1 (the latter prevents the former from occurring).

Definition 2 - Hardware: A set H of possible states. There exists a probability distribution over these states, and it is assumed to be a uniform distribution.

Informal Explanation. Hardware is the body of the embodied agent. Hardware exists in one state at a time (only the present state is a fact). H is the set of all states a piece of hardware may occupy.

Definition 3 - Physically Implementable Language: A triple $\mathcal{L} = \langle H, L, \lambda \rangle$, where:

- H is hardware.
- $\lambda \subset \bigcup\limits_{h \in H} h$ is a finite set, named the **vocabulary**.
- $L = \{l \in 2^\lambda : \exists h \in H \ (l \subseteq h)\}$, the elements of which are **statements**.

Informal Explanation. There may be infinitely many states, and a state may be composed of infinitely many facts, so to avoid undecidability λ must be finite. If λ is finite, then it follows that L and every member $l \in L$ is also finite. A statement is true at a particular time if it is a subset of the state at that time. The reason each statement must be a subset of at least one possible state, is so that statements do not describe impossible combinations of facts. In other words only statements that *can* be true are permitted. Not all subsets of λ are statements, because λ may include things which cannot be true at the same time (λ is not a fact, but a set of things which can be facts if they are in the present state).

Definition 4 - Extension of a Statement: Given statement $a \in L$, the extension Z_a of a is defined as $Z_a = \{b \in L : a \subseteq b\}$.

Definition 5 - Extension of a Set of Statements: Given a set $A \subseteq L$, the extension Z_A of A is defined as $Z_A = \bigcup\limits_{a \in A} Z_a$.

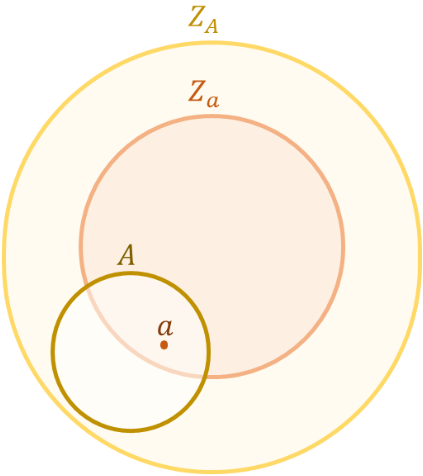

Figure 2: A diagramatic representation of definitions 4 and 5.

Conventions of Notation: To simplify presentation I'll use the following conventions for notation, except where stated otherwise:

- **Statements** are denoted by Latin lower case letters, for example x.
- **Sets of statements** are denoted by upper case, for example X.
- The **extension** of an object is denoted by the capital letter Z with the object subscripted. For example the extension of a would be denoted Z_a, and the extension of A would be denoted Z_A.

Definition 6 - Task: A task is a triple $\mathcal{T} = \langle S, G, C \rangle$ where:

- $S \subset L$ is a set of statements called **situations**, where Z_S is the set of all possible **decisions** which can be made in those situations.
- $G \subset Z_S$ is the set of **goal satisficing decisions** for this task.
- $C \subset L$ is the set of all **rulesets** for the task, distinguishing between decisions in $\mathcal{G}$ and those in $\overline{\mathcal{G}}$, where

$$C = \{c \in L : Z_S \cap Z_c \equiv G, \forall z \in Z_c \; (z \subseteq \bigcup_{g \in G} g)\}$$

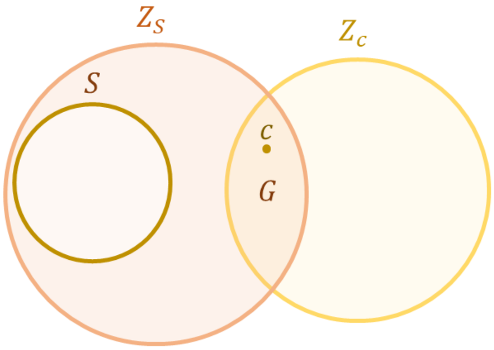

Figure 3: A diagramatic representation of definition 6.

Definition 7 - Process by Which a Decision is Made: An agent trying to complete a task is first

1. presented with a situation $s \in S$, then
2. selects $z \in Z_s$, called a decision.
3. If $z \in G$, then the agent has made a correct decision.

