



II. Solving problems with AI

Interlude on the history of AI: Starting from Search

AI is arguably as old as computer science. Long before we had computers, people thought of the possibility of automatic reasoning and intelligence. As we already mentioned in Chapter 1, one of the great thinkers who considered this question was Alan Turing. In addition to the Turing test, his contributions to AI, and more generally to computer science, include the insight that anything that can be computed (= calculated using either numbers or other symbols) can be automated.

Note

Helping win WWII

Turing designed a very simple device that can compute anything that is computable. His device is known as the Turing machine. While it is a theoretical model that isn't practically useful, it led Turing to the invention of programmable computers: computers that can be used to carry out different tasks depending on what they were programmed to do.

So instead of having to build a different device for each task, we use the same computer for many tasks. This is the idea of programming. Today this invention sounds trivial but in Turing's days it was far from it. Some of the early programmable computers were used during World War II to crack German secret codes, a project where Turing was also personally involved.

The term Artificial Intelligence is often attributed to John McCarthy (1927-2011) – who is often also referred to as the Father of AI – but in fact he denies coming up with the term. He was nevertheless influential in its adoption as the name for the emerging field. The term became established when it was chosen as the topic of a summer seminar, known as the [Dartmouth conference](#), which was organized by McCarthy and held in 1956 at Dartmouth College in New Hampshire. In the proposal to organize the seminar, McCarthy continued with Turing's argument about automated computation. The proposal contains the following crucial statement:

Note

John McCarthy's key statement about AI

“The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it.”

In other words, any element of intelligence can be broken down into small steps so that each of the steps is as such so simple and “mechanical” that it can be written down as a computer program. This statement was, and is still today, a conjecture, which means that we can’t really prove it to be true. Nevertheless, the idea is absolutely fundamental when it comes to the way we think about AI. For example, it shows that McCarthy wanted to bypass any arguments in the spirit of Searle's Chinese Room: intelligence is intelligence even if the system that implements it is just a computer that mechanically follows a program.

Why search and games became central in AI research

As computers developed to the level where it was feasible to experiment with practical AI algorithms in the 1950s, the most distinctive AI problems (besides cracking Nazi codes) were games. Games provided a convenient restricted domain that could be formalized easily. Board games such as checkers, chess, and recently quite prominently Go (an extremely complex strategy board game originating from China at least 2500 years ago), have inspired countless researchers, and continue to do so.

Closely related to games, search and planning techniques were an area where AI lead to great advances in the 1960s: algorithms with names such as the Minimax algorithm or Alpha-Beta Pruning, which were developed then are still the basis for game playing AI, although of course more advanced variants have been proposed over the years. In this Chapter, we will study games and planning problems on a conceptual level.

Next section

III. Search and games



[Course overview](#)[About](#)[FAQ](#)[Privacy Policy](#)[My profile](#)[Sign out](#)

Please help us understand how you study by taking this short questionnaire. All answers are handled anonymously. Thank you!

0 of 8 answered

powered by

[Create your own user feedback survey](#)