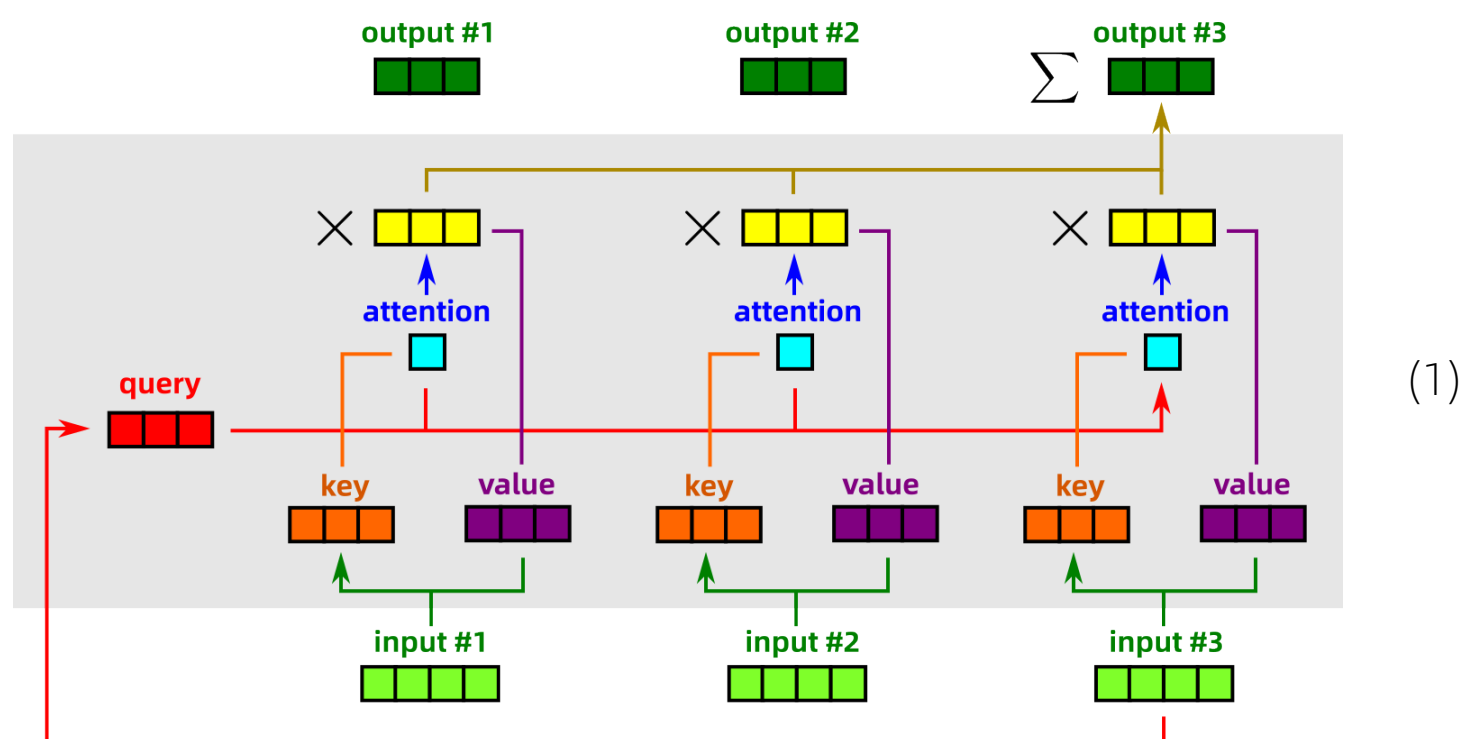


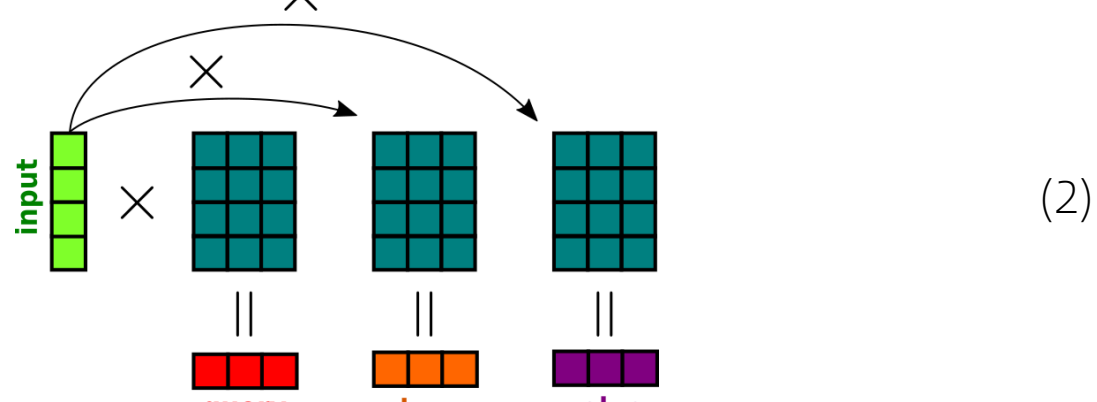
Transformer as Logic-Base

In this infographic I'd explain a major finding that is the culmination of many years of my research: the Transformer is a symbolic-logic machine.

