Next, there is a selection of key historical developments in the field of AI planning and search that have impacted the field of AI in the following areas:

- **Representation Language:** The representation language -or formal language of the inputs for automated planner- used by STRIPS (Fikes and Nilsson, 1971 [1]) was very influential and is regarded as the most popular representation for restricted state-transitions systems. In fact, it is the base for most of the languages for expressing automated planning problem instances in use today. It was named after the STRIPS planner and is similar to propositional representation, but uses first-order literals instead of propositions. Posteriorly, the Problem Domain Description Language, PDDL (Ghallab et al., 1998 [2]), was introduced as a computer-parsable, standardized syntax for representing planning problems and has been used as the standard language for the International Planning Competition since 1998. The PDDL is a slight relaxation of STRIPS. Similar to STRIPS it describes the initial and goal states as conjunctions of literals, and actions in terms of their preconditions and effects, but literals that are not mentioned, instead of being assumed to be false under the STRIPS closed word assumption, in PDDL they are regarded as unknown.
- **Planning Algorithms:** Classical automated planning algorithms were based on forward chaining state space search, backward chaining and partial-order planning. In the 1990 there was a shift from partial-order planning to GRAPHPLAN and SATPLAN (Weld 1994 [3], Weld 1999). GRAPHPLAN was developed in 1995 by Merrick Furst and Avrim Blum [4] and its name is due to the use of planning graph, where the nodes instead of being the possible states, are sets of actions and atomic facts, arranged into alternative levels. It reduces the amount of search needed to find the solution by pruning incompatible facts that cannot be true at the same time -which corresponds to a level- and incompatible actions that cannot be executed together.
- **Heuristics for Planning:** Heuristics are especially important for forward searching algorithms. Forward search space nodes correspond to individual (grounded) states of the plan state-space and has as larger branching factor than backward searching where space nodes correspond to sets of plan state-space states [5]. For this reason, forward search is only feasible if good heuristics are available. An important milestone was the use of heuristics in A* search algorithms to achieve better performance [6]. A* operates essentially the same as Dijkstra's algorithm except that it uses a heuristic estimate to guide its search towards the most promising states, potentially saving a significant amount of computation.

[1] Richard E. Fikes, Nils J. Nilsson (Winter 1971). "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving" (PDF). Artificial Intelligence. 2 (3–4): 189–208. doi:10.1016/0004-3702(71)90010-5.
[2] McDermott, Drew; Ghallab, Malik; Howe, Adele; Knoblock, Craig; Ram, Ashwin; Veloso, Manuela; Weld, Daniel; Wilkins, David (1998). "PDDL---The Planning Domain Definition Language" (PDF). Technical Report CVC TR98003/DCS TR1165.
[3] Barrett, A., and Weld, D. (1993). Partial-Order Planning: Evaluating Possible Efficiency Gains. University of Washington: Department of Computer Science and Engineering.
[4] A. Blum and M. Furst (1997). Fast planning through planning graph analysis. Artificial intelligence. 90:281-300.
[5] Artificial Intelligence: A Modern Approach by Stuart Russell, Peter Norvig
[6] Hart, P. E.; Nilsson, N. J.; Raphael, B. (1968). "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". IEEE Transactions on Systems Science and Cybernetics SSC4. 4 (2): 100–107. doi:10.1109/TSSC.1968.300136.