Short-circuiting the definition of mathematical knowledge for an Artificial General Intelligence

Samuel Allen Alexander[1][0000−0002−7930−110X]

The U.S. Securities and Exchange Commission samuelallenalexander@gmail.com
https://philpeople.org/profiles/samuel-alexander/publications

Abstract. We propose that, for the purpose of studying theoretical properties of the knowledge of an agent with Artificial General Intelligence (that is, the knowledge of an AGI), a pragmatic way to define such an agent's knowledge (restricted to the language of Epistemic Arithmetic, or EA) is as follows. We declare an AGI to know an EA-statement ϕ if and only if that AGI would include ϕ in the resulting enumeration if that AGI were commanded: "Enumerate all the EA-sentences which you know." This definition is non-circular because an AGI, being capable of practical English communication, is capable of understanding the everyday English word "know" independently of how any philosopher formally defines knowledge; we elaborate further on the non-circularity of this circular-looking definition. This elegantly solves the problem that different AGIs may have different internal knowledge definitions and yet we want to study knowledge of AGIs in general, without having to study different AGIs separately just because they have separate internal knowledge definitions. Finally, we suggest how this definition of AGI knowledge can be used as a bridge which could allow the AGI research community to import certain abstract results about mechanical knowing agents from mathematical logic.

Keywords: AGI · machine knowledge · quantified modal logic

1 Introduction

It is difficult to define knowledge, or what it means to know something. In Plato's dialogues, again and again Socrates asks people to define knowledge[1], and no-one ever succeeds. Neither have philosophers reached consensus even in our own era [15].

At the same time, the problem is often brushed aside as something only philosophers care about: pragmatists rarely spend time on this sort of debate. One exception is in the study of agents with Artificial General Intelligence (*AGIs*, or *Type II AIs* in the terminology of [7]), where even the staunchest pragmatists admit the importance of the question.

In this paper, we narrow down the question "what is knowledge" and offer a simple answer within that narrow context: we propose a definition of what

[1] Perhaps the best example being in the *Theaetetus* [18].

it means for a suitably idealized AGI to know a mathematical sentence[2] in the language of Epistemic Arithmetic [21] (hereafter EA). EA is the language of Peano Arithmetic along with an additional modal operator K for knowledge. To be precise:

- EA-terms are built up from variables $x, y, \ldots$ and a constant symbol for 0, the unary function symbol S for the successor function, and binary function symbols for addition and multiplication.
- EA-formulas are built up from atomic EA-formulas (which are of the form $s = t$ where s and t are EA-terms), propositional connectives $\rightarrow, \neg$, universal quantifiers $\forall x, \forall y, \ldots$, existential quantifiers $\exists x, \exists y, \ldots$, and modal operator K.

The EA-sentence $K(1 + 1 = 2)$ might be read "I know $1 + 1 = 2$" or "the knower knows $1 + 1 = 2$". Our proposed definition is parsimonious (at the price of appearing deceptively circular). We say that an AGI knows an EA-sentence ϕ if and only if ϕ would be among the sentences which that AGI would enumerate if that AGI were commanded:

"Enumerate all the EA-sentences which you know."

This is non-circular because an AGI, being capable of practical English communication, is therefore capable of understanding the everyday English word "know" in the above command, independently of how any philosopher formally defines knowledge. We discuss this further in Subsection 3.1.

A primary motivation for this paper was the author's experience in the AGI research community where applications of mathematical logic are hindered by questions like "What does it mean for an AGI to know something?" For example, philosophers have long known that a suitably idealized mechanical knowing agent cannot know its own code and its own truthfulness[3]. But in informal conversations, we find AGI researchers struggle with this assertion, and we can hardly blame them, since, without agreeing what it means for the AGI to know something, of course the question arises, "What does it mean for an AGI to know something?" Likewise, we have proposed [6] an AGI intelligence measure based on the AGI's knowledge, and this, too, often provokes the response "What does it mean for an AGI to know something?" In Section 5 we will consider these examples, and related examples from the same area, using our proposed definition to translate them into a more concrete form, not in terms of what the AGI knows, but in terms of the AGI's stimulus-responses.