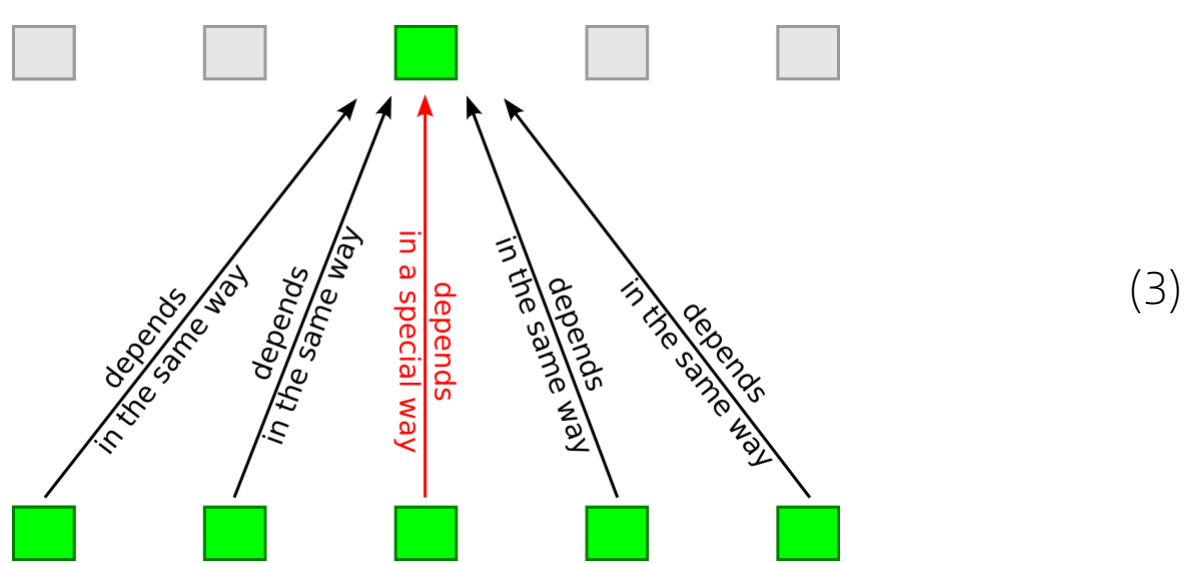
For your convenience here is a refresher on the Transformer's **Self-Attention** mechanism:



"Input" tokens are translated to Q, K, V (query, key, value)'s via matrix multiplication, which can be regarded as a kind of table look-up, or **memory store**:



