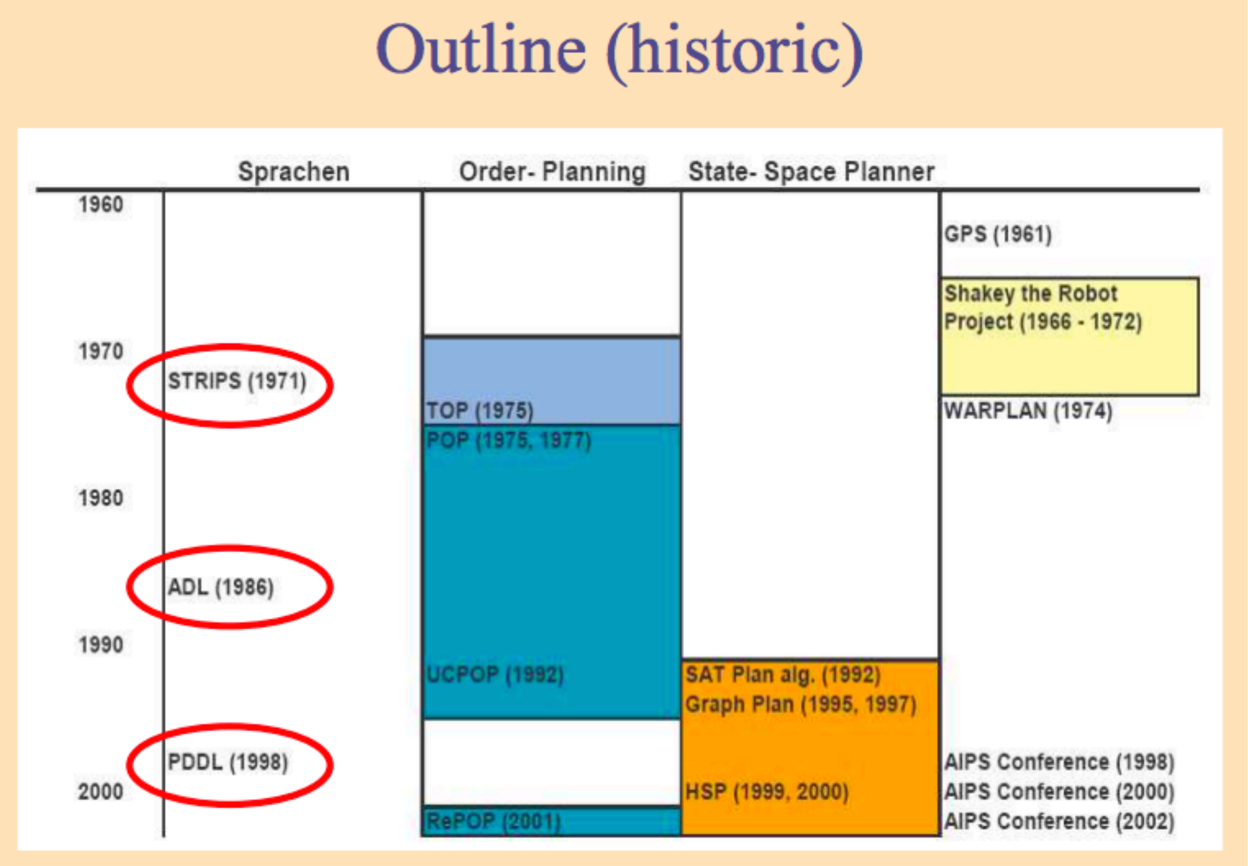
Research Review

The field of AI planning and search attracts many researchers who contributed to myriad of literatures over the past few decades. AI planning “concerns the realization of strategies or action sequences, typically for execution by intelligent agents, autonomous robots and unmanned vehicles.” (“Automated planning and scheduling.” <https://en.wikipedia.org/wiki/Automated_planning_and_scheduling>). And this problem can be solved by search. This report will focus on three of the important historic developments in planning and search, highlight the relationships between the developments and their impact on the field of AI as a whole.