The structure of this paper is as follows.

- In Section 2 we discuss the AGIs whose knowledge we are attempting to define.

[2] By a *sentence*, we mean a formula with no free variables. Thus, $x^2 > 0$ is not a sentence, but $\forall x(x^2 > 0)$ is.

[3] Often phrased more like "cannot know its own code", with knowledge-of-own-truthfulness taken for granted.

- In Section 3 we propose a knowledge definition for AGIs for EA-sentences.
- In Section 4 we extend our knowledge definition to formulas with free variables.
- In Section 5 we use this knowledge definition as a bridge to translate some ideas from mathematical logic into the field of AGI.
- In Section 6 we summarize and make concluding remarks.

2 Idealized AGIs

In this paper, we approach AGI using what Goertzel [13] calls the Universalist Approach: we adopt "...an idealized case of AGI, similar to assumptions like the frictionless plane in physics", hoping that by understanding this "simplified special case, we can use the understanding we've gained to address more realistic cases." At the same time, an AGI might serve as a kind of hyper-idealized proxy for human cognition, and we hope that the development of the logic of AGI may serve as a step toward development of "new forms of logic as the basis of cognitive and substrate-independent studies of intelligent interaction" [11].

We do not have a formal definition for what an AGI is, but whatever it is, we assume an AGI is a deterministic machine which repeatedly reads sensory input from its environment and outputs English words based on what sensory inputs it has received so far[4]. When we say that this AGI is a "deterministic machine", we mean that said outputs (considered as a function of said inputs) could be computed by a Turing machine. We further assume the AGI can understand English commands and is capable of practical English communication. Thus, if we were to command the AGI in English, "Tell us the value of $1 + 1$", the AGI would respond in English and reply "2", or "$1 + 1 = 2$", or something along those lines[5].

We assume an AGI is capable of everyday English discussions which would cause no difficulty to a casual English speaker, even if these discussions involve topics, such as "knowledge", which might be philosophically tricky. A casual English speaker does not get stuck in philosophical questions about the nature of knowledge just in order to answer a question like "Do you know that $1+1 = 2$?", and therefore neither should our AGI.

We also assume an AGI is better than a casual human English speaker in certain ways. We assume an AGI would have no objections to performing tedious tasks indefinitely, if so commanded. If we asked a casual human English speaker to begin computing and reciting all the prime numbers until further notice,

[4] We should note that, with the AGI research field being so young, there is little consensus even on basic things. Some researchers would consider some things to be AGI which have no communication ability (applying the term to entities who have certain adaptation abilities or pattern-matching abilities, for example, even if those entities have no means of communicating), however, we believe that to be a minority opinion.

[5] We assume the AGI explicitly follows commands (that it is "under explicit control", to use Yampolskiy's terminology [24]).

and then we waited silently forever listening to the results, said human would eventually get tired of the endless tedium and would disobey our command (and would probably make arithmetic errors along the way). We assume an AGI has no such limitations and would happily compute and recite prime numbers for all eternity, if so commanded (without arithmetic mistakes). Of course, in reality the AGI would eventually run out of memory, terminate when the world ends, etc., but we are speaking of idealized AGI here and we intentionally ignore such possibilities, in the same way a Turing machine is assumed to have infinite tape and infinite time to run.

3 An elegant definition of mathematical knowledge

The following definition may initially look circular, but we will argue it is not.

Definition 1. *Let X be an AGI. For any EA-sentence ϕ, we say that X knows ϕ if and only if X would eventually include ϕ in the resulting enumeration if X were commanded:*

"Enumerate all the EA-sentences which you know."

Definition 1 is non-circular because the AGI is capable (see Section 2) of practical English communication, including that involving everyday English words such as the word "know", independently of how any philosophers formally define things[6]. More on this in Subsection 3.1.

One of the strengths of Definition 1 is that it is uniform across different AGIs: many different AGIs might internally operate based on different definitions of knowledge, but Definition 1 works equally well for all these different AGIs regardless of those different internal knowledge definitions[7].

