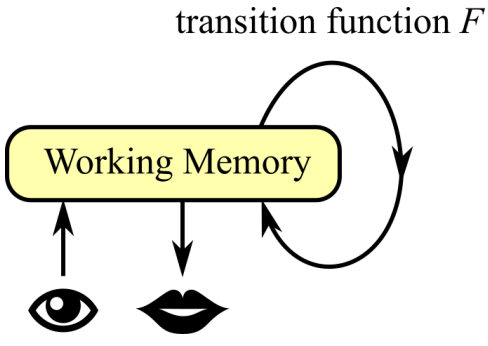
Logicalized AGI basics

This is our basic **RL** (reinforcement learning) setup[1]:



(2)

WM (working memory) is **updated** by a transition function. If this transition function is trained in a **closed loop**, it may be able to "explain" (in the sense of predicting) the input data.

In classical logic-based AI, **inductive learning** means searching for a **theory** T (= set of logic rules) that "explains" or **implies** positive examples but not negative ones:

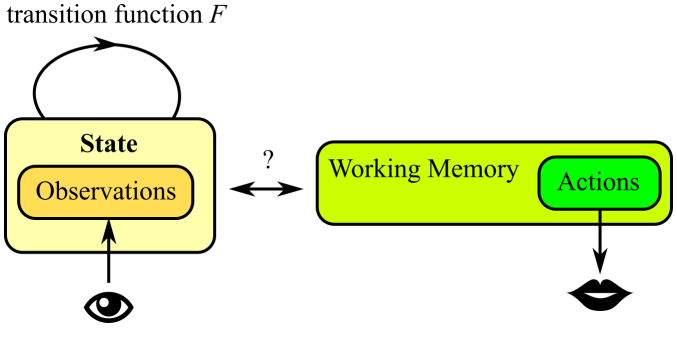
$$T \vdash e^+, \quad T \nvdash e^- \tag{3}$$

While logic learning is powerful, it relies on **combinatorial search** and was too inefficient, which caused "AI Winter".

The main thrust of my argument is that training the function F in the RL closed loop can effectively **discover** the logic theory T.

---

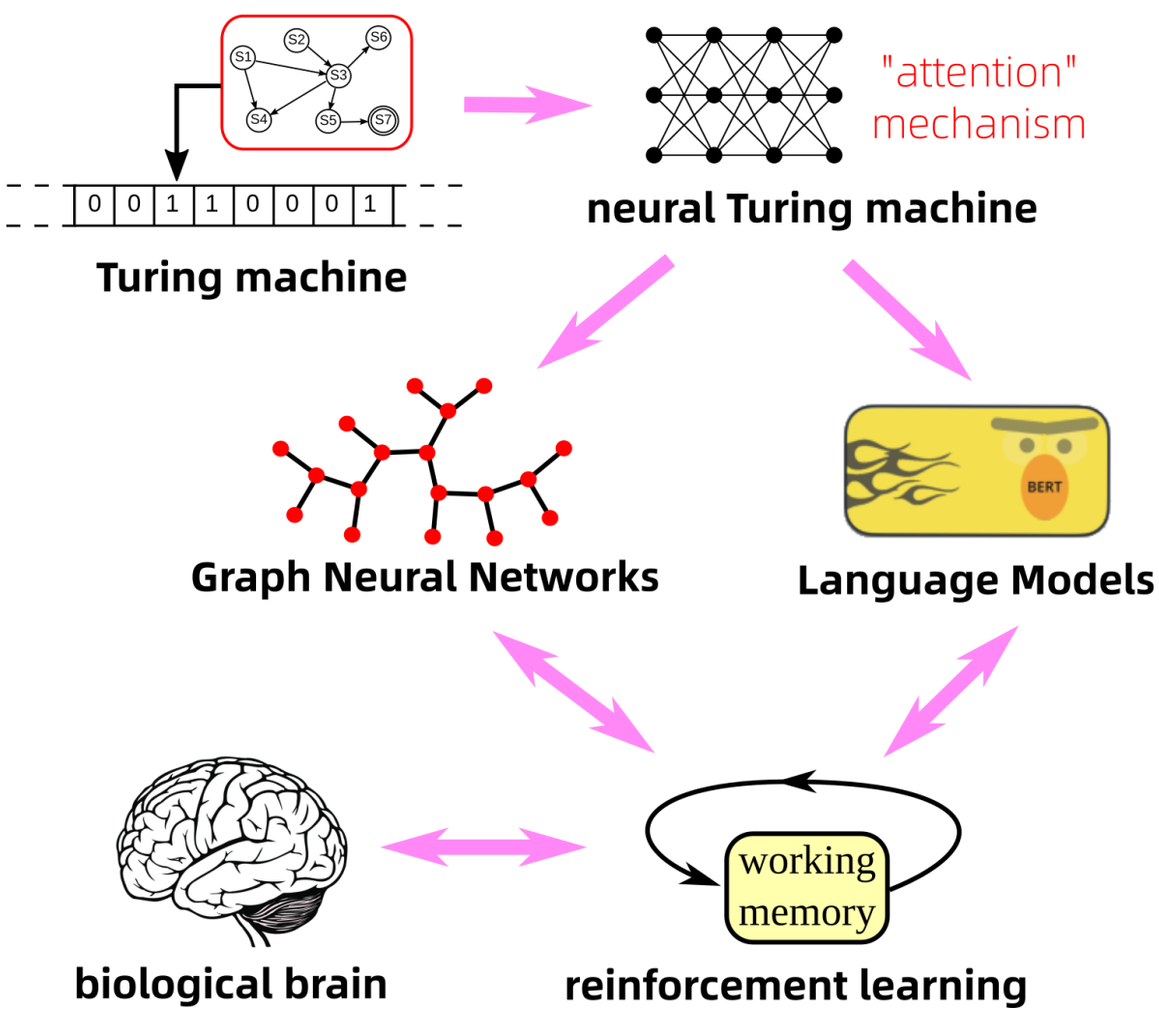
[1] The above diagram is somewhat inaccurate as there are subtle differences between the notions of "state" and Working Memory. In RL, state usually refers to the external environment through observations. WM is slightly different in the sense that an internal belief may be wrong about the environment – eg, mistaking seeing something that doesn't exist. There is on-going research as to the relation between RL and WM. I have not fully resolved this issue.



