For 3 historical developments in the field of AI planning and search, I will present Logic Theorist(LT) from 1956, General Problem Solver (GPS) from 1957 and Soar (State, Operator, And Result) from 1983.

1956 Logic Theorist (LT)

The first demonstration of the Logic Theorist (LT) written by Allen Newell, J.C. Shaw and Herbert A. Simon (Carnegie Institute of Technology, now [[Carnegie Mellon University] or CMU]). This is often called the first AI program, though Samuel's checkers program also has a strong claim.

Logic Theorist is a computer program written in 1955 and 1956 by Allen Newell, Herbert A. Simon and Cliff Shaw. It was the first program deliberately engineered to mimic the problem solving skills of a human being and is among the "the first artificial intelligence program".(Logic theorist is usually considered the first true AI program, although Arthur Samuel's checkers program is earlier. Christopher Strachey also wrote a checkers program in 1951.)

Simon was a political scientist who had already produced classic work in the study of how bureaucracies function as well as developed his theory of bounded rationality (for which he would later win a Nobel Prize). The study of business organizations may seem, on the surface, to be very different from artificial intelligence, but it requires the same insight into the nature of human problem solving and decision making. Simon remembers consulting at RAND in the early 1950s and seeing a printer typing out a map, using ordinary letters and punctuation as symbols. He realized that a machine that could manipulate symbols could just as well simulate decision making and possibly even the process of human thought.

The program that printed the map had been written by Newell, a RAND Corporation scientist studying logistics and organization theory. For Newell, the decisive moment was in 1954 when Oliver Selfridge came to RAND to describe his work on pattern matching. Watching the presentation, Newell suddenly understood how the interaction of simple, programmable units could accomplish complex behavior, including the intelligent behavior of human beings.

Newell and Simon's first project was a program that could prove mathematical theorems like the ones used in Bertrand Russell and Alfred North Whitehead's Principia Mathematica. They took help of computer programmer Cliff Shaw, also from RAND, to develop the program.

Logic Theorist introduced several concepts that would be central to AI research:

Reasoning as search

Logic Theorist explored a search tree: the root was the initial hypothesis, each branch was a deduction based on the rules of logic. Somewhere in the tree was the goal: the proposition the program intended to prove. The pathway along the branches that led to the goal was a proof – a series of statements, each deduced using the rules of logic, that led from the hypothesis to the proposition to be proved.

Heuristics

Newell and Simon realized that the search tree would grow exponentially and that they needed to "trim" some branches, using "rules of thumb" to determine which pathways were unlikely to lead to a solution. They called these ad hoc rules "heuristics", using a term introduced by George Pólya in his classic book on mathematical proof, How to Solve It. Heuristics would become an important area of research in artificial intelligence and remains an important method to overcome the intractable combinatorial explosion of exponentially growing searches.

List processing

To implement Logic Theorist on a computer, the three researchers developed a programming language, IPL, which used the same form of symbolic list processing that would later form the basis of McCarthy's Lisp programming language, an important language still used by AI researchers.

1957 The General Problem Solver (GPS)

General Problem Solver or G.P.S. was a computer program created in 1959 by Herbert A. Simon, J.C. Shaw, and Allen Newell intended to work as a universal problem solver machine. Any problem that can be expressed as a set of well-formed formulas (WFFs) or Horn clauses, and that constitute a directed graph with one or more sources (viz., axioms) and sinks (viz., desired conclusions), can be solved, in principle, by GPS. Proofs in the predicate logic and Euclidean geometry problem spaces are prime examples of the domain the applicability of GPS. It was based on Simon and Newell's theoretical work on logic machines. GPS was the first computer program which separated its knowledge of problems (rules represented as input data) from its strategy of how to solve problems (a generic solver engine). GPS was implemented in the third-order programming language, IPL.

While GPS solved simple problems such as the Towers of Hanoi that could be sufficiently formalized, it could not solve any real-world problems because search was easily lost in the combinatorial explosion. Put another way, the number of "walks" through the inferential digraph became computationally untenable.

The user defined objects and operations that could be done on the objects, and GPS generated heuristics by Means-ends analysis in order to solve problems. It focused on the available operations, finding what inputs were acceptable and what outputs were generated. It then created subgoals to get closer and closer to the goal.