Remark 1. In Definition 1 when we speak of what the AGI would do if given such a command, implicitly we intend this to be understood as what the AGI would do if given such a command and then allowed to respond to the command in isolation, without outside distractions. An AGI could potentially update its knowledge based on observations of the world, and so its knowledge might change from one time to the next: its knowledge at a given instant is defined by Definition 1 to consist of what the AGI would enumerate if the AGI were so commanded at that particular instant (and immediately secluded from further distracting observations).

Although Definition 1 may differ significantly from a particular AGI X's own internal definition of knowledge, the following theorem states that materially the two definitions have the same result.

[6] This is reminiscent of Williamson's contextualism [23].

[7] This is reminiscent of Elton's proposal that instead of trying to interpret an AI's outputs by focusing on specific low-level details of a neural network, we should instead let the AI explain itself [12].

Theorem 1. *Suppose X is an AGI. For any EA-sentence ϕ, the following are equivalent:*

1. *X is considered to know ϕ (based on Definition 1).*
2. *X knows ϕ (based on X's own internal understanding of knowledge).*

Proof. By Definition 1, (1) is equivalent to the statement that X would include ϕ in the list which X would output if X were commanded:

"Enumerate all the EA-sentences which you know."

Since we have assumed (in Section 2) that X is obedient, X would output ϕ in the resulting list if and only if (2). □

Theorem 2. *Let X be an AGI. The set of EA-sentences ϕ such that X knows ϕ (based on Definition 1) is computably enumerable.*

Proof. This follows from our assumption (in Section 2) that X is a deterministic machine. □

3.1 Non-Circularity of Definition 1

> 'What is said by a speaker (what she meant to say, her "meaning-intention") is understood or misunderstood by a hearer ("an interpreter").'
> —Albrecht Wellmer [22]

Definition 1 is non-circular because an AGI's response to an English command only depends on how the AGI understands the words in that command, not on how *we* (the speakers) understand those words. Recall from Section 2 that we are assuming an AGI is a deterministic machine which outputs English words based on sensory inputs from its environment. Those outputs depend *only* on those environmental inputs, and not on any decisions made by philosophers.

If the reader wants to further convince themselves of the non-circularity of Definition 1, we need only point out that the apparent circularity would disappear if we changed Definition 1 to define what it means for X to "grok" sentence ϕ, rather than to "know" sentence ϕ (without changing the command itself). In other words, we could define that X "groks" ϕ if and only if X would include ϕ in the list of sentences that would result if X were commanded,

"Enumerate all the EA-sentences which you know."

This would make the non-circularity clearer, because the word "grok" does not appear anywhere in the command.

We will further illustrate the non-circularity of Definition 1 with two examples.

- (The color blurple) Bob could (without Alice's awareness) define "blurple" to be the color of the card which Alice would choose if Bob were to run up to Alice, present her a red card and a blue card, and demand: "Quick, choose the blurple card! Do it now, no time for questions!" There is nothing circular about this, because Alice's choice cannot depend on a definition which Alice is unaware of.
- (Zero to the zero) If asked to compute 0^0, some calculators output 1, and some output an error message or say the result is undefined[8]. For any calculator X, it would be perfectly non-circular to define "the 0^0 of X" to be the output which X outputs when asked to compute 0^0. Said output is pre-programmed into the calculator; the calculator does not read the user's mind in order to base its answer on any definitions that exist there.

These considerations hinge on the AGI being separate from the reader. The human reader can apply Definition 1 to AGIs which she creates, but not to herself. An AGI X could apply Definition 1 to child AGIs that X created, but X could not apply the definition to X's own knowledge[9].

3.2 Sentences using the Knowledge Operator

Definition 1 is particularly interesting when ϕ itself makes use of EA's K operator for knowledge.

Example 1. Applying Definition 1, we consider an AGI X to know $K(1+1=2)$ if and only if X would output $K(1+1=2)$ when commanded to enumerate all the EA-sentences he knows. X would (when so commanded) output $K(1+1=2)$ if and only if X knows (in his own internal sense of the word "know") that he knows (in his own internal sense of the word "know") $1+1=2$.

