AGI from the perspectives of Categorical Logic and Algebraic Geometry

King-Yin Yan[0009−0007−8238−2442]

general.intelligence@gmail.com

Abstract. To "situate" AGI in the context of some current mathematics, so that readers can more easily see whether current mathematical ideas can be fruitfully applied to AGI.

Keywords: AGI · categorical logic · Curry-Howard isomorphism · homotopy type theory · algebraic geometry · topos theory.

1 Goal of this research

The bottleneck of AGI development is the speed of learning algorithms. The daily cost of training GPT-4 was rumored to be $100M by Sam Altman. To speed up learning, one needs **inductive biases**, according to the **No Free Lunch theorem**. One principled way to introduce inductive bias is by the structure of logic. The reason being that, if humans have discovered the structure of logic in this world, an intelligent program may re-discover the same structure. So our question is: what is the mathematical structure of logic?

2 Conclusions thus far

The conclusions of this paper are mostly *negative*. That is to say, the mathematical structures described here seem unable to offer practical ways to accelerate AGI, unless the reader can discover more ingenious ideas. Nevertheless the author hopes the presentation of these ideas thus far can help the readers on their way.

In each section below, we look at one aspect of the categorical structure of logic and speculate on how it might aid AGI architecture.

2.1 Where is GPT?

GPT [4] can be regarded as a **logic consequence operator** mapping from the space of propositions to itself (as a **set-valued map**), as shown in Fig.1.

Without loss of generality, GPT can be seen as acting on propositions, as probability distributions on tokens are equivalent to probability distributions on sentences (propositions), see Fig. 2.

To what extent can we say that GPT outputs a vector in a Curry-Howard type-theoretic space? The answer is: only in an abstract sense, but not literally.

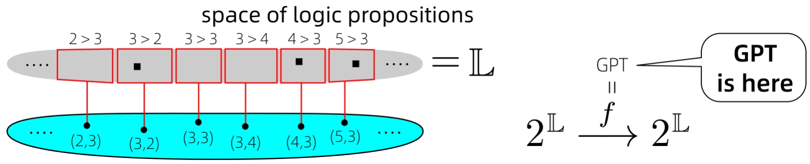


Fig. 1. (Left) A sheaf of propositions over pairs of natural numbers; (Right) The function space where GPT lives.

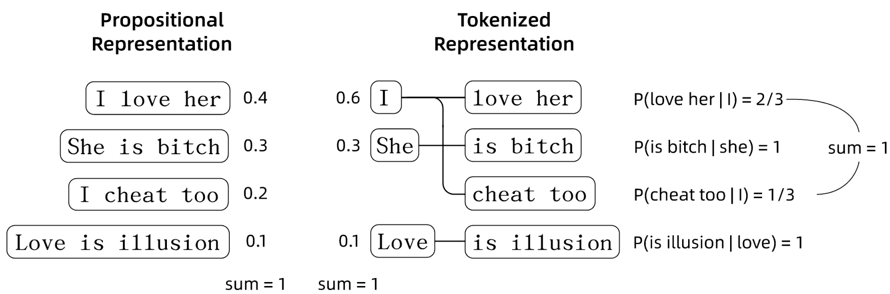


Fig. 2. The equivalence of probability distributions over propositions and over tokens. For simplicity the decomposition for just the leading token is shown.

A neural network (eg. GPT) always maps an input to the the same output vector, as it is a *deterministic* map. Thus it seems meaningless to ask what is the meaning of the *neighborhood* of an output vector, if the network never goes there during inference time.

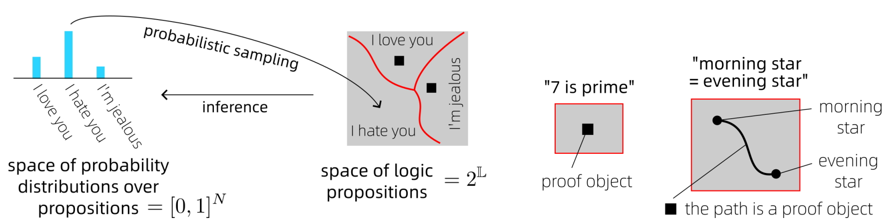


Fig. 3. (Left) How probabilistic inference in GPT results in deterministic vector positions; (Right) In HoTT, a path is a proof object of an identity type.

Fig.3(Left) illustrates why even despite probabilistic sampling, the outputs of GPT seem to follow fixed trajectories (as soon as learning has finished). Nevertheless, if we artificially "perturb" the input, the output probability distribution will change *smoothly*, say from favoring token A to favoring token B, as neural networks are always *differentiable* functions. One can define the **boundary** between tokens A and B as where their probabilities reverse in magnitude. This forms a Voronoi-like tessellation of the output space that can be regarded as Curry-Howard type-spaces.

2.2 Homotopy type theory

For a neural network to process HoTT information, it needs a way to

Therefore we can conclude that current neural networks are unlikely to process HoTT structures.

2.3 Commutativity of $\wedge$ and $\vee$

Permutation symmetry is the easiest to recognize and implement [8] [7]. It is well-known the Transformer is **equivariant** to permutations of inputs. This may be seen as evidence that Transformer **tokens** are proposition-like entities, with the caveat that we may be confusing the propositional level with the sub-propositional level of atomic concepts. An easy-to-remember example is: $I \heartsuit U \neq U \heartsuit I$, but $I \heartsuit U \wedge U \heartsuit I = U \heartsuit I \wedge I \heartsuit U$.

