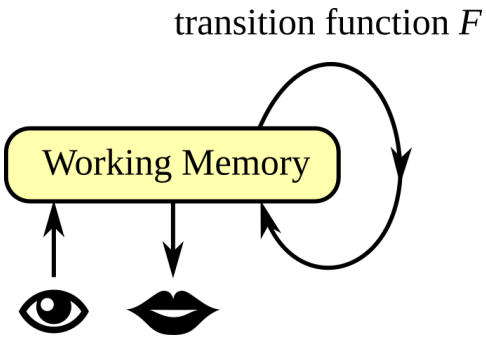
逻辑化 AGI 基础

这是 **强化学习** 最基本的 setup[1]:



(2)

状态转移函数 F 负责 **更新** 工作记忆 (WM). 如果 F 在一个 **闭环** 内训练，它似乎可以 "**解释**" (或 预测) 输入的讯息。

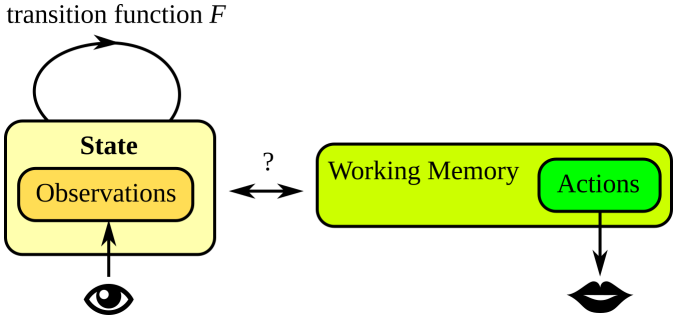
In classical logic-based AI, **inductive learning** means searching for a **theory** T (= set of logic rules) that "explains" or **implies** positive examples but not negative ones:

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

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

我理论的重点是：透过 F 在 RL 闭环内的训练，可以做到 **发现** 逻辑理论 T 的效果。

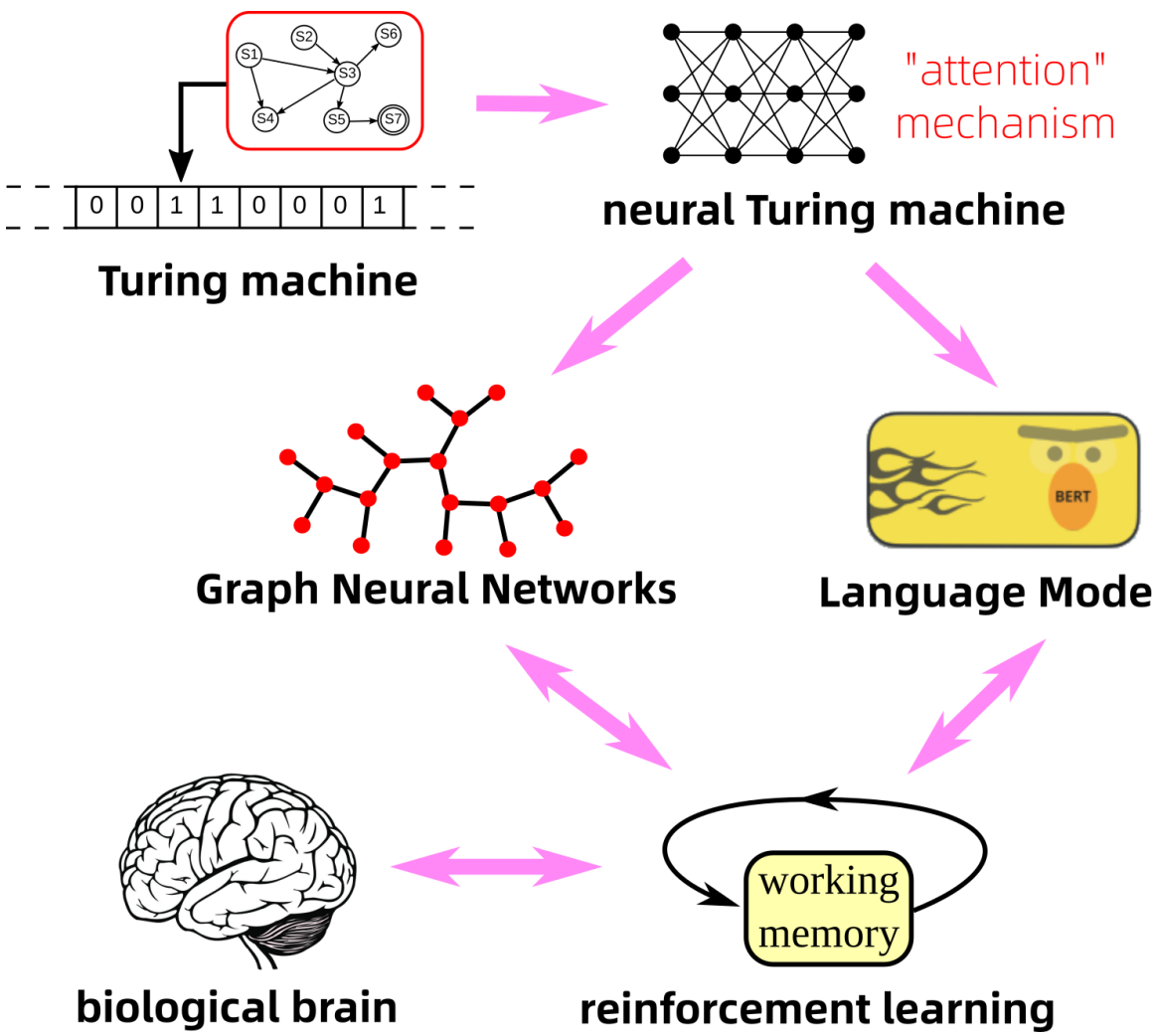
[1]The above diagram is somewhat inaccurate as there are subtle differences between the notions of "state" and Working Memory. In RL, state usually refers to the external environment through observations. WM is slightly different in the sense that an internal belief may be wrong about the environment – 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)