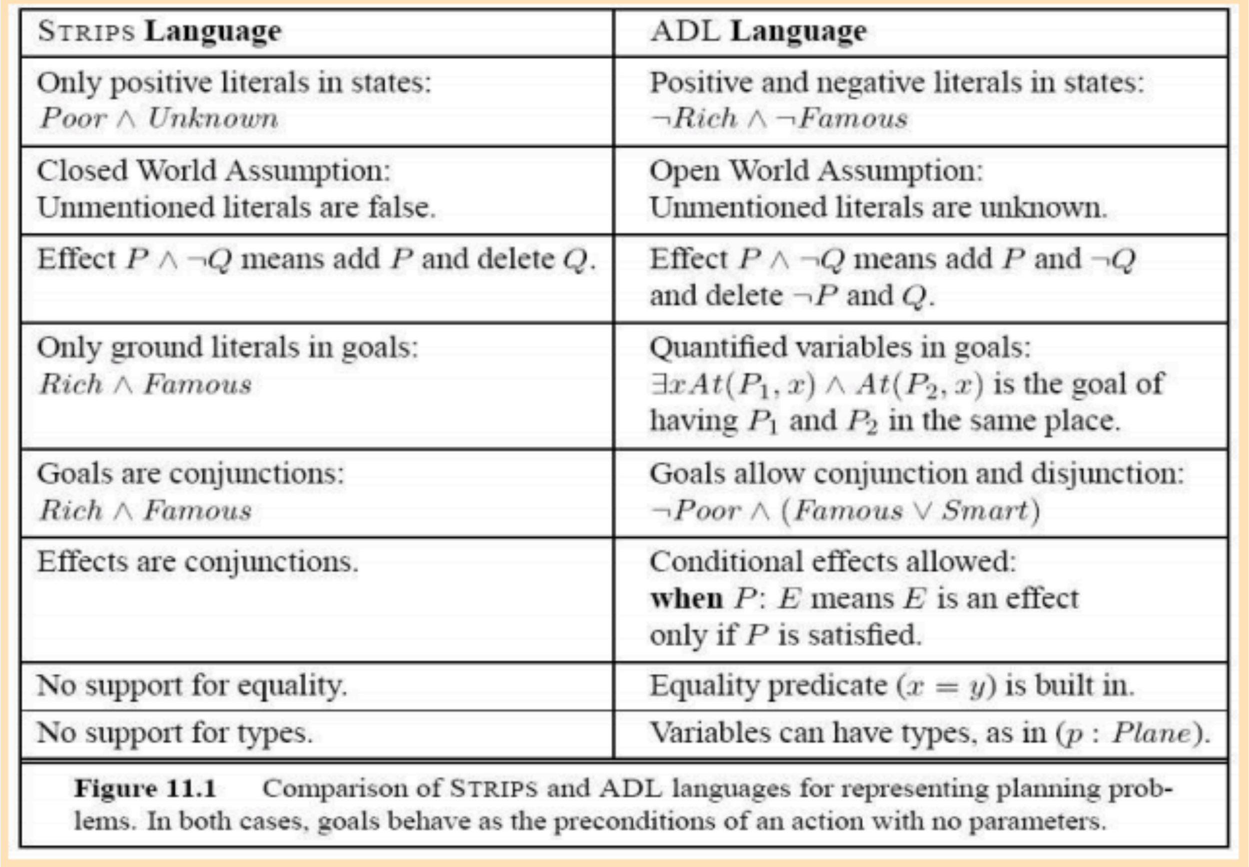


Outline (historic)

(source: http://www.cogsys.wiai.uni-bamberg.de/teaching/ws0405/s_planning/slides/Introduction_AI_Planning_Folien.pdf)

STRIPS (Stanford Research Institute Problem Solver) developed by Richard Fikes and Nils Nilsson in 1971 is the first representation language for planning problems. That is, you first describe the world by providing objects, actions, preconditions and effects. You can then provide a problem set consists of an initial and a goal state. Stated in the abstract of their published paper, “we describe a new problem solver called STRIPS that attempts to find a sequence of operators in a space of world models to transform a given initial world model into a model in which a given goal formula can be proven to be true. STRIPS represents a world model as an arbitrary collection of first-order predicate calculus formulas and is designed to work with models consisting of large numbers of formulas. It employs a resolution theorem prover to answer questions of particular models and uses means-ends analysis to guide it to the desired goal-satisfying model." ("STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving" Richard E. Fikes and Nils J. Nilsson. <http://ai.stanford.edu/~nilsson/OnlinePubs-Nils/PublishedPapers/strips.pdf>). Two different search algorithms are used in this paper, one used to plan the actions and one used to track the current state of the world. The ideas and techniques described in this paper inspired many modern planners.

ADL (Action Description Language), proposed by Pednault in 1986, is a combination of the STRIPS language and situation calculus. ADL relaxes some of STRIPS's assumptions and is therefore considered an advancement of STRIPS. The detailed comparison is below and can also be found here:



STRIPS **Language**	ADL **Language**
Only positive literals in states: $Poor \wedge Unknown$	Positive and negative literals in states: $\neg Rich \wedge \neg Famous$
Closed World Assumption: Unmentioned literals are false.	Open World Assumption: Unmentioned literals are unknown.
Effect $P \wedge \neg Q$ means add P and delete Q.	Effect $P \wedge \neg Q$ means add P and $\neg Q$ and delete $\neg P$ and Q.
Only ground literals in goals: $Rich \wedge Famous$	Quantified variables in goals: $\exists x At(P_1, x) \wedge At(P_2, x)$ is the goal of having P_1 and P_2 in the same place.
Goals are conjunctions: $Rich \wedge Famous$	Goals allow conjunction and disjunction: $\neg Poor \wedge (Famous \vee Smart)$
Effects are conjunctions.	Conditional effects allowed: **when** P: E means E is an effect only if P is satisfied.
No support for equality.	Equality predicate $(x = y)$ is built in.
No support for types.	Variables can have types, as in $(p : Plane)$.

Figure 11.1 Comparison of STRIPS and ADL languages for representing planning problems. In both cases, goals behave as the preconditions of an action with no parameters.

(source: http://www.cogsys.wiai.uni-bamberg.de/teaching/ws0405/s_planning/slides/Introduction_AI_Planning_Folien.pdf)

This paved the road for standardizing planning domain modeling.

PDDL (Problem Domain Description Language), developed by Drew McDermott and his colleagues in 1998, "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.” (“Artificial Intelligence: A Modern Approach”(3rd ed.) Stuart Russel and Peter Norvig. Chapter 10 *Classical Planning*, Page 394)
“The PDDL language provides the foundation on which an expressive standard can be constructed, enabling the domain models of the applications-driven community to be shared and motivating the development of the planning field towards realistic application. “
(“PDDL2.1 : An Extension to PDDL for Expressing Temporal Planning Domains” Maria Fox. <https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume20/fox03a-html/JAIRpddl.html>)
Many versions of PDDL, as well as successors/variants/extensions have been developed since then.