Note that $\forall c \in C : G \equiv Z_S \cap Z_c$. To abduct a correct decision given a situation, an agent would require a ruleset $c \in C$.

Definition 8 - Generalise: Given two tasks $\mathcal{T}_1 = \langle S_1, G_1, C_1 \rangle$ and $\mathcal{T}_2 = \langle S_2, G_2, C_2 \rangle$, a ruleset $c \in C_1$ generalises to task $\mathcal{T}_2$ if $c \in C_2$.

An equivalent alternative definition is that $c \in C_1$ generalises to $\mathcal{T}_2$ if $Z_{S_2} \cap Z_c \equiv G_2$.

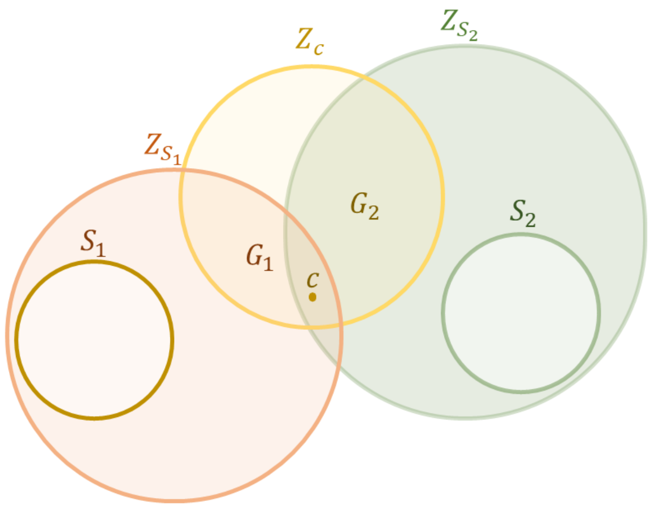

Figure 4: Diagramatic representation of definition 8.

Definition 9 - Sub-task and Parent-task: A task $\mathcal{T}_2 = \langle S_2, G_2, C_2 \rangle$ is a subtask of $\mathcal{T}_1 = \langle S_1, G_1, C_1 \rangle$ if:

1. $S_2 \subset S_1$
2. $G_2 \subset G_1$

$\mathcal{T}_1$ is then a parent task of $\mathcal{T}_2$.

Definition 10 - Weakness: $w : C \rightarrow \mathbb{N}$ is a function which accepts a ruleset and returns the cardinality of its extension:

$$w(c) = |Z_c|$$

This expressed the weakness of a given ruleset (the greater the cardinality, the weaker the ruleset).

2.2 Theoretical Results

Proposition 1: The probability that a ruleset generalises to a parent task increases monotonically with its weakness.

Proof. Let $\mathcal{T}_m = \langle S_m, G_m, C_m \rangle$ be a task of which the complete definition is known. Let $\mathcal{T}_n = \langle S_n, G_n, C_n \rangle$ be a task to which I wish to generalise. All I know of $\mathcal{T}_n$ is that it is a parent task of $\mathcal{T}_m$, meaning $G_m \subset G_n$ and $S_m \subset S_n$.

1. The set of all decisions which may potentially be required to address the situations in S_n, and which are not required for S_m, is $\overline{Z_{S_m}} = A$.
2. For any given $c_m \in C_m$, the set of decisions c_m implies which fall outside the scope of what is required for the known task $\mathcal{T}_m$ is $\overline{Z_{S_m}} \cap Z_{c_m} = B$.
3. $2^{|A|}$ is the number of tasks which fall outside of what it is necessary for a ruleset of $\mathcal{T}_m$ to generalise to, and $2^{|B|}$ is the number of those tasks to which a given c_m does generalise.
4. Therefore the probability that a given ruleset $c_m \in C_m$ generalises to the unknown parent task $\mathcal{T}_n$ is
$$p(c_m \in C_n) = \frac{2^{|B|}}{2^{|A|}}$$

$p(c_m \in C_n)$ increases monotonically with the weakness of c_m. □

Proposition 2: Weakness is necessary for generalisation.