首先可以了解一下几种 “机器” 之间的关系。图灵机 演变成 **神经图灵机**，这是 “**注意力机制**” 的起源。**自注意力** 的特点是它有 equivariance 这种对称性，亦即是说，输入 / 输出元素的**次序**不重要。也可以将这些元素看成一个 **集合** (例如 $\{1,2,3\}$ 跟 $\{3,2,1\}$ 是同一个集合)。这种结构适合处理 **逻辑命题**，因为 $A \wedge B = B \wedge A$.

Graph 也是一种逻辑结构，因为 graph 可以**分解**为一堆节点之间的关系，例如 张三是李四的朋友 $\Rightarrow$ friend(Zhang$_3$, Li$_4$). 所以 GNN 是一种逻辑处理器。另方面，语言模型 也使用 Transformer / Self-Attention. 所以我们推测，在 Transformer 语言模型里 也有逻辑规则的**涌现** (emergence), 而 Transformer circuits 的研究部分地证实了这一想法。



(4)

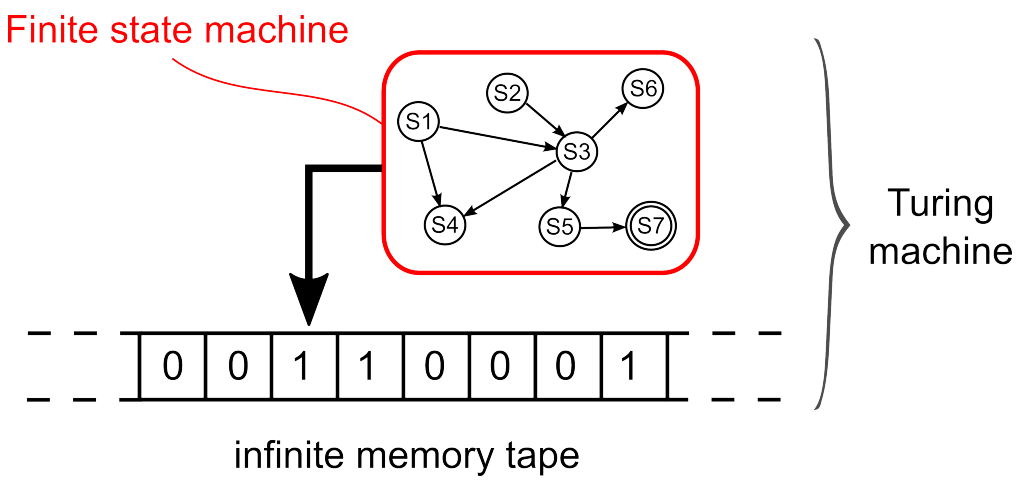
然后关键的一步就是将 end-to-end 训练的模型 变成 RL 训练的形式[1]。前者可以看成是 进行 “few-steps” 的逻辑推导，而后者是可以 “任意多步” 的推导，而且这些推导 互相之间有 **协同效应** (一个推出来的结果帮助另一个新的推导)。正是因为 **闭环训练** 容许了这些 协同效应，令 RL 系统有可能学习出能够**解释**世界的 logic theory.

[1] 通常 RL 的 actions 是在环境中的动作，但我的 RL 模型 要求 actions 是一些 **思想** (thoughts)，这并不规范，可能产生传统 RL 没有的问题。

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.

