PDDL/ADL, Explicit state-space search & Planning by SAT

The Planning problem in Artificial Intelligence is about the decision making in order to achieve some goals. AI planning arose from investigations into state-space search, theorem proving, and control Theory .It involves choosing a sequence of actions that will transform the state of the world, so that it will satisfy the goal. The world is typically viewed to consist of atomic facts (state variables), and actions make some facts true and some facts false. Rest of the article discusses about development of PDDL/ADL, Explicit state-space search & SAT Planning.

PDDL/ADL : STRIPS ,the first major planning system, is the simplest language used for formalizing actions. In STRIPS, the state variables have the domain {0,1} (equivalently {FALSE, TRUE}), and an action consists of three sets of state variables, the PRECONDITION, the ADD={a1,a2,...,an} list, and the DELETE={d1,d2,...,dm} list. The Action Description Language, or ADL , relaxed some of the STRIPS restrictions and made it possible to encode more realistic problems. Nebel explores schemes for compiling ADL into STRIPS. The Problem Domain Description Language or PDDL was introduced as a computer-parsable, standardized syntax for representing planning problems.

PDDL can be reduced to STRIPS, but there is NO one to one correspondence b/w the PDDL actions and the STRIPS actions, because several STRIPS actions may be needed for one PDDL action, and the set of STRIPS actions may have a size that is exponential in the size of the set of PDDL actions. This is because a PDDL action may need to be reduced to an exponential number of STRIPS actions.

Explicit state-space search : Explicit state-space search, meaning the generation of states reachable from the initial state one by one, is the earliest and most straightforward method for solving some of the most important problems about transition systems, including model-checking , planning, and others, widely used since at least the 1980s. All the current main techniques for it were fully developed by 1990s, including symmetry and partial order methods, informed search algorithms, and optimal search algorithms (A* already in the 1960s). It is relatively easy to implement efficiently, but, when the number of states is high, its applicability is limited by the necessity to do the search only one state at a time. However, when the number of states is less than some tens of millions, this approach is efficient and can give guarantees of finding solutions in a limited amount of time.

Search algorithms
There are several types of search algorithms that are routinely applied in planning. These include the following.

1. well-known uninformed search algorithms like depth-first search, breadth-first search
2. systematic heuristic search algorithms, for example A* and its variants like IDA*, WA*
3. systematic heuristic search algorithms without optimality guarantees, for example the standard "best-first" search algorithm which is like A* but ignores the cost-so-far component
4. incomplete unsystematic search algorithms, most notably stochastic search algorithms

SAT Planning : Many transition systems have too many states to consider them explicitly one by one, and then factored representations that allow representing large numbers of states and state sequences may be a more efficient alternative. Such representations and search methods are known as symbolic or factored. The earliest symbolic search methods were based on Binary Decision Diagrams , pursued in state-space research since late 1980s. The currently most scalable method (since the late 1990s) is based on reduction to the propositional satisfiability problem SAT. The Boolean Satisfiability Problem (abbreviated as SATISFIABILITY or SAT) is the problem of determining if there exists an interpretation that satisfies a given Boolean formula. In other words, it asks whether the variables of a given Boolean formula can be consistently replaced by the values TRUE or FALSE in such a way that the formula evaluates to TRUE. If this is the case, the formula is called satisfiable.

References:

1. Artificial Intelligence: A Modern Approach by Norvig and Russell
2. https://users.ics.aalto.fi/rintanen/planning.html
3. https://en.wikipedia.org/wiki/Boolean_satisfiability_problem
4. D. Mitchell, A SAT Solver Primer, EATCS Bulletin, 85, pp. 112-133, February 2005.
5. J. Rintanen, Planning and SAT, in A. Biere, H. van Maaren, M. Heule and Toby Walsh, Eds., Handbook of Satisfiability, pp. 483-504, IOS Press, 2009.