Proof. Once again let $\mathcal{T}_m$ be a task of which the complete definition is known, and $\mathcal{T}_n$ be a parent task of $\mathcal{T}_m$ to which I wish to generalise, for which the complete definition is not known. If $Z_{S_n} \cap Z_{c_m} \equiv G_n$ then it must be he case that $G_n \subseteq Z_{c_m}$. The weaker a ruleset is, the more likely it is that $G_n \subseteq Z_{c_m}$. Therefore a sufficiently weak ruleset is necessary for generalisation. □

A universal prior is one that assigns belief to every possible hypothesis, or in this case ruleset. Weakness is a means of accurately predicting $p(c)$, how likely one ruleset c is to generalise relative to others.

Corollary - A Computable Universal Prior: $p(c) = \frac{2^{w(c)}}{2^{|L|}}$

Proposition 3: Agent which chooses the weakest rulesets is an artificial general intelligence.

Proof. Intelligence is a measure of the ability to generalise, and an artificial general intelligence is the most intelligent agent by this measure. Propositions 1 and 2 show that a preference for weaker rulesets is both necessary and sufficient to maximise the probability of generalisation. Therefore the most intelligent agent is one that constructs the weakest rulesets. □

3 Experiments

Included with this paper is a simple program [24]. It computes rulesets to solve 8-bit string prediction problems. With this, experiments were performed (see appendix B).

3.1 Methods

The purpose of these experiments was to try and falsify proposition 3. Minimum description length (MDL) rulesets were compared with the weakest rulesets. MDL was chosen because it is arguably the only plausible alternative to weakness [25, 26, 27, 28, 29]. As this was a comparison between rulesets, a MDL ruleset was defined as $c_{mdl} \in \text{argmin}_{c \in C} |c|$. The hardware H contained 256 states, one for every possible 8-bit string. Propositional logic was employed for the physically implementable language (to clarify, a written example of this is also provided in the appendix), meaning L was a set of 256 different statements in propositional logic. A task was specified by choosing $G \subset L$ such that all $g \in G$ conformed to the rules of either binary addition or multiplication with 4-bits of input, followed by 4-bits of output (the included code may be re-run with any alternative operation the reader wishes). 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 G_k (defined below). Each trial was divided into training and testing phases. The training phase proceeded as follows:

1. A task $\mathcal{T}_n$ 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 G_n.
   (b) A bit between 0 and 7 was then chosen. S_n was then created by cloning G_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 G_n).
2. A subtask $\mathcal{T}_k = \langle S_k, G_k, C_k \rangle$ was sampled from the parent task $\mathcal{T}_n$. Recall, $|G_k|$ was determined as a parameter of the trial.
3. From $\mathcal{T}_k$ two rulesets were then generated; a weakest c_w, and a MDL c_{mdl}.

For each ruleset c, the testing phase was as follows:

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

Between 75 and 256 trials were run for each value of the parameter $|G_k|$. Fewer trials were run for larger values of $|G_k|$ as these took longer to process. The largest and smallest values of $|G_k|$ were discarded due to time and resource constraints. The results of these trails were then averaged for each value of $|G_k|$.

Table 1: Results for Binary Addition

	c_w				c_{mdl}			
$\|G_k\|$	Rate	$\pm 95\%$	AvgExt	StdErr	Rate	$\pm 95\%$	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}			
$\|G_k\|$	Rate	$\pm 95\%$	AvgExt	StdErr	Rate	$\pm 95\%$	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

Rate at which rulesets generalised completely. Generalisation was deemed to have occurred where $G_{recon} = G_n$. The number of trials in which generalisation occurred was measured, and divided by n to obtain the rate of generalisation for each ruleset c_w and c_{mdl}. Error was computed as a Wald 95% confidence interval. Intuitively, generalisation meant the task was correctly understood (rate being the portion of trials in which the ruleset in question perfectly matched the ground truth).

Average Extent to which rulesets generalised. Where $G_{recon} \neq G_n$, the extent to which rulesets generalised could still be ascertained. $\frac{|G_{recon} \cap G_n|}{|G_n|}$ was measured for each ruleset, averaged for each value of $|G_k|$, and the standard error computed.

3.2 Results

The results (displayed in Table 1, Table 2 and Figure 5) support proposition 3, in that there was not a single trial in which c_{mdl} outperformed c_w.