3.3 A Simpler Definition, and Why It Does Not Work

> "It is difficult to be aware of whether one knows or not. For it is difficult to be aware of whether we know from the principles of a thing or not—and that is what knowing is. (...) Let that demonstration be better which, other things being equal, depends on fewer postulates or suppositions. For if they are equally familiar, knowing will come about more quickly in this way; and that is preferable." —Aristotle [8]

[8] Which is incorrect—see [16].

[9] This is reminiscent of a recent argument [17] that humans maintain superiority over the AIs they create, as, for example, today's latest and greatest chess-playing AI is better at tactically playing individual games of chess, but is incapable of designing its own replacement (tomorrow's latest and greatest chess-playing AI), which will instead be designed by humans (making humans still better at chess in a higher-level sense).

The reader might wonder why we would not further simplify Definition 1 and declare that X knows ϕ if and only if X would respond "yes" if X were asked: "Do you know ϕ? (Yes or no)". We will argue that this would be a poor candidate for an idealized knowledge definition.

Definition 2. *If X is an AGI and ϕ is an EA-sentence, say that X* quick-knows *ϕ if and only if X would respond "yes" if X were asked, "Do you know ϕ? (Yes or no)".*

The following should be contrasted with Theorem 2.

Theorem 3. *Let X be an AGI. The set of EA-sentences ϕ such that X quick-knows ϕ is computable.*

Proof. This follows from our assumption (in Section 2) that X is a deterministic machine. □

By Theorem 3, it seems that if we used Definition 2 as a knowledge definition, it would contradict Aristotle's claim that "it is difficult to be aware of whether one knows or not". It is more plausible that knowledge be *computably enumerable* (as in Theorem 2) than that knowledge be *computable*. A prototypical example of a set which is computably enumerable but not computable is: the consequences of Peano arithmetic[10] (hereafter PA). Said consequences cannot be computable, lest they could be used to solve the Halting Problem (because a Turing machine halts if and only if PA proves that it halts).

Theorem 4. *Let X be an AGI and assume X does not quick-know any falsehoods. At least one of the following is true:*

1. *There is an axiom of PA which X does not quick-know.*
2. *There exist PA-sentences ϕ and ψ such that X quick-knows ψ and X quick-knows $\psi \to \phi$, but X does not quick-know ϕ.*

Proof. It is well-known that a sentence ϕ is provable from PA if and only if there is a sequence $\phi_1, \ldots, \phi_n$ such that:

1. ϕ_n is ϕ.
2. For every i, either ϕ_i is an axiom of PA, or else there are $j, k < i$ such that ϕ_k is $\phi_j \to \phi_i$.

(Loosely speaking: proofs from PA can be carried out using no rules of inference besides Modus Ponens.) For any formula ϕ which PA proves, let $|\phi|$ be the smallest n such that there is a sequence $\phi_1, \ldots, \phi_n$ as above.

Call a PA-sentence ϕ *elusive* if PA proves ϕ but X does not quick-know ϕ. By Theorem 3, the fact that X does not quick-know any falsehoods, and the unsolvability of the Halting Problem, it follows that some elusive ϕ exists—otherwise, to computably determine whether or not a given Turing machine M

[10] We assume Peano arithmetic is true.

halts, we could simply ask X, "Do you know Turing machine M halts? (Yes or no)".

Since some elusive ϕ exists, there exists an elusive ϕ such that $|\phi|$ is as small as possible—that is, such that $|\phi| \leq |\psi|$ for every elusive ψ. Fix such a ϕ.

Case 1: ϕ is an axiom of PA. Then condition (1) of the theorem is satisfied, as desired.