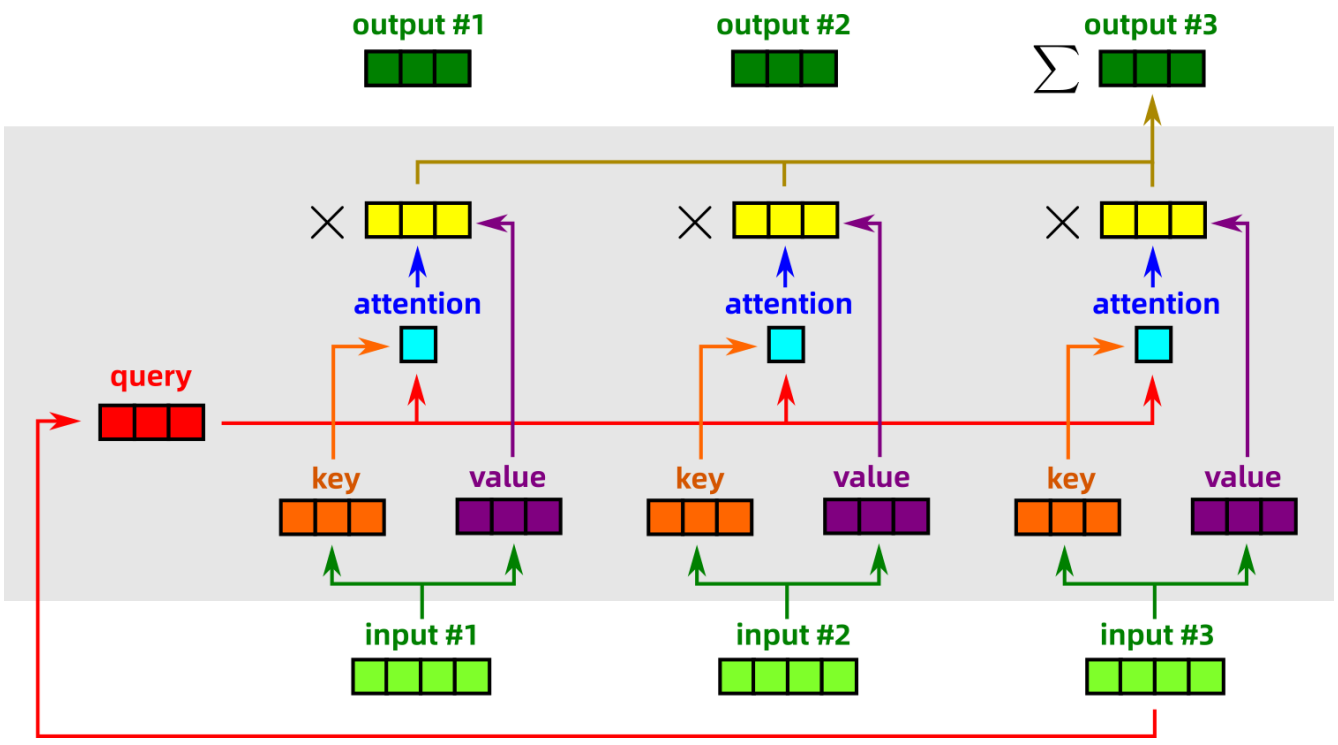
The Memory Matrix M consists of N items, each of constant size. The discreteness of the address would introduce discontinuities in gradients of the output, hence we need an **Attention Vector** to focus on a specific location in the memory matrix M.

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.

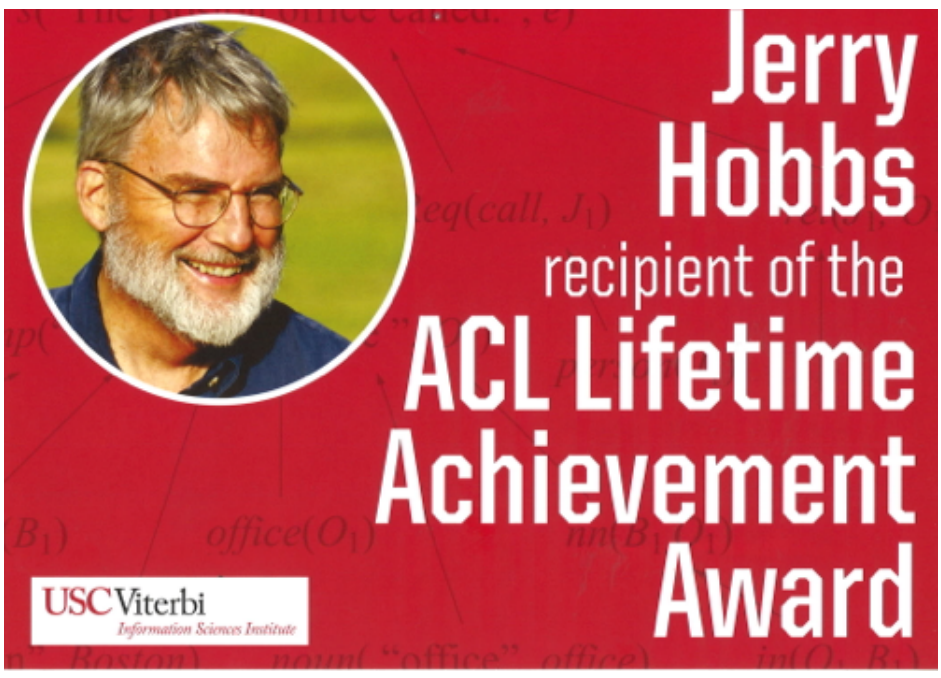
1 Abductive reasoning

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 (11). We can further abstract this with the following diagram, where F and G are not functions but optimization constraints:

$$\begin{array}{ccc} & \vec{y} & \\ F & \circlearrowleft\circlearrowright & G \\ & \vec{x} & \end{array} \tag{10}$$

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