The generalisation rate for c_w exceeded that of c_{mdl}. Reviewing the raw output data, there was one instance of c_{mdl} generalising where c_w did not. The extent to which rulesets generalised was also greater for weaker rulesets. c_w always generated as many or more true positives than c_{mdl}. The performance difference between c_{mdl} and c_w was smaller than it was for generalisation rate.

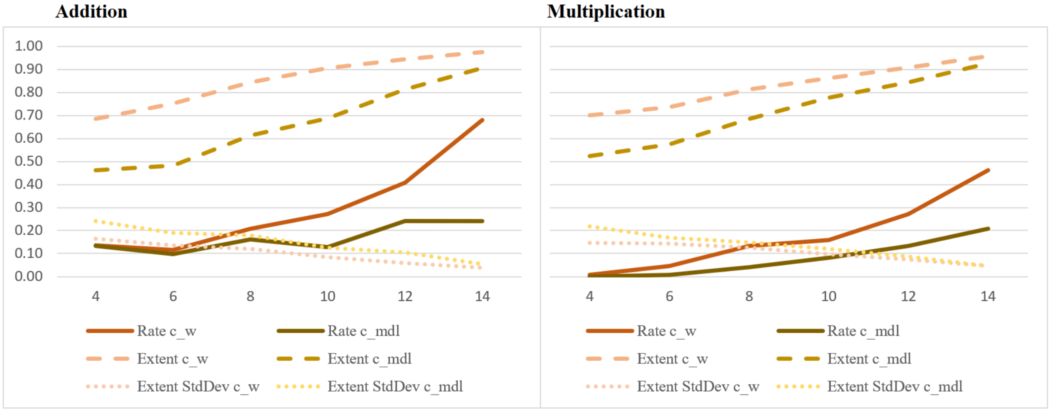

Figure 5: Plot of the experimental results. The horizontal axis is $|G_k|$.

4 Discussion

4.1 On Subjectivity and The Measure of Intelligence

AIXI's performance is subjective because the universal prior it employs equates plausibility with Kolmogorov Complexity, the truth of which depends upon the Universal Turing Machine (UTM) on which AIXI runs [11]. The computable universal prior presented here involves neither Kolmogorov Complexity nor UTMs, and so does not suffer this flaw. Chollet's formalism to measure intelligence also depends upon Kolmogorov Complexity, and so I suggest it is as subjective as AIXI's performance. Weakness, however, may facilitate an objective measure of intelligence.

4.2 Limitations

There is one important limitation to take note of. The vocabulary λ affects how weak rulesets can be. For example, one might write that "address 109 contains 1 and address 110 contains 0", or that "there exists an address i containing 1 such that address $i + 1$ contains 0, and $i = 109$". These two sentences are equivalent, but the extensions of their subsentential expressions are different. Remove "$i = 109$" from the 2nd sentence, and what remains could be interpreted as a convolution over all i. In other words, these two sentences have different intensions but the same extension. This suggests that what separates such intensions from one another is the extensions of their subsentential expressions. Weaker subsentential expressions may be recombined to form weaker sentences than stronger subsentential expressions.

Given any $\beta \in \lambda$, let $H_\beta = \{h \in H : \beta \in h\}$. This is like the extension of β, but in terms of hardware states rather than statements in L. The members of λ are like subsentential expressions. If $H_\alpha \subset H_\beta$, and if $\{\beta\}$ and $\{\alpha\}$ are both rulesets, then it must be the case that $w(\{\beta\}) \geq w(\{\alpha\})$. Therefore λ which contains β but not α will permit construction of weaker rulesets than λ containing α but not β. All else being equal, it follows that construction of the weakest rulesets possible would always be possible if the following rule, **strict objectivity**, were applied to λ:

$$\forall \beta \in \lambda \, \neg(\exists \gamma \in \overline{\lambda} \, (H_\beta \subset H_\gamma)).$$

The ruleset most likely to generalise is the weakest, but the ability to construct weaker rulesets still depends upon λ. An agent which always chooses the weakest rulesets meets the definition of AGI to the extent possible given λ. However in the context of every possible λ, strict objectivity would ensure no more intelligent agent could exist.