Case 2: ϕ is not an axiom of PA. Let $\phi_1, \ldots, \phi_{|\phi|}$ be as in the first paragraph of this proof (so $\phi_{|\phi|}$ is ϕ). Then since ϕ is not an axiom of PA, there must be $j, k < |\phi|$ such that ϕ_k is $\phi_j \to \phi_{|\phi|}$. Now, the sequence $\phi_1, \ldots, \phi_k$ witnesses that PA proves ϕ_k and $|\phi_k| \leq k < |\phi|$; and the sequence $\phi_1, \ldots, \phi_j$ witnesses that PA proves ϕ_j and $|\phi_j| \leq j < |\phi|$. Thus, since ϕ was chosen to be elusive with $|\phi|$ as small as possible, it follows that ϕ_k and ϕ_j are not elusive. Thus, X quick-knows ϕ_j, and X quick-knows ϕ_k, but ϕ_k is $\phi_j \to \phi$. Thus condition (2) of the theorem is satisfied, as desired. □

Theorem 4 shows that Definition 2 makes a poor notion of idealized knowledge. An AGI should certainly know the axioms of PA, and should certainly be capable of the minimal logical reasoning needed to conclude ϕ from ψ and $\psi \to \phi$. And the way we have established the unsuitability of Definition 2 is nicely anticipated by the words of Aristotle quoted at the beginning of this subsection.

4 Quantified Modal Logic

Definition 1 only addresses sentences with no free variables. In this section, we will extend Definition 1 to formulas which possibly include free variables. We are essentially adapting a trick from Carlson [10].

Definition 3. *We define so-called* numerals*, which are EA-terms, one numeral* $\overline{n}$ *for each natural number* $n \in \mathbb{N}$*, by induction:* $\overline{0}$ *is defined to be* 0 *(the constant symbol for zero from PA) and for every* $n \in \mathbb{N}$*,* $\overline{n+1}$ *is defined to be* $S(\overline{n})$ *(where* S *is the successor symbol from PA).*

For example, the numeral $\overline{3}$ is the term $S(S(S(0)))$.

Definition 4. *If* ϕ *is an EA-formula (with free variables* $x_1, \ldots, x_k$*), and if* s *is an assignment mapping variables to natural numbers, then we define* ϕ^s *to be the sentence*

$$\phi(x_1|\overline{s(x_1)})(x_2|\overline{s(x_2)}) \cdots (x_k|\overline{s(x_k)})$$

obtained by substituting for each free variable x_i *the numeral* $\overline{s(x_i)}$ *for* x_i*'s value according to* s*.*

Example 2. Suppose $s(x) = 0$, $s(y) = 1$, and $s(z) = 3$. Then

$$((z > y + x) \wedge \forall x(K(z > y + x - x)))^s$$

is defined to be

$$((\overline{3} > \overline{1} + \overline{0}) \wedge \forall x(K(\overline{3} > \overline{1} + x - x)))$$

(note that the numeral is not substituted for the later occurrences of x because these are bound by the $\forall x$ quantifier).

Definition 5. *If ϕ is any $\mathscr{L}$-formula, and s is any assignment mapping variables to $\mathbb{N}$, we say that X knows ϕ (with variables interpreted by s) if and only if X knows ϕ^s according to Definition 1.*

Armed with Definition 5, the Tarskian notion [14] of truth can be extended to EA.

Example 3. Assume an AGI X is clear from context. Suppose ϕ is an EA-formula, of one free variable x, which expresses "the xth Turing machine eventually halts". Suppose we want to assign a truth value to the formula

$$\exists x(\neg K(\phi) \wedge \neg K(\neg\phi)).$$

We proceed as follows.

- Following Tarski, we should declare $\exists x(\neg K(\phi) \wedge \neg K(\neg\phi))$ is true if and only if for every assignment s mapping variables to $\mathbb{N}$, $\exists x(\neg K(\phi) \wedge \neg K(\neg\phi))$ is true (with variables interpreted by s).
- By the semantics of $\exists$, the above is true if and only if for every assignment s, there is some $n \in \mathbb{N}$ such that $\neg K(\phi) \wedge \neg K(\neg\phi)$ is true (with variables interpreted by $s(x|n)$), where $s(x|n)$ is the assignment that agrees with s except for mapping x to n.
- By Definition 5, this is the case if and only if for every assignment s there is some $n \in \mathbb{N}$ such that X does not know $\phi^{s(x|n)}$ (according to Definition 1) and X does not know $\neg\phi^{s(x|n)}$ (according to Definition 1).
- By Definition 4 and the fact that x is the only free variable in ϕ, the above is the case if and only if there is some $n \in \mathbb{N}$ such that X does not know $\phi(x|\overline{n})$ (according to Definition 1) and X does not know $\neg\phi(x|\overline{n})$ (according to Definition 1).