From an abstract point of view, the Transformer has the following structure, which gives rise to its **equivariance** property (if input elements are swapped in a certain order, the output elements changes the same way):



The equivariance property corresponds to the **exchangeability** of logic propositions:

$$A \wedge B \Leftrightarrow B \wedge A \quad (4)$$

For example:

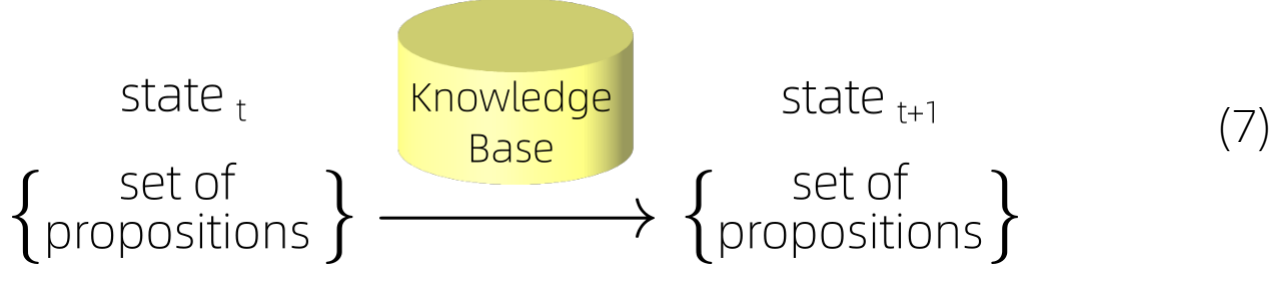
$$\text{it's raining} \wedge \text{I'm heart-broken} \Leftrightarrow \text{I'm heart-broken} \wedge \text{it's raining} \quad (5)$$

Propositions are made up of **atomic concepts**, but here, at the sub-propositional level, atoms cannot be permuted freely, eg:

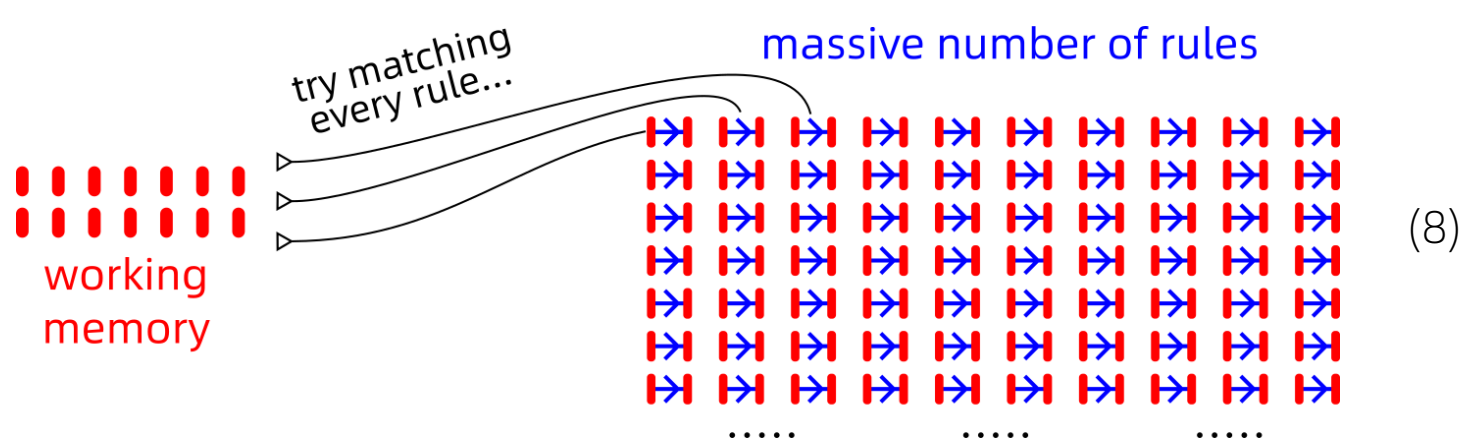
$$\text{I} \cdot \text{love} \cdot \text{her} \neq \text{she} \cdot \text{loves} \cdot \text{me} \quad (6)$$

otherwise there would be no such things as heart-breaks.