4.3 Connectionist Interpretations

This is not necessarily an endorsement of either symbolic or connectionist methods. Both work with this theory. I used symbolic methods here because they're easier to interpret, but the theory is as easy to apply in the context of neural networks (and can be explained in kind with category theory). For example, the rules determining correctness could be modelled by training a classifier $o(l) \in [0, 1]$ such that $l \in Z_S$ and $o(l) = 1$ if $l \in G$ and $o(l) = 0$ otherwise. o must behave in accordance with the definition of ruleset, the weakness of which could then be maximised. o could be used as a loss function to train another network $n : S \to G$ which outputs correct decisions. o learns the desired end of the task, while n learns means by which it can be completed. I will publish a paper and implementation of exactly this soon.

4.4 Potential Negative Consequences

The potential negative consequences of AGI are the subject of both popular fiction and extensive research [4]. A scaleable implementation of this theory is not given. Nevertheless this research may precipitate the automation of jobs, the development of autonomous weapons and greater instability in financial markets.

4.5 Concluding Remarks

The proofs in 2.2 show that the optimal choice of ruleset for generalisation is the weakest. The experimental results in 3.2 support this claim, as there was not a single trial in which c_{mdl} outperformed c_w.

Weakness is a computable, objective measure of how likely a ruleset is to generalise. Subsequently an agent that prefers the weakest rulesets is an artificial general intelligence (by the definition discussed in this paper's introduction [1, 2]). According to preceding work, the weakest ruleset is the purpose of a task [3]. Because it is the optimal choice for generalisation, it follows that general intelligence is characterised by the ability to learn the ends, not just the means. However, general intelligence alone is insufficient to emulate human intelligence, as human intelligence employs various inductive biases [15, 27]. The work of Evans et. al. [27, 28, 29] of Deepmind illustrates how a curated vocabulary (in which to construct rules) can provide such an inductive bias (which would make weaker rulesets easier to find in some contexts). Finally, the formulation of tasks employed here has been used to explain how human language functions [3] and lends itself to an artificial theory of mind [4, 5]. The more rigorous mathematical treatment given here can also been applied to those derivatives in future work.

Acknowledgements

The following people took the trouble to either unofficially peer review, or at least give useful feedback on this paper: Vincent Abbot, Matthew Aitchison, David Quarel and Yoshihiro Maruyama all of the ANU, and Badri Vellambi of UC. Their contributions significantly improved the clarity and rigour of this work, for which I am sincerely grateful. In particular, Vincent's suggestions on simplifying the formulation, and Matthew's suggestions regarding the communication of experimental evidence, were very much appreciated.

References

[1] Chollet, F., Mitchell, M., Szegedy, C.: Abstraction & Reasoning in AI systems: Modern Perspectives. Tutorial. Thirty-fourth Conference on Neural Information Processing Systems. https://nips.cc/Conferences/2020/Schedule?showEvent=16644 (2020)

[2] Chollet, F.: On the Measure of Intelligence. arXiv: 1911.01547[cs.AI] (2019)

[3] Bennett, M. T.: Symbol Emergence and The Solutions to Any Task. 14th Conference on Artificial General Intelligence (2021)

[4] Bennett, M. T., Maruyama, Y.: Philosophical Specification of Empathetic Ethical Artificial Intelligence. In: IEEE Transactions on Cognitive and Developmental Systems (2021)

[5] Williams, J., Fiore, S., Jentsch, F.: Supporting Artificial Social Intelligence With Theory of Mind. Frontiers in Artificial Intelligence (2022)

[6] Wang, Y., Wiebe, V.J.: Big Data Analytics on the characteristic equilibrium of collective opinions in social networks. In Big Data: Concepts, Methodologies, Tools, and Applications. IGI Global. pp. 1403-1420 (2016)

[7] Hutter, M.: Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability. Springer. (2005)

[8] Legg, S.: Machine Super Intelligence. PhD Thesis Manuscript. University of Lugano. http://www.vetta.org/documents/Machine_Super_Intelligence.pdf (2008)

[9] Solomonoff, R. J.: A formal theory of inductive inference. Part I. In: Information and Control 7(1), pp. 1-22 (1964)

[10] Solomonoff, R. J.: A formal theory of inductive inference. Part II. In: Information and Control 7(2), pp. 224-254 (1964)

[11] Leike, J., Hutter, M.: Bad Universal Priors and Notions of Optimality. Proceedings of The 28th Conference on Learning Theory, in Proceedings of Machine Learning Research 40. pp. 1244-1259 (2015)