(1)

Let's look at the interconnections between some "learning machines". Turing Machines inspired Neural Turing Machines, which is the origin of the Attention mechanism. Self-Attention has the symmetry of equivariance, which means that the order of input / output elements is unimportant. Another way to put it is that such elements have a set structure; for example $\{1, 2, 3\}$ and $\{3, 2, 1\}$ are the same set. The set structure is also suitable for handling logic propositions, because $A \wedge B = B \wedge A$.

A graph is also a logical structure, as it can be decomposed into a bunch of nodes and links (relations). For example, John is Pete's friend $\Rightarrow$ friend(John, Pete). Thus the GNN is a logic processor. Similarly, Language Models are built from Transformers / Self-Attention. So we believe that logic-like rules may emerge in Transformer-based language models. Research on "Transformer circuits" partially confirms this.



(4)

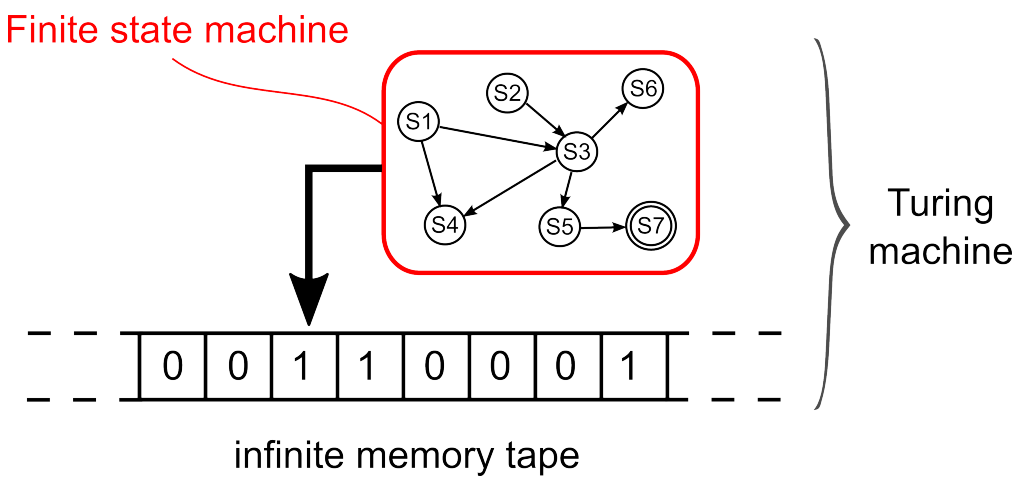
The next, crucial step is to change end-to-end training to the RL setting[1]. The former can be seen as allowing "few-steps" logic inference, whereas the latter allows arbitrary number of inference steps. This enables logic rules to make use of results from other rules, creating a cooperative effect. Precisely due to closed-loop training and the synergy, an RL agent may be able to learn a logic theory that explains the world.

[1] Normally, actions in RL refer to what are performed in the external environment, but in my somewhat unorthodox formulation, actions are "thoughts". This may create some problems not seen in traditional RL.