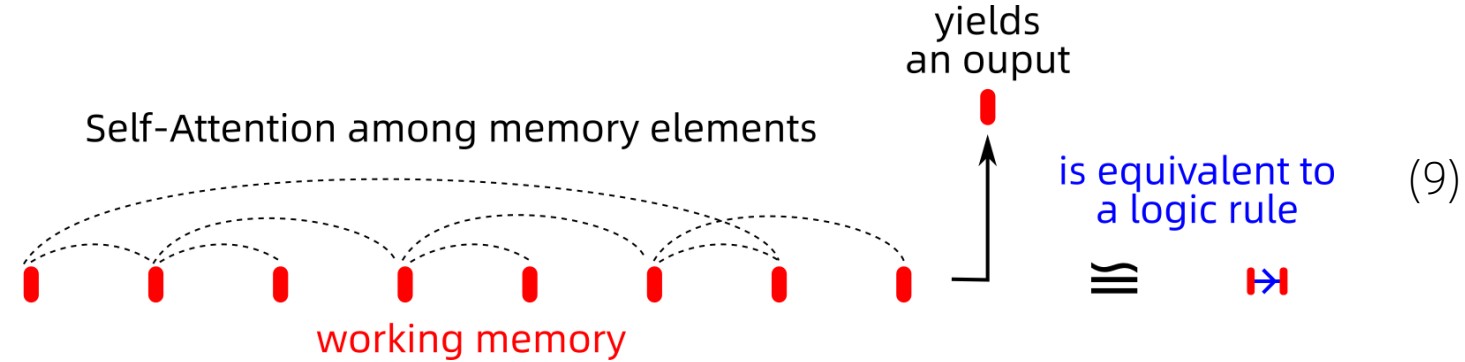
Now let's recall some notions of **classical logic-based AI**. This is its basic architecture:



There would be a huge number of rules in the Knowledge Base, and the system needs to match these rules one by one against propositions in the system's **state** (= working memory):



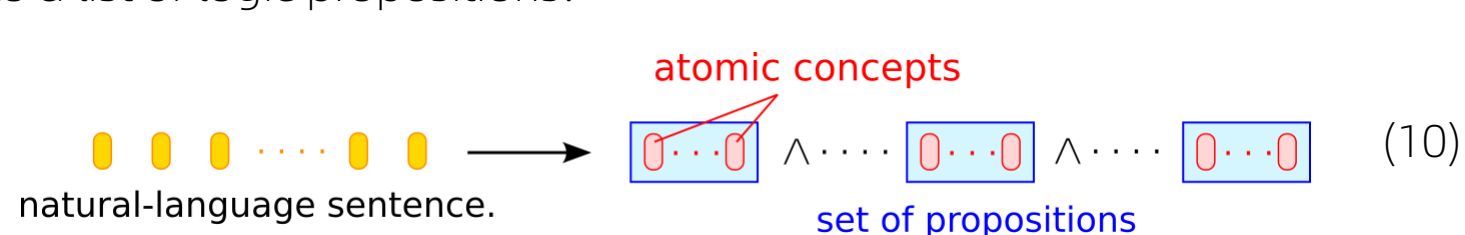
For the Transformer, it is a kind of memory stored **between** input elements (stored as the Q, K, V matrices), and it **implicitly** plays the role of a logic rule-base:



A crucial insight is that the **Self-Attention** mechanism had its origin in NTMs (**Neural Turing Machines**) proposed by Graves *et al* 2014. The Turing machine needs to have a "memory tape" but in the neural setting this memory must be *differentiable*. If the memory is addressed by an index $i \in \mathbb{N}$, then it won't be differentiable. So they came up with a **content-addressable** memory mechanism where a memory matrix is looked up using the "query-key-value" method. A nice explanation of NTMs can be found in the book *Fundamentals of Deep Learning* [Buduma, Locascio 2017].

