Short-circuiting the definition of mathematical knowledge for an AGI

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 (AGI), a pragmatic way to define such an agent's knowledge is as follows. We declare the AGI to know a certain statement (in a given mathematical language) if and only if, when commanded to enumerate all the statements that it knows in that language (with whatever internal knowledge definition it has), said AGI would include said statement in the resulting enumeration. This elegantly solves the problem that different AGIs may have very different internal knowledge definitions and yet we want to be able 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

1 Introduction

It is difficult to define knowledge, or what it means to know something. In Plato's dialogs, 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 (see [8]).

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 exceptional area where the question becomes important even to pragmatists is in the area of Artificial General Intelligence (AGI). In AGI research, these old philosophical conundrums rear their ugly heads once more.

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 it means for a suitably idealized AGI to know a mathematical sentence[2]. Our

[1] Perhaps the best example being in the *Theaetetus* [9].

[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.

proposed definition is short and sweet: we say that an AGI knows a mathematical sentence (in some standard mathematical language $\mathscr{L}$) if and only if that sentence would be among the sentences which that AGI would enumerate if that AGI were commanded: "Enumerate, in the language $\mathscr{L}$, every mathematical sentence which is both expressible in $\mathscr{L}$ and part of your own mathematical knowledge."

Our proposed definition is not directly intended for methodological purposes—it would not directly be helpful in the quest to construct an AGI. Instead, it is intended for the purpose of understanding the properties of AGI (some of which we discuss in Section 5). We hope that a better understanding of theoretical properties of AGIs will indirectly help in the eventual creation of same.

The structure of this paper is as follows.

- In Section 2 we discuss the AGIs for whose knowledge we are attempting to propose a definition.
- In Section 3 we propose a knowledge definition for AGIs for sentences in some standard mathematical language.
- In Section 4 we extend our knowledge definition to formulas with free variables (when the language in question is arithmetical).
- In Section 5 we use this knowledge definition as a bridge to allow some results from mathematical logic to inform AGI.
- In Section 6 we summarize and make concluding remarks.

2 Idealized AGIs

In this paper, we approach AGI using what Goertzel [6] 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." We assume the AGI is designed in such a way as to understand English commands, and is capable (if so commanded) of outputting data in any desired format, provided that format can be expressed (say) as a computer file or as a string of unicode characters. Below, we will state some additional idealized assumptions, but first we will need a preliminary definition.

Definition 1. *By a* standard mathematical language*, we mean a mathematical language for which an intended interpretation is implicitly understood.*

For example, the language of Peano Arithmetic is a standard mathematical language, the obvious intended interpretation being the model with universe $\mathbb{N}$ which interprets the arithmetical symbols of Peano Arithmetic in the usual ways. Presumably an AGI suitably familiar with the mathematical literature should be aware of this intended interpretation for Peano Arithmetic.

Armed with Definition 1, we are ready to articulate the key additional assumption which we make about an AGI.

Definition 2. *An AGI X is* obedient *if, for every standard mathematical language $\mathscr{L}$, the following is true. When commanded to enumerate the $\mathscr{L}$-sentences that he knows, X will obey the command: X will enumerate exactly the $\mathscr{L}$-sentences that he knows (according to his own internal definition of knowledge, whatever that may be).*

Note that in Definition 2, we do not require the AGI to use the definition of knowledge that we are proposing in this paper (if we did so, our definition would be circular).

Definition 2 would be inappropriate for human agents, who are forgetful and error-prone and who tend to resist tedious tasks such as enumerating endless lists of mathematical sentences. But arguably Definition 2 is plausible for a sufficiently idealized AGI. Such an AGI can presumably perform calculations with no risk of mechanical error. And such an AGI should have unlimited patience and have no problem tediously enumerating mathematical sentences as long as memory-banks and electricity are available.

3 An elegant definition of mathematical knowledge

If X is an idealized AGI which is obedient, we define the mathematical knowledge of X (as far as sentences go) as follows (where $\mathscr{L}$ denotes a standard mathematical language).

Definition 3. *For any $\mathscr{L}$-sentence ϕ, we say that an obedient AGI X knows ϕ if and only if X would eventually list ϕ among the $\mathscr{L}$-sentences which X would list if X were commanded: "Enumerate, in the language $\mathscr{L}$, every mathematical sentence which is both expressible in $\mathscr{L}$ and part of your own mathematical knowledge."*

One of the strengths of Definition 3 is that it is uniform across different AGIs: many different AGIs might internally operate based on different definitions of knowledge, but Definition 3 works equally well for all these different AGIs irrespective of those different internal knowledge definitions[3].

Although Definition 3 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.

Theorem 1. *Suppose X is an obedient AGI, and let $\mathscr{L}$ be a standard mathematical language. For any $\mathscr{L}$-sentence ϕ, the following are equivalent:*

1. *X is considered to know ϕ (based on Definition 3).*
2. *X is considered to know ϕ (based on X's own inernal definition of knowledge).*

[3] 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 [4].