2.4 $\forall$ and $\exists$ as adjunctions

It seems difficult to translate this structure into a structural modification of neural networks. From our experience in logic-based AI, logic rules are usually implicitly $\forall$-quantified, and $\exists$ is usually implicit by the **Closed-World Assumption**.

The following two conditions concern the well-behavior of quantification, as described in [5] and on nLab [1] [2]:
The **Beck-Chevalley condition**.
The **Frobenius condition** corresponds in logic to saying that $\exists x.(\phi \wedge \psi)$ is equivalent to $(\exists x.\phi) \wedge \psi$ if x is not free in ψ.

2.5 Predicates as fibration

2.6 Iteration of $\vdash$ and Looped Transformers

2.7 Modal logic

Grothendieck topology is re-captured as a modal operator.

2.8 Algebraic geometry

The fundamental duality in algebraic geometry is:

$$\left\{ \begin{matrix} \text{spaces, or} \\ \text{varieties} \end{matrix} \right\} \longleftrightarrow \left\{ \begin{matrix} \text{commutative} \\ \mathbf{k}\text{-algebras} \end{matrix} \right\} \tag{1}$$

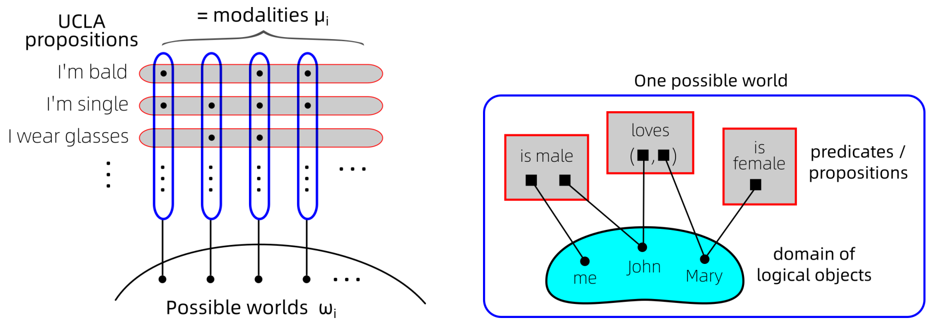


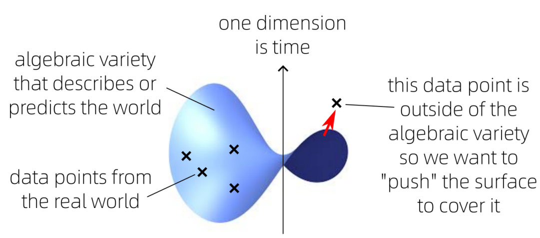
Fig. 4. (Left) The set underneath are indexes to possible worlds, for example $\omega_i = \{1, 2, 3...\}$. Each "stalk" represents a possible world, together they form a fibration over the base space.

within this correspondence, we have:

$$\left\{ \begin{array}{c} \text{points in} \\ \text{geometry} \end{array} \right\} \longleftrightarrow \{ \text{prime ideals} \} \tag{2}$$

An approach suggested by Yuri Manin is to turn logic into an algebra, such as the Boolean ring (but this can only handle propositional logic). Varieties defined by such Boolean polynomials [6] live in the space $\mathbb{Z}_2^n$, the discrete hypercube.

My visualization of the "Grothendieck picture" of AGI is like this:

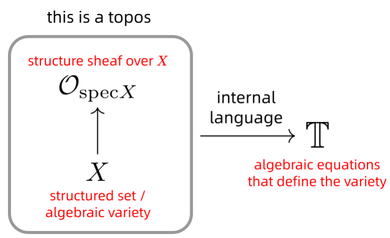


One of the interesting discoveries in categorical logic is that every topos admits an **internal language**. This is a simple consequence of Curry-Howard: since a type-space corresponds to a logic proposition, and categorical logic interprets a type-space as an object in the category, thus every category (satisfying extra conditions) can be interpreted as having an "internal" logic. The converse of this correspondence is the **classifying topos** of a logic theory:

$$\underset{\text{classifying topos}}{\mathcal{E}_{\mathbb{T}}} \overset{\text{internal language}}{\rightleftarrows} \underset{\text{theory}}{\mathbb{T}} \tag{3}$$

An alternative approach by Ingo Blechschmidt [3].

What is the "internal logic" of classical algebraic geometry? It may allow us to reason about polynomials, but that is different from AI logic.



(4)

References

[1] nLab authors. *Beck-Chevalley condition*. URL: `https://ncatlab.org/nlab/show/Beck-Chevalley+condition`.

[2] nLab authors. *Frobenius reciprocity*. URL: `https://ncatlab.org/nlab/show/Frobenius+reciprocity`.

[3] Ingo Blechschmidt. "Using the internal language of toposes in algebraic geometry". PhD thesis. University Augsburg, 2017. URL: `arXiv:2111.03685v1`.

[4] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: `2005.14165 [cs.CL]`.

[5] Bart Jacobs. *Categorical logic and type theory*. Elsevier, 1999.

[6] Samuel Lundqvist. *Boolean ideals and their varieties*. 2015. arXiv: `1211.3398 [math.AC]`.

[7] Qi et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space". In: *Advances in Neural Information Processing Systems* (2017), pp. 5105–5114.

[8] Zaheer et al. "Deep sets". In: *Advances in Neural Information Processing Systems* 30 (2017), pp. 3391–3401.