0 Neural Turing Machines and Transformers

The attention mechanism was first proposed in the "Neural Turing Machine" paper by Graves et al [2014].

Recall that a Turing machine is a Finite State Machine augmented with a Memory Tape:



(5)

In Neural Turing Machines, Graves et al proposed the attention mechanism for an RNN "Controller" (playing the role of the Finite State Machine) to read and write from a Memory Matrix (the tape), using a content-based addressing method.

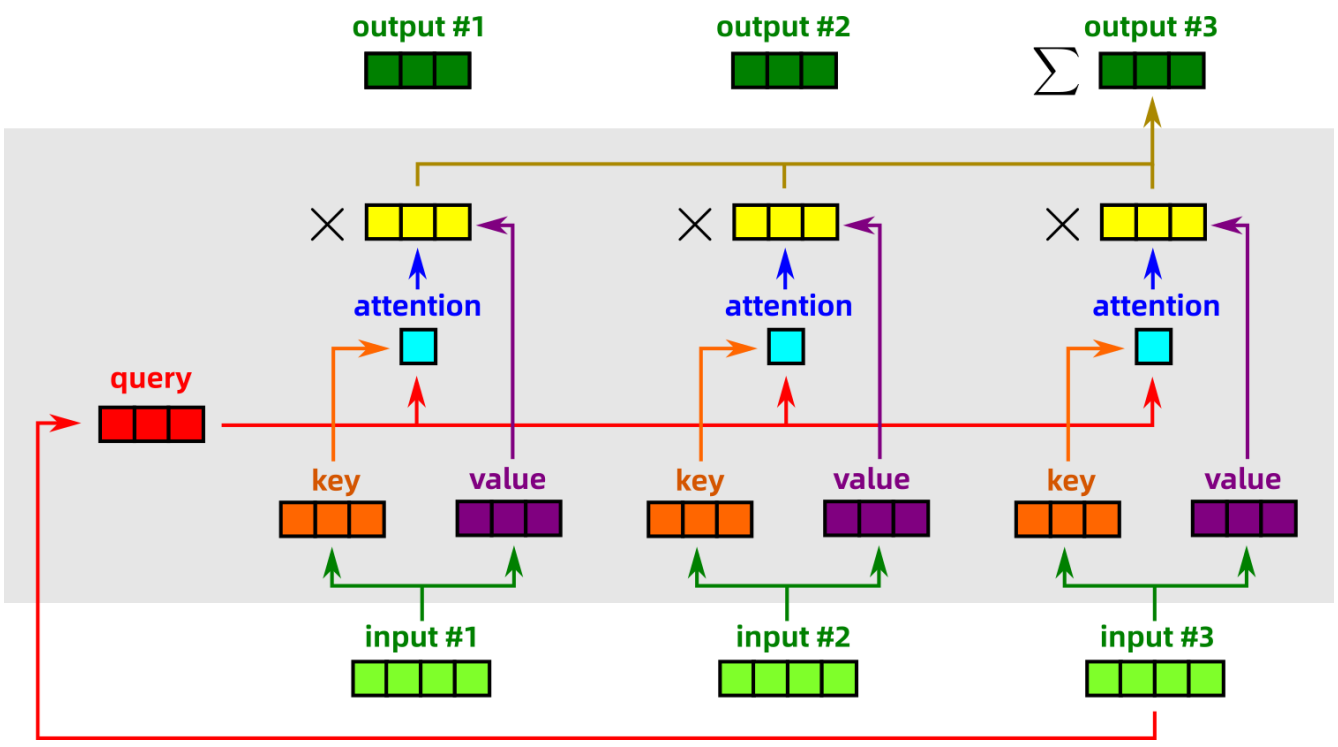
The Memory Matrix M consists of N items, each of constant size. The discreteness of the address would introduce discontinuities in gradients of the output, hence we need an Attention Vector to focus on a specific location in the memory matrix M.

The Attention Vector $\vec{a}$ is calculated via the following formula, familiar to students of the Transformer:

$$\vec{a} = \text{soft}\max_i\{\mathcal{D}(K, M_i)\} \tag{6}$$

where $D()$ is a similarity measure between the key K and memory item M_i. The key K is emitted by the Controller as the value that it is looking for.

This then evolved into the Self Attention mechanism used in all Transformers. Now let us refresh with this diagram illustrating Self-Attention (redrawn from a blog article on the web):



(7)

The research done by Olah et al, in their 2021 paper A Mathematical Framework for Transformer Circuits, is very helpful towards understanding Transformers and Self-Attention.

For example when we say "all men are mortal":

$$\forall x.\ \text{Human}(x) \Rightarrow \text{Mortal}(x) \tag{8}$$

any object instantiated as x (eg. "Socrates") would have to be copied from the LHS to the RHS.