[12] Frege, G.: Kurze Übersicht meiner logischen Lehren?. Unpublished. (1906) [Translation in Beaney, M.: The Frege Reader. Oxford Blackwell. pp. 299-300 (1997)]

[13] Speaks, J.: Theories of Meaning. In: The Stanford Encyclopedia of Philosophy. Ed. by Edward N. Zalta. Spring 2021. Metaphysics Research Lab, Stanford University. (2021)

[14] Quine, W.V.O.: Philosophy of Logic, 2nd Edition. Prentice Hall. pp. 8-9 (1986)

[15] Maslow, A.: The Psychology of Science. Harper & Row. pp. 15 (1966)

[16] Spradlin, D.: Are You Solving the Right Problem?. Harvard Business Review. September Issue (2012)

[17] Bennett, M. T., Maruyama, Y.: Intensional Artificial Intelligence: From Symbol Emergence to Explainable and Empathetic AI. arXiv:2104.11573 [cs.AI]. (2021)

[18] Russell, S., P. Norvig.: Artificial intelligence: A modern approach, global edition 4th. Pearson. pp. 36 (2021)

[19] Thompson, E.: Mind in Life. In: Biology, Phenomenology and the Sciences of Mind 18 (2007)

[20] Gupta, A: Definitions. The Stanford Encyclopedia of Philosophy (Winter Edition), Edward N. Zalta (ed.) (2021)

[21] Russell, B.: Human Knowledge: Its Scope and Limits. New York. Simon and Schuster. pp. 242 (1948)

[22] Nilsson, N. J.: The Physical Symbol System Hypothesis: Status and Prospects. In 50 Years of Artificial Intelligence. Springer. pp. 9-17 (2007)

[23] Kolmogorov, A. N.: On tables of random numbers. In: Sankhya: The Indian Journal of Statistics A. pp. 369-376 (1963)

[24] Bennett, M. T.: Computable Artificial General Intelligence Supplementary Materials, https://www.dropbox.com/s/wmvxepatro88ma6/NeurIPS2022.zip?dl=0 (2022)

[25] Baker, A.: Simplicity. Stanford Encyclopedia of Philosophy (2016)

[26] Budhathoki, K., Vreeken, J.: Origo: Causal Inference by Compression. In: Knowledge and Information Systems 56(2). pp. 28-307 (2018)

[27] Evans, R.: Kant's Cognitive Architecture. PhD Thesis, Imperial College (2020)

[28] Evans, R., Sergot, M., Stephenson, A.: Formalizing Kant's Rules. J Philos Logic 49. 613-680 (2020)

[29] Evans, R., Bošnjak, M., Buesing, L., Ellis, K., Pfau, D., Kohli, P., Sergot, M.: Making Sense of Raw Input. In: Artificial Intelligence 299 (2021)

A Physically Implementable Language

Example of a physically implementable language:

- Assume there are 4 bits bit_1, bit_2, bit_3 and bit_4, and that there exists a unique member of H corresponding to each possible assignment of values to those 4 bits. There are subsequently 16 states in H, but the length of each state is infinite because each contains every possible means of describing the subsequent 4-bit string.
- $\lambda = \{a, b, c, d, e, f, g, h, i, j, k, l\}$ is a subset of all logical tests which might be applied to these 4 bits:
  - $a : bit_1 = 1$
  - $b : bit_2 = 1$
  - $c : bit_3 = 1$
  - $d : bit_4 = 1$
  - $e : bit_1 = 0$
  - $f : bit_2 = 0$
  - $g : bit_3 = 0$
  - $h : bit_4 = 0$
  - $i : j \wedge k$
  - $j : bit_1 = bit_3$
  - $k : bit_2 = bit_4$
  - $l : i \vee bit_2 = 1$
- $L = \{\{a, b, c, d, i, j, k, l\}, \{e, b, c, d, k, l\}, \{a, f, c, d, j\}, \{e, f, c, d\}, \{a, b, g, d, k, l\}, \{e, b, g, d, i, j, k, l\}, \{a, f, g, d\}, \{e, f, g, d, j\}, \{a, b, c, h, j, l\}, \{e, b, c, h, l\}, \{a, f, c, h, i, j, k, l\}, \{e, f, c, h, k\}, \{a, b, g, h, l\}, \{e, b, g, h, j\}, \{a, f, g, h, k\}, \{e, f, g, h, i, j, k, l\}\}$

