Review of important historical developments in the field of AI planning and search
By Brennon York
On 2017/03/19

Write a one-page report on three of these developments, highlighting the relationships between the developments and their impact on the field of AI as a whole.

We begin this report with an analysis of the paper “Programming a Computer for Playing Chess” by Claude E. Shannon [1]. Published in March, 1950 it was one of the first papers describing the concepts of search and constraint propagation to solve a large state space problem thought only solvable by humans. It is now cited over 1200 times according to scholar.google.com and continues to be cited in 2017, nearly 70 years later. We also know now that the game of chess is a solved AI problem with the momentous game between Garry Kasparov and Deep Blue, an artificial intelligence created by IBM.

In the seminal paper Claude describes how one could create a chess playing agent, not necessarily because of a “practical importance,” but rather of “theoretical interest.” He goes on to state that, if this were solved, it could be used in a number of other fields such as “machines for designing relay and switching circuits” or “machines for making strategic decisions in simplified military operations.” He continues on to describe how a chess playing agent could be constructed, the complexity of the state space, and various search strategies to solve the problem.

This paper is likely one of the more important papers in the field of artificial intelligence. Claude, for the first time, applied logic, through planning and search, to discuss the development of a machine that could solve a problem space never before approached. This, and its subsequent work, began larger movements of AI design and spurred many to begin thinking in a new way.

The next paper we look at is “STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving” by Nilsson and Fikes [2] from 1971. This paper discusses the idea of leveraging planning solvers as a basis for robot control. They created a language that acts like situational calculus ensuring each statement consists of a name of its operator and its parameters, its preconditions, and its effects. It only seems to leave out the idea of fluents. STRIPS goes further in attempt to prove its theorems such that these applications could be placed into a robot to solve “world models.”

This paper was a foundational work because of its ability to present the basis of situational calculus and apply both a proof as well as a real world application to “world models.” According to scholar.google.com it has been cited nearly 6000 times and was the basis for planning research.

The last paper we discuss is “PDDL - The Planning Domain Definition Language” by Ghallab et al. This paper was written in 1998 and leverages all of the work done on the STRIPS language over time, creating a machine readable language to “encourage empirical evaluation of planner performance, and development of standard sets of problems all in comparable notations.” They go on to discuss the number of previous languages that went into the design of PDDL, such as ADL, UMCP, and UCPOP, to allow for a vast audience and the ability to share planning constructs between languages. What is described is a formal grammar, written in extended Backus-Naur form (EBNF), that very much resembles the LISP language that most computer scientists studying artificial intelligence in the 60’s and 70’s wrote.

This final paper is important to the artificial intelligence community because it ties all the research and hard work in the search and planning domain together. So much so that the PDDL is the high level language that was used in this very assignment. Ghallab et al. developed a powerful, cross domain, language that nearly all AI researchers in the planning space could use and, thus, share their works to others working in different planning languages.

[1] Programming a Computer for Playing Chess by Claude E. Shannon
[2] STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving by Nilsson and Fikes
[3] PDDL - The Planning Domain Definition Language by Ghallab et al.