Proof. By Definition 3, (1) is equivalent to the statement that X would include ϕ in the list which X would output if X were commanded to output all the $\mathscr{L}$-sentences that X knows. Since X is obedient, X would output ϕ in that list if and only if (2).

3.1 Languages with Knowledge Operators

Definition 3 is particularly interesting when $\mathscr{L}$ itself contains an operator for the agent's knowledge. An example of such a language would be the language of Epistemic Arithmetic (or EA) from [12], which consists of the language of Peano Arithmetic with the addition of an operator K for knowledge: $K(1+1=2)$ should be read as something like "I know $1{+}1=2$" or "the knower knows $1{+}1=2$". In the context of this paper, if $\mathscr{L}_0$ is a standard mathematical language, and if $\mathscr{L}$ is obtained from $\mathscr{L}_0$ by the addition of a knowledge operator K, then we also consider $\mathscr{L}$ to be a standard mathematical language. The intended model of $\mathscr{L}$ shall have the same universe and interpretation of $\mathscr{L}_0$-symbols as the intended model of $\mathscr{L}_0$. As for K, the intended interpretation (by an AGI X) of a formula $K(\phi)$ shall be that X knows ϕ (according to the AGI's internal definition of knowledge).

Example 1. Applying Definition 3 to the language of EA, we consider an obedient AGI X to know $K(1+1=2)$ if and only if that AGI would output $K(1+1=2)$ when commanded to output all sentences that X knows in the language of EA. By the intended interpretation of EA, X would (when so commanded) output $K(1+1=2)$ if and only if X knows that he knows $1+1=2$.

4 Quantified Modal Logic

Definition 3 only addresses sentences with no free variables. For suitable languages, we will extend this to formulas which possibly include free variables. Here, we are essentially adapting a trick from Carlson [2].

Definition 4. *A standard mathematical language $\mathscr{L}$ is said to be* arithmetical *if the following requirements hold.*

1. *$\mathscr{L}$ contains all the symbols of Peano Arithmetic.*
2. *$\mathscr{L}$'s intended model has universe $\mathbb{N}$ and interprets the symbols of Peano Arithmetic in the usual ways.*

Definition 5. *If $\mathscr{L}$ is arithmetical, then we define so-called* numerals, *which are $\mathscr{L}$-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 Peano Arithmetic) and for every $n \in \mathbb{N}$, $\overline{n+1}$ is defined to be $S(\overline{n})$ (where S is the successor symbol from Peano Arithmetic).*

For example, the numeral $\overline{5}$ is the term $S(S(S(S(S(0)))))$.

Definition 6. *If $\mathscr{L}$ is arithmetical and ϕ is an $\mathscr{L}$-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 7. *If $\mathscr{L}$ is arithmetical, ϕ is any $\mathscr{L}$-formula, and s is any assignment mapping variables to $\mathbb{N}$, we say that X knows ϕ (according to s) if and only if X knows ϕ^s according to Definition 3.*

Armed with Definition 7, the Tarskian notion [7] of truth can be extended to a complete semantics for knowledge in any arithmetical language with exactly one knowledge operator K.

Example 3. Assume an obedient AGI X is clear from context. Suppose ϕ is a formula of one free variable x, in the language of EA, which expresses "the xth Turing machine eventually halts". Suppose we want to assign a truth value to the formula $\exists x(\neg K(\phi)\wedge\neg K(\neg\phi))$.

- Following Tarski, we should declare $\exists x(\neg K(\phi)\wedge\neg K(\neg\phi))$ is true if and only if there is some assignment s mapping variables to $\mathbb{N}$ such that both $K(\phi)$ and $K(\neg\phi)$ are false according to s.
- By Definition 7, this is the case if and only if there is some s such that X does not know ϕ^s and X does not know $\neg\phi^s$ (according to Definition 3).
- This is the case if and only if there is some s such that X would not list ϕ^s nor $\neg\phi^s$ if X were commanded to enumerate his own knowledge in the language of EA.
- Since ϕ has just one free variable x, it follows that the above is equivalent to: there is some $n\in\mathbb{N}$ such that X would not list $\phi(x|\overline{n})$ nor $\neg\phi(x|\overline{n})$ if X were commanded as above.

5 Knowledge axioms

In this section, we will look at some axioms of knowledge and interpret them in the context of AGI in terms of Definitions 3 and 7.

Example 4. (Basic axioms of knowledge) The following axiom schemas, in the language of EA, are taken from Carlson [2] (we restrict them to sentences for purposes of simplicity).

