AI that thinks in its mind

Dimiter Dobrev

Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
d@dobrev.com

If we want to create AGI, we need to give it the ability to understand the world.

The key to understanding is called a world model. This is what AGI should be looking for.

In fact, we will not be looking for a model, but for a description of the world.

To do this, we need a language for description of worlds. In creating this language, we will use the game of chess.

This was done

We have already done this in a previous paper, but then the agent was able to see the chessboard, while now it will play blind.

Playing without seeing the chessboard makes the problem more complex and requires the addition of abstract ED models.

The result will be a world model which will enable AGI think in its mind and plan its actions.

Why world model?

Why are we talking about a world model when everyone is talking about large language models (LLM)? Because understanding gives better results than a statistical approach.

In medicine, we have doctors who treat with drugs and try to understand why the drug cures. We also have people who treat with herbs and completely trust statistics without understanding what is happening and why the herbs help. We do not deny the statistical approach, but understanding gives better results. You can use statistics only if you have accumulated vast experience.

LLMs cannot play chess

For example, LLMs cannot play chess because they cannot understand the position on the chessboard. This should not surprise us. LLMs are unable to even do something as simple as addition of two arbitrary numbers. In other words, thinking without understanding is not proper thinking but only imitation of thinking.

This assertion is supported by many. Read e.g. the opinion of Gary Marcus. Very recently Yann LeCun also endorsed this opinion.

AGI must understand

To play chess, AGI must understand the world. It must understand the state of the world (the position on the chessboard) and the rules of the game. In the world of chess there is also an opponent who plays against the agent. The world model consists of the set of internal states and a function that describes the rules of the game and the opponent.

We already created a model of the chess game in a previous paper. That model was created on the basis of Event-Driven models (ED models). In this paper, we will not deal with the question of what is an event is and accordingly what is an Event-Driven model. Here, we will not provide a precise definition of these terms and instead will only offer a few examples to illustrate the concept.

What is new

In this paper we divide ED models into two types: real and abstract. An example of a real ED model are the days of the week. This is a model with seven states, and the event that triggers the transition from one state to the next one is “midnight”. This is a real model because the states are real and at any given time the world is in one of these states (one of the days of the week).

An example of an abstract ED model are the remainders when dividing by 7 (i.e., the numbers from 0 to 6). Again, we have a model with seven states. The event that switches between them will be “next”. The two models are very similar, but the latter is an abstract one.

In the previous paper we described the chessboard by using real ED models. Here we are going to play blind, so we will have to use abstract models.

Think in mind

In addition to being able to understand, we also need to think in mind. The addition of two arbitrary numbers requires us to execute an algorithm. When executing an algorithm, we do not know how long this execution will take. It may take long or even to run forever. Therefore, thinking in mind must be asynchronous so that it does not block our AGI.

The world model

We can imagine that the set of internal states is $\mathbb{N}$ (because we can encode them in $\mathbb{N}$). We can imagine that the function of the model is from $\mathbb{N}$ to $\mathbb{N}$ (here again we use encoding to add the actions and observations).

In other words, the task of describing the world is reduced to just describing the functions from $\mathbb{N}$ to $\mathbb{N}$. These functions are continuum many and therefore cannot be described precisely. We need to make some compromise and limit our search to a smaller set of functions. Most authors make a major compromise and assume that the function we are looking for is recursive. Our compromise here will not be that big as we will assume that we are looking for a partially recursive function.

The partially recursive functions

Why not stick to the total functions? If our aim is to emulate the world, then the function must be total and even computable in a reasonable time. However, we do not aim to emulate the world – we want only to describe it so that we can plan our future actions. This allows us to use semi-decidable predicates in the description of the world. For example, in the real world we often use the rule: “If there is proof, then it is true.” Although the predicate “there is proof” is semi-decidable, we do not hesitate to use it in the description of the real world.

Is the world totally computable? This is a philosophical question, but even if we assume that it is, we cannot do without semi-decidable predicates. The reason is that we are not looking for the function f that describes the world, but for some function g that returns a good strategy for the agent. When we are about to make a journey, it is perfectly reasonable to ponder whether the journey is finite or infinite, and this is a semi-decidable question.