So ultimately, we consider $\exists x(\neg K(\phi) \wedge \neg K(\neg\phi))$ to be true if and only if there is some $n \in \mathbb{N}$ such that, in response to the command "Enumerate all the EA-sentences which you know", X would not include $\phi(x|\overline{n})$ nor $\neg\phi(x|\overline{n})$ in the resulting enumeration.

5 Translating knowledge formulas

In this section, we will look at some formulas about knowledge and translate them into statements about AGI stimulus-response, using Definitions 1 and 5. First, we will start by translating some simple axioms of knowledge, to give the reader a feel for how this translation works. Then, we will advance to the examples we mentioned in the Introduction, and closely related examples.

Although the statements in the following example may seem plausible, our purpose is not to claim that every AGI must satisfy them. Rather, they serve to classify AGIs: for each axiom schema, one can speak of AGIs who satisfy that axiom schema, and of AGIs who do not satisfy it.

Example 4. (Basic axioms of knowledge) The following axiom schemas, in the language of EA, are taken from Carlson [10] (we restrict them to sentences for purposes of simplicity).

- (E1) $K(\phi)$ whenever ϕ is valid (i.e., true in every model). Translated for an AGI X using Definition 1, this becomes: "If commanded to enumerate his knowledge in EA, X will include all valid EA-sentences in the resulting list."
- (E2) $K(\phi \to \psi) \to K(\phi) \to K(\psi)$. This becomes: "If commanded to enumerate his knowledge in EA, if X would include $\phi \to \psi$ and if X would also include ϕ, then X would also include ψ."
- (E3) $K(\phi) \to \phi$. This becomes: "If commanded to enumerate his knowledge in EA, the resulting statements X enumerates would be true."
- (E4) $K(\phi) \to K(K(\phi))$. This becomes: "If commanded to enumerate his knowledge in EA, if X would list ϕ, then X would also list $K(\phi)$."

Our purpose in Example 4 is not to declare that an AGI must satisfy E1–E4. Rather, our goal is to translate these modal logical axioms into AGI language—note that the translations in quotation marks in Example 4 do not directly depend on the AGI's knowledge, but only on the AGI's stimulus-response. When studying AGI in broadest generality, even E3, the factivity of knowledge, might be questioned (certain AGIs might satisfy it and other AGIs might not). By translating E3 into a concrete statement about the AGI's stimulus-response, we can talk about "AGIs who satisfy E3" or "AGIs who fail E3," without getting stuck on hard questions like "What does it mean to know something?"

Example 5. (Reinhardt's strong mechanistic thesis [19] [20] [10]) Reinhardt suggested the EA-schema

$$\exists e \forall x (K(\phi) \leftrightarrow x \in W_e)$$

as a formalization of the mechanicalness of the knower. Here, W_e is the eth computably enumerable set of natural numbers[11] (W_e can also be thought of as the set of naturals enumerated by the eth Turing machine). For simplicity, consider the case where x is the lone free variable of ϕ. Then in terms of Definition 5, the schema becomes: "If X were commanded to enumerate his knowledge in the language of EA, then the set of $n \in \mathbb{N}$ such that X would include $\phi(x|\overline{n})$ in the resulting list, would be computably enumerable." If Φ is the universal closure[12] of the above EA-schema, then the schema $K(\Phi)$ is *Reinhardt's strong mechanistic thesis*. Reinhardt conjectured that his strong mechanistic thesis is consistent with basic axioms about knowledge (i.e., that it is possible for a knowing machine to know that it is a machine). This conjecture was proved by Carlson [10] using sophisticated structural results about the ordinals [9]. See [4] for an elementary proof of a weaker version of the conjecture.

[11] It can be shown that W_e is definable in the language of Peano arithmetic, therefore we can use expressions like "$x \in W_e$" in EA-formulas as shorthand.

[12] A *universal closure* of a formula ϕ is a sentence $\forall x_1 \cdots \forall x_k \phi$, and the *universal closure* of a schema of formulas is the schema of universal closures of those formulas.