1983 Soar

Soar is a cognitive architecture, originally created by John Laird, Allen Newell, and Paul Rosenbloom at Carnegie Mellon University. It is now maintained and developed by John Laird's research group at the University of Michigan.

The goal of the Soar project is to develop the fixed computational building blocks necessary for general intelligent agents that can perform a wide range of tasks and encode, use, and learn all types of knowledge to realize the full range of cognitive capabilities found in humans, such as decision making, problem solving, planning, and natural language understanding. It is both a theory of what cognition is and a computational implementation of that theory. Since its beginnings in 1983 as John Laird's thesis, it has been widely used by AI researchers to create intelligent agents and cognitive models of different aspects of human behavior.

Soar embodies multiple hypotheses about the computational structures underlying general intelligence, many of which are shared with other cognitive architectures, including ACT-R, which was created by John R. Anderson, and LIDA, which was created by Stan Franklin. Recently, the emphasis on Soar has been on general AI (functionality and efficiency), whereas the emphasis on ACT-R has always been on cognitive modeling (detailed modeling of human cognition).

The original theory of cognition underlying Soar is the Problem Space Hypothesis. The Problem Space Hypothesis contends that all goal-oriented behavior can be cast as search through a space of possible states (a problem space) while attempting to achieve a goal. At each step, a single operator is selected, and then applied to the agent's current state, which can lead to internal changes, such as retrieval of knowledge from long-term memory or modifications or external actions in the world. (Soar's name is derived from this basic cycle of State, Operator, And Result; however, it is no longer regarded as an acronym.) Inherent to the Problem Space Hypothesis is that all behavior, even a complex activity such as planning, is decomposable into a sequence of selection and application of primitive operators, which when mapped onto human behavior take ~50ms.

A second hypothesis of Soar's theory is that although only a single operator can be selected at each step, forcing a serial bottleneck, the processes of selection and application are implemented through parallel rule firings, which provide context-dependent retrieval of procedural knowledge.

A third hypothesis is that if the knowledge to select or apply an operator is incomplete or uncertain, an impasse arises and the architecture automatically creates a substate. In the substate, the same process of problem solving is recursively used, but with the goal to retrieve or discover knowledge so that decision making can continue. This can lead to a stack of substates, where traditional problem methods, such as planning or hierarchical task decomposition, naturally arise. When results created in the substate resolve the impasse, the substate and its associated structures are removed. The overall approach is called Universal Subgoaling.

These assumptions lead to an architecture that supports three levels of processing. At the lowest level, is bottom-up, parallel, and automatic processing. The next level is the deliberative level, where knowledge from the first level is used to propose, select, and apply a single action. These two levels implement fast, skilled behavior, and roughly correspond to Kahneman's System 1 processing level. More complex behavior arises automatically when knowledge is incomplete or uncertain, through a third level of processing using substates, roughly corresponding to System 2.

A fourth hypothesis in Soar is that the underlying structure is modular, but not in terms of task or capability based modules, such as planning or language, but instead as task independent modules including: a decision making module; memory modules (short-term spatial/visual and working memories; long-term procedural, declarative, and episodic memories), learning mechanisms associated with all long-term memories; and perceptual and motor modules. There are further assumptions about the specific properties of these memories described below, including that all learning is online and incremental.

A fifth hypothesis is that memory elements (except those in the spatial/visual memory) are represented as symbolic, relational structures. The hypothesis that a symbolic system is necessary for general intelligence is known as the physical symbol system hypothesis. An important evolution in Soar is that all symbolic structures have associated statistical metadata (such as information on recency and frequency of use, or expected future reward) that influences retrieval, maintenance, and learning of the symbolic structures.

References:

https://en.wikipedia.org/wiki/Timeline_of_artificial_intelligence

https://en.wikipedia.org/wiki/Logic_Theorist

https://en.wikipedia.org/wiki/General_Problem_Solver

http://bitsavers.informatik.uni-stuttgart.de/pdf/rand/ipl/P-1584_Report_On_A_General_Problem-Solving_Program_Feb59.pdf

https://en.wikipedia.org/wiki/Soar_(cognitive_architecture)

http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=12784

http://ai.eecs.umich.edu/soar/sitemaker/docs/misc/GentleIntroduction-2006.pdf

http://www.isi.edu/soar/papers/soar-papers-book/soar-papers.html