Now consider LLMs (**Large Language Models**) such as BERT and GPT.

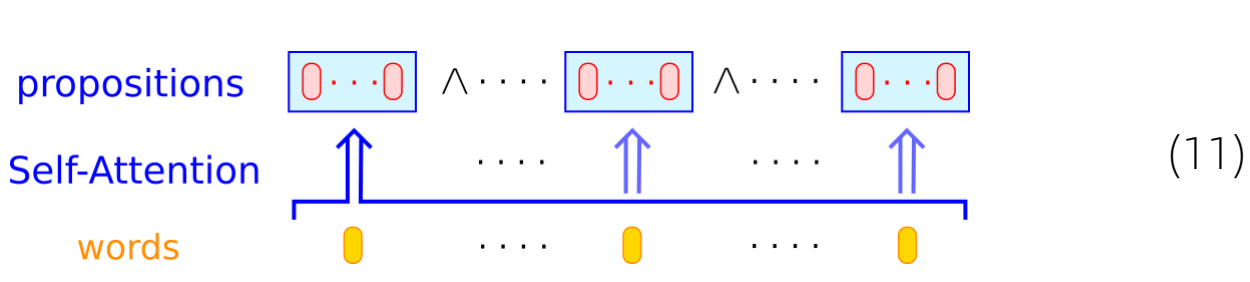
Given a natural-language sentence, we'd like to convert or **decompose** it into a list of logic propositions:



The structure on the right of (10) is the **mental state** of a logical AI system. It is composed of (exchangeable) propositions, which are in turn composed of atomic concepts. This 2-level structure is characteristic of all **symbolic logic** systems.

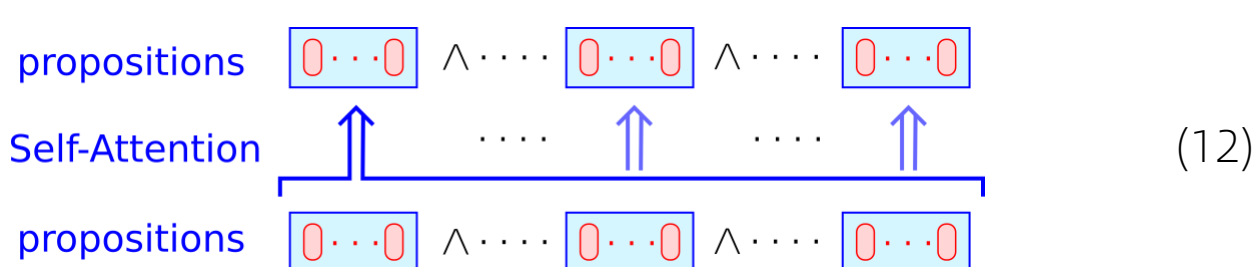
Surprisingly, we found that the Transformer completely satisfies this 2-level logic structure.

On the **first layer**, a Transformer transforms each input word token into one proposition:



The crucial point here is that the propositions are composed of atoms (\square), this is achieved in the Transformer by **adding** vectors (that represent atomic concepts), ie, by **superposition**. Note also that the Transformer is equivariant, so we must add "positional encoding" to each word, to indicate their ordering.

At **higher layers**, there is no need for positional encoding, and logic propositions can be freely exchanged, exactly as what happens in Transformers:



Note that in the above, every \uparrow arrow uses the same (Q, K, V) matrices as "rule-base", that may limit the number of rules that can be represented. To circumvent this, **Multi-Head Attention** allows to use different (Q, K, V) matrices on the same layer.