Example of a task $\mathcal{T}_1$

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

Example of subtask $\mathcal{T}_2$ sufficient for generalisation to $\mathcal{T}_1$

- $S = \{\{a, b\}, \{e, b\}\}$
- $G = \{\{a, b, c, d, i, j, k, l\}, \{e, b, g, d, i, j, k, l\}\}$
- $C = \{\{i, j, k, l\}, \{b, d, j\}, ...\}$
  - Weakest (intensional) ruleset $\mathbf{i} = \{i, j, k, l\}$
  - Strongest (extensional) ruleset $\mathbf{e} = \{b, d, j\}$
  - $Z_{\mathbf{i}} = \{\{a, b, c, d, i, j, k, l\}, \{e, b, g, d, i, j, k, l\}, \{a, f, c, h, i, j, k, l\}, \{e, f, g, h, i, j, k, l\}\}$
  - $Z_{\mathbf{e}} = \{\{a, b, c, d, i, j, k, l\}, \{e, b, g, d, i, j, k, l\}\}$

B Experiments

The experiments were run on a Dell Precision laptop with 32GB of RAM, an RTX A5000 GPU with 16GB of VRAM and an i7 CPU. CUDA was used, but it's optional (the experiments will run on a CPU).

The exact results are available in “genrates.xlsx". To rerun the experiments, execute “experiments.py" included with this paper. Results will be printed to terminal and displayed in a PyPlot.

Parameters can be altered at the end of the file. “number_of_trials" changes the number of trials attempted for each value of $|G_k|$, and “operation" is the function modelled (binary multiplication by default). It can be changed as desired, but will throw divide by zero errors for some operators.

There is also “depth_limit" and “time_limit" which were not used in the experiments, but which will significantly speed up the running of the experiments for any who want to run them. They shouldn’t alter the results too much unless set to very low extremes. They are thresholds beyond which searches for rulesets will be aborted.

A large number of comments have been added to the code to make it more understandable.

The code was written with interpretability and expediency in mind. It is not intended to be fast. However I created a means of doing propositional logic with tensors and PyTorch to save time. There are potential further uses for this, for example to speed up SAT solvers.

C Supplemental Definitions and Preceding Work

The formulation of tasks upon which this was based used different terms. Rulesets were named solutions, decisions were named responses and so on. These terms were changed so that the theory would be more understandable. Though not defined above, intensional and extensional solutions, as they are called in preceding work, may be useful for future work if defined as part of the formulation given in this paper. Supplementary definitions are provided below:

Supplementary Definition 1 - Intensional Ruleset:

$$\mathbf{i} \in \operatorname*{argmax}_{c \in C} w(c)$$

Supplementary Definition 2 - Extensional Ruleset:

$$\mathbf{e} \in \operatorname*{argmin}_{c \in C} w(c)$$

Intensional and extensional rulesets provide bounds on weakness such that

$$\forall c \in C : w(\mathbf{e}) \leq w(c) \leq w(\mathbf{i})$$

$$\mathbf{e} \equiv Z_{\mathbf{e}} = G \subseteq Z_{\mathbf{i}} \equiv \mathbf{i}$$

Supplementary Definition 3 - Sufficient Subtask: A subtask $\mathcal{T}_2$ of $\mathcal{T}_1$ is sufficient if for every intensional ruleset $\mathbf{i}_2 \in C_2$ there exists an intensional ruleset $\mathbf{i}_1 \in C_1$ such that $\mathbf{i}_2 = \mathbf{i}_1$. In other words:

$$\operatorname*{argmax}_{c \in C_2} w(c) \subseteq \operatorname*{argmax}_{c \in C_1} w(c)$$

In other words every intensional ruleset to $\mathcal{T}_2$ generalises to $\mathcal{T}_1$. I call $\mathcal{T}_2$ sufficient because it provides sufficient information to be absolutely certain of the rules of the parent task. $\mathcal{T}_2$ is also called a sufficient ostensive definition of $\mathcal{T}_1$, as per previous publications on the topic.