Example 6. (Reinhardt's absolute version of Gödel's incompleteness theorem) If we vary the formula from Example 5 by requiring that the knower know the value of e, we obtain:

$$\exists e K(\forall x(K(\phi) \leftrightarrow x \in W_e)).$$

Carlson [10] glosses this schema in English as: "I am a Turing machine, and I know which one." Reinhardt showed that this schema is *not* consistent with basic axioms about knowledge. Following Carlson's gloss, this shows that it is impossible for a suitably idealized AGI to know its own code[13].

Remark 2. As far as I know, AGI has not yet received much attention in the mathematical logical literature. Instead, mathematical logicians tend to concern themselves with *knowing agents* or *knowing machines*. Presumably, every suitably idealized AGI is a knowing agent and a knowing machine, but certainly not every knowing agent (or knowing machine) is an AGI. Thus, in general, inconsistency results about knowing agents or knowing machines carry directly over to AGIs (if no knowing agent, or no knowing machine, can satisfy some property, then in particular no suitably idealized AGI can either). Consistency results do not generally carry over to AGIs (it may be possible for a knowing agent or a knowing machine to satisfy some property, but it might be that none of the knowing agents or knowing machines which satisfy that property are AGIs). Nevertheless, a consistency result about knowing agents or knowing machines should at least count as evidence in favor of the corresponding consistency result for AGIs, at least if there is no clear reason otherwise. In the examples above:

- Reinhardt's strong mechanistic thesis (Example 5) was proven to be consistent with basic knowledge axioms, so it is possible for a knowing machine to know that it is a machine (without necessarily knowing which machine). Since not every knowing machine is an AGI, it might still be impossible for an AGI to know it is a machine. But the consistency of Reinhardt's strong mechanistic thesis at least suggests evidence that an AGI can know it is a machine.
- Reinhardt's absolute version of the incompleteness theorem (Example 6) is an inconsistency result. As such, it transfers over directly to AGI, proving that no suitably idealized AGI can know its own code[14].

Example 7. (Intuitive Ordinal Intelligence) In [5] we defined an intelligence measure for idealized mechanical knowing agents (who are aware of the computable ordinals) as follows. If A is such a knowing agent, we define the intelligence of A to be the supremum of the set of ordinals α such that α has some code c such that A knows that c is a code of a computable ordinal. In [6] we specialized

[13] We have pointed out elsewhere [3] that (i) Reinhardt implicitly assumes that the knower knows its own truthfulness; and (ii) it is possible for a knowing machine to know its own code if it is allowed to be ignorant of its own truthfulness, despite still being truthful. See [1] and [2] for some additional discussion.

[14] Or rather, its own code and its own truthfulness—we have pointed out [3] that Reinhardt implicitly assumes the knower knows its own truthfulness.

this to AGIs, and called it *Intuitive Ordinal Intelligence*. Let $\mathscr{L}$ be a language like EA but including an additional predicate symbol O for the set of codes of computable ordinals. Modifying Definition 1 accordingly, we can systematically perform said specialization to AGIs, and it becomes: "The Intuitive Ordinal Intelligence of an AGI X is the supremum of the set of ordinals α such that α has some code c such that X would include $O(\overline{c})$ in the resulting enumeration if we asked X to enumerate all the $\mathscr{L}$-sentences that he knows."

6 Conclusion

What does it mean to know something? This is a difficult question and there probably is no one true answer. In the field of AGI, how can we systematically investigate the theoretical properties of knowledge, when different AGIs might not even agree about what knowledge really means? So motivated, we have proposed an elegant way to brush these philosophical questions aside. In Definition 1, we declare that an AGI knows an EA-sentence if and only if that AGI would enumerate that sentence if commanded:

"Enumerate all the EA-sentences which you know"

(this definition might look circular at first glance but we have argued that it is not; see Subsection 3.1). In Definition 5 we extended this to formulas with free variables, not just sentences.

This universal knowledge definition sets the study of AGI knowledge on a firmer theoretical footing. In Section 5 we give examples of how our definition can serve as a bridge to translate knowledge-related formulas from mathematical logic into the realm of AGI.