- (E1) $K(\phi)$ whenever ϕ is valid (i.e., a tautology). Interpreted for an AGI X using Definition 3, this becomes: "If commanded to enumerate his knowledge in EA, X will include all that language's tautologies in the resulting list." This is plausible because the set of tautologies in a given computable language is computable, and an AGI should have no problem enumerating them.
- (E2) $K(\phi \rightarrow \psi) \rightarrow K(\phi) \rightarrow K(\psi)$. This becomes: "If commanded to enumerate his knowledge in EA, if X would include $\phi \rightarrow \psi$ and if X would also include ϕ, then X would also include ψ." This is plausible because an AGI should certainly be capable of basic logical reasoning.
- (E3) $K(\phi) \rightarrow \phi$. This becomes: "If commanded to enumerate his knowledge in EA, the resulting statements X enumerates will be true." This is plausible since knowledge is widely regarded as having truthfulness as one of its requirements. Truthfulness is not a requirement in the definition proposed in this paper, but for any particular AGI, truthfulness is probably a requirement of that AGI's internal definition of knowledge. There is no need to worry about the AGI being misinformed about contingent facts about the physical world, because EA is a purely mathematical language in which no such contingent facts are expressible.
- (E4) $K(\phi) \rightarrow K(K(\phi))$. This becomes: "If commanded to enumerate his knowledge in EA, if X would list ϕ, then X would also list $K(\phi)$." This is plausible because presumably when X enumerates ϕ in response to the command, X should in some sense understand why he is enumerating ϕ, namely because he knows ϕ—so X should therefore know that he knows ϕ, which knowledge is expressible in EA as $K(K(\phi))$.

Example 5. (Reinhardt's strong mechanistic thesis [10] [11] [2]) 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 (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 7, 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." This is not just plausible but obvious[4], since X himself is an AGI and thus presumably a computer.

[4] What is much less obvious is the fact that it is consistent for the knower himself to know the schema in question. This was conjectured by Reinhardt, and proved by Carlson [2]. See [1] for some further discussion.

Example 6. (The Epistemic Church's Thesis [5] [3]) The following EA-schema has been suggested as a kind of epistemic formalization of Church's Thesis:

$$(\forall x \exists y (K(\phi))) \rightarrow (\exists e K(\forall x \exists y (E(e,x,y) \wedge \phi))),$$

where $E(e,x,y)$ is an EA-formula which expresses that the eth Turing machine outputs y on input x. This becomes: "Suppose X were commanded to enumerate his EA-knowledge. Assume there is a (not necessarily computable) function $f : \mathbb{N} \rightarrow \mathbb{N}$ such that for every $n \in \mathbb{N}$, $\phi(x|\overline{n})(y|\overline{f(n)})$ is included in the resulting enumeration. Then in fact there is a *computable* function $f' : \mathbb{N} \rightarrow \mathbb{N}$ (with Turing index e) such that f' has the same property and such that the enumeration includes a statement that f' has said property." This beautiful formalism seems to capture the AGI's self-reflection ability. We can imagine the AGI dutifully enumerating statement after statement and as she goes, she discovers and predicts patterns in her own enumeration. Flagg proved that the Epistemic Church's Thesis is consistent with basic axioms of knowledge [5], and Carlson proved that it is also consistent with Reinhardt's strong mechanistic thesis [3].

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 amongst eachother about what knowledge really means? So motivated, we have proposed an elegant way to brush these philosophical questions aside. In Definition 3, we declare that an AGI knows a sentence in a standard mathematical language if and only if that AGI would enumerate that sentence if commanded to enumerate all the things it knows (according to its own internal knowledge definition) in that mathematical language. In Definition 7 we extend this to formulas with free variables, not just sentences.

This one-size-fits-all 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 results from mathematical logic into the realm of AGI.

References

1. 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)
2. 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)
3. Carlson, T.J.: Collapsing knowledge and epistemic church's thesis. Gödel's Disjunction: The Scope and Limits of Mathematical Knowledge pp. 129–148 (2016)
4. Elton, D.: Self-explaining ai as an alternative to interpretable ai. In: ICAGI (forthcoming)

5. Flagg, R.C.: Church's thesis is consistent with epistemic arithmetic. In: Studies in Logic and the Foundations of Mathematics, vol. 113, pp. 121–172. Elsevier (1985)
6. Goertzel, B.: Artificial general intelligence: concept, state of the art, and future prospects. JAGI **5**, 1–48 (2014)
7. 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)
8. 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)
9. Plato: Theaetetus. In: Cooper, J.M., Hutchinson, D.S., et al. (eds.) Plato: complete works. Hackett Publishing (1997)
10. Reinhardt, W.N.: Absolute versions of incompleteness theorems. Nous **19**, 317–346 (1985)
11. Reinhardt, W.N.: Epistemic theories and the interpretation of gödel's incompleteness theorems. Journal of Philosophical Logic **15**(4), 427–474 (1986)
12. Shapiro, S.: Epistemic and intuitionistic arithmetic. In: Studies in Logic and the Foundations of Mathematics, vol. 113, pp. 11–46. Elsevier (1985)