Non-deterministic model

We will assume that in the world there is randomness as well as other agents whose behavior is not fully predictable. If we were to look for a deterministic world model we would have to understand the world completely and thereby say exactly what is going to happen.

Furthermore, even if the world is deterministic, it is still better to simplify its description by presenting the agents as black boxes. We will represent these black boxes as oracles, and the world model will be a computable function that uses oracles.

Actions and observations

The agent's action will consist of the coordinates of two chess squares:

$$\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle$$

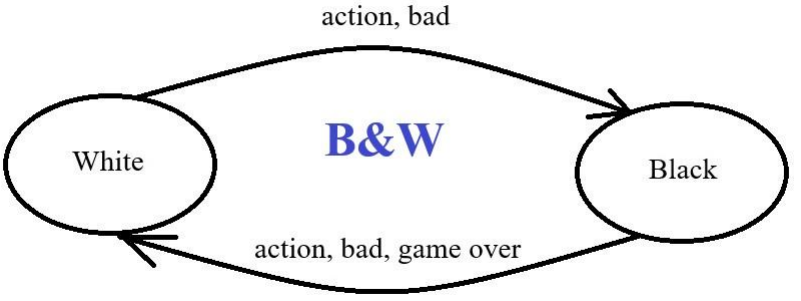
These two squares describe the agent's move. (The piece in $\langle x_1, y_1 \rangle$ is moved to $\langle x_2, y_2 \rangle$.)

The observation will have the following form:

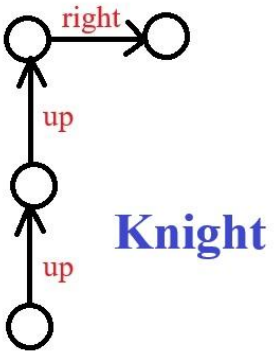
$$\langle x_3, y_3 \rangle, \langle x_4, y_4 \rangle, \mathit{Result}$$

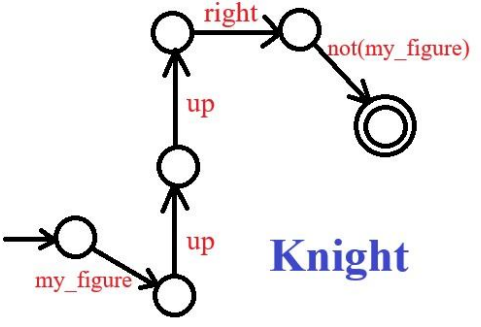
Real ED model

B&W model which tells us which pieces the agent is playing with when the agent is playing alone.

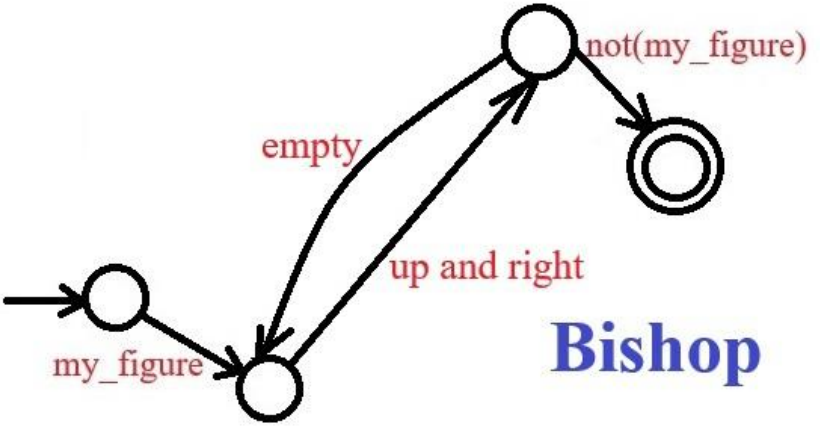


Algorithm in abstract ED model





The algorithm of bishop



Conclusion

This paper tells you how to create AGI, but don't rush to do it right away. AI is not just another step on the way; it is the final step. Before you take that final step and leap into the abyss of the unknown, think carefully about what kind of AI you want to create. Read papers such as "How Can We Make AI with a Nice Character?" in which this issue is discussed.