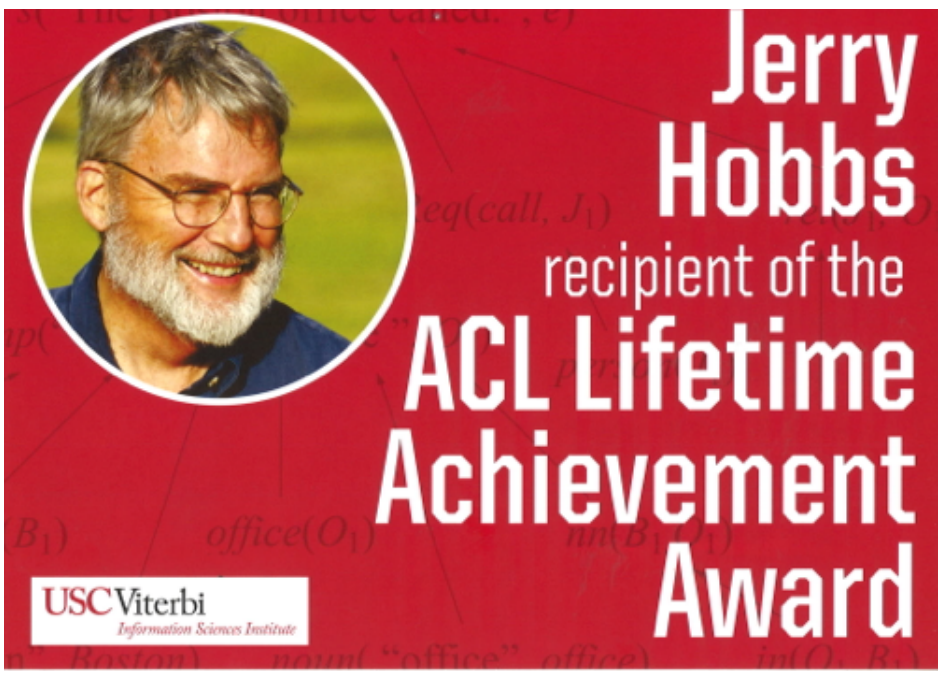
1 Abductive reasoning

Abduction has been relatively neglected in AGI research, which had focused on forward inference. Recently there is a call to study this important aspect.

In logic, abduction means finding the explanation for some known facts. An explanation E is simply some propositions that imply the known fact F, ie, $E \Rightarrow F$.

For example, why do we think a certain actress, say Marilyn Monroe, is "sexy" [1] ? That's because we recognize she has some features (visual or otherwise, no need to enumerate them explicitly) that we consider sexy. So, $E_1 \wedge E_2 \wedge .... \Rightarrow$ Sexy. Those conditions imply she is sexy, and they are the explanation for her sexiness.

Why is abduction important? For example, when a waitress says "The Ham Sandwich left a big tip", Ham Sandwich here refers to the customer who ordered it (an example of metonymy). The AI knows the plain facts such as that someone ordered a ham sandwich, and then it abduces that the most likely interpretation of the phrase "Ham Sandwich" is as the person associated with it. This is the basis of Abductive Interpretation of Natural Language proposed by Jerry Hobbs:



(9)

So abductive reasoning is basically just bidirectional inference.

When a system has both forward and backward connections, it forms a loop and its dynamics is likely to produce "resonance". This harks back to the ART (Adaptive Resonance Theory) proposed by Grossberg and Carpenter beginning in the 1980s.

Such resonance behavior can be viewed as the system seeking to minimize an energy, ie, trying to find the "best explanation" to a set of facts.

This is also corroberated by neuroscientific evidence: areas in the cerebral cortex are replete with both forward- as well as back-projections, as depicted in diagram (**??**). We can further abstract this with the following diagram, where F and G are not functions but optimization constraints:

$$\begin{array}{c} \vec{y} \\ F \circlearrowright G \\ \vec{x} \end{array} \tag{10}$$

If the input $\vec{x}$ produces the output $\vec{y}$ after some iterations, then it is likely that the output $\vec{y}$ would produce $\vec{x}$ in the inverse direction. In other words, we have a neural mechanism that implements a function f and its inverse f^{-1}. The significance of this (from the learning point of view) is that we only need to learn the function f and we get f^{-1} for free.

In logic, if forward inference is denoted as $\vdash_{\text{KB}}$, where KB is a set of logic rules, then abduction is $(\vdash_{\text{KB}})^{-1}$. Abductive interpretation is basically a constraint-satisfaction process that uses inference rules in both directions.

2 Dealing with assumptions