STRIPs

The Stanford Research Institute Problem Solver (STRIPS) is a model created in 1971 by Fikes and Nilsson that introduces a set of descriptions to describe the planning world in AI. The objects are referred to as well-formed formulas and are grouped into families known as schemata which essentially define a class of similar actions. For example, the action move(A, B) defines an action to move object A to location B; that is schematically equivalent to move(C, D). Actions in STRIPS are pre-instantiated with parameters which make them distinct operators if the parameters are different, as opposed to interpreting them as functions with placeholder variables, which is why this grouping is made.

Operators act on states with descriptions have pre-requisite conditions and effects. Effects are classified into additions and deletions that determine the truth of states after the action has been applied and also describe what actions are possible with a specific set of literals. Goal states are also described as a well-formed formula and are essentially a collection of states. The expansion of actions is determined separately by the choice of algorithms/heuristics. By effectively applying possible actions successfully, STRIPS provides useful terminology that many popular AI planners are based upon.

Graphplan

Graphplan is a planner developed in 1996 by Blum and Furst to tackle problems in Artificial Intelligence. The paper introduces the Planning Graph; a directed graph representing propositions and operators in the context of STRIPS planning domains. Planning Graphs make use of alternating stages of proposition and operator nodes to store additional information that aid the planner. For example, the author makes use of exclusions relations to eliminate actions that have interfering actions or depend on the opposite preconditions. The sequential nature of the graph results in constraints being propagated to the later levels pruning the number of possible actions even further. This structure is a useful illustration of possible operators that can be applied at different time steps and has seen success when applied to toy problems in planning.

Blum and Furst introduced a system that computationally outperformed the state of art at the time (Prodigy, UCPOP and others). The plan is guaranteed to terminate when there are no changes when expanding the layers further, and is proven to determine that a plan does not exist if the problem is unsolvable. The issue of this method is that it can be only applied to STRIPS-like problems, it requires that mutual exclusivity identify important constraints or that actions significantly reduce the depth of the search, and that it attempts to find the shortest solution which may take a long time to compute. Graphplan is a flexible system that can be customised when choosing how to expand layers or select actions, and can be augmented with heuristics to solve deterministic problems in AI.

Cplan

Constraint Programming (CPlan) is a methodology described by Beek and Chen in 1999 to solve large combinatorial problems. The method at the time proposed a different approach to the prevalence of using domain knowledge and general-purpose search algorithms to solve planning problems. CPlan is independent of the search algorithm used and is simply used to represent the domain. This system is formulated as set of variables each belonging to a domain, and a collection of constraints (known as the scheme) relating to these variables. Typically used with backtracking, constraints are used to reduce the size of the domain for each unassigned variable, benefitting from the propagation of arc consistency when back jumping. The paper defines a number of constraint types which are used to guide the assignment of unassigned variables in a problem.

Formulating planning problems as constraint satisfaction can often yield a more condensed representation by assigning values to variables. This is in contrast to STRIPs of having multiple predicates to represent the truth of a variable having a specific value and the falsity of that variable having other values. When competing against implementations of planning graphs such as Blackbox and IPP, CPlan outperformed these planners in finding solutions to the gripper and logistics planning problems more quickly. Constraint programming expands on the concept of mutual exclusivity in planning graphs to introduce new ways to restrict the search space even further, which is why backtracking is a viable algorithm to be used with this method.