Acknowledgments

We gratefully acknowledge Alessandro Aldini, Phil Maguire, Brendon Miller-Boldt, Philippe Moser, and the reviewers for comments and feedback.

References

1. Aldini, A., Fano, V., Graziani, P.: Do the self-knowing machines dream of knowing their factivity? In: AIC. pp. 125–132 (2015)
2. Aldini, A., Fano, V., Graziani, P.: Theory of knowing machines: Revisiting Gödel and the mechanistic thesis. In: International Conference on the History and Philosophy of Computing. pp. 57–70. Springer (2015)
3. Alexander, S.A.: A machine that knows its own code. Studia Logica **102**(3), 567–576 (2014)
4. Alexander, S.A.: Fast-collapsing theories. Studia Logica **103**(1), 53–73 (2015)
5. Alexander, S.A.: Measuring the intelligence of an idealized mechanical knowing agent. In: CIFMA (2019)

6. Alexander, S.A.: AGI and the Knight-Darwin law: why idealized AGI reproduction requires collaboration. In: International Conference on Artificial General Intelligence. pp. 1–11. Springer (2020)
7. Aliman, N.M., Elands, P., Hürst, W., Kester, L., Thórisson, K.R., Werkhoven, P., Yampolskiy, R., Ziesche, S.: Error-correction for AI safety. In: International Conference on Artificial General Intelligence. pp. 12–22. Springer (2020)
8. Aristotle: Posterior analytics. In: Barnes, J., et al. (eds.) The Complete Works of Aristotle. Princeton University Press (1984)
9. Carlson, T.J.: Ordinal arithmetic and Σ_1-elementarity. Archive for Mathematical Logic **38**(7), 449–460 (1999)
10. Carlson, T.J.: Knowledge, machines, and the consistency of Reinhardt's strong mechanistic thesis. Annals of Pure and Applied Logic **105**(1-3), 51–82 (2000)
11. Cerone, A., Fazli, S., Malone, K., Pietarinen, A.V.: Interdisciplinary aspects of cognition. In: CIFMA (2019)
12. Elton, D.: Self-explaining AI as an alternative to interpretable AI. In: International Conference on Artificial General Intelligence (2020)
13. Goertzel, B.: Artificial general intelligence: concept, state of the art, and future prospects. Journal of Artificial General Intelligence **5**, 1–48 (2014)
14. Hodges, W.: Tarski's truth definitions. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, fall 2018 edn. (2018)
15. Ichikawa, J.J., Steup, M.: The analysis of knowledge. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, summer 2018 edn. (2018)
16. Knuth, D.E.: Two notes on notation. The American Mathematical Monthly **99**(5), 403–422 (1992)
17. Maguire, P., Moser, P., Maguire, R.: Are people smarter than machines? Croatian Journal of Philosophy **20**(1), 103–123 (2020)
18. Plato: Theaetetus. In: Cooper, J.M., Hutchinson, D.S., et al. (eds.) Plato: complete works. Hackett Publishing (1997)
19. Reinhardt, W.N.: Absolute versions of incompleteness theorems. Nous **19**, 317–346 (1985)
20. Reinhardt, W.N.: Epistemic theories and the interpretation of Gödel's incompleteness theorems. Journal of Philosophical Logic **15**(4), 427–474 (1986)
21. Shapiro, S.: Epistemic and intuitionistic arithmetic. In: Studies in Logic and the Foundations of Mathematics, vol. 113, pp. 11–46. Elsevier (1985)
22. Wellmer, A.: Skepticism in interpretation. In: Conant, J.F., Kern, A. (eds.) Varieties of Skepticism: Essays after Kant, Wittgenstein, and Cavell. Walter de Gruyter (2014)
23. Williamson, T.: Knowledge, context, and the agent's point of view. In: Preyer, G., Peter, G. (eds.) Contextualism in philosophy: Knowledge, meaning, and truth, pp. 91–114. Oxford University Press (2005)
24. Yampolskiy, R.: On controllability of artificial intelligence. Technical report (2020)