AI planning and search

1、 UCPOP

UCPOP is a Partial Order Planner whose step descriptions include Conditional effects and Universal quantification. Both universal and existential quantification are permitted in the step reconditions, effect preconditions, effect post conditions, and goals. Although UCPOP assumes a closed world (actions cannot add or delete from the fixed universe of objects) and does not allow domain axioms or disjunctive preconditions, it is considerably more expressive than other rigorous partial order planners.

The UCPOP algorithm starts with an initial, dummy plan that consists solely of a “start" step (whose effects encode the initial conditions) and a ”goal" step (whose preconditions encode the goals). UCPOP then attempts to complete this initial plan by adding new steps and constraints until all preconditions are guaranteed to be satisfied. The main loop makes two types of choices: supporting “open" preconditions and resolving “threats”.

2、 GRAPHPLAN

Blum and Furst’s (1997, 1995) GRAPHPLAN algorithm is one of the most exciting recent developments in AI planning for two reasons: First, GRAPHPLAN is a simple, elegant algorithm that yields an extremely speedy planner—in many cases, orders of magnitude faster than previous systems such as SNLP (McAllester and Rosenblitt 1991), PRODIGY (Minton et al. 1989), or UCPOP (Penberthy and Weld 1992). Second, the representations used by GRAPHPLAN form the basis of the most successful encodings of planning problems into propositional SAT; hence, familiarity with GRAPHPLAN aids in understanding SAT-based planning systems. GRAPHPLAN alternates between two phases: (1) graph expansion and (2) solution extraction. The graph-expansion phase extends a planning graph forward in “time” until it has achieved a necessary (but possibly insufficient) condition for plan existence. The solution-extraction phase then performs a backward-chaining search on the graph, looking for a plan that solves the problem; if no solution is found, the cycle repeats by further expanding the planning graph.

3、 SATPLAN

Despite the early formulation of planning as theorem proving (Green 1969), most researchers have long assumed that special-purpose planning algorithms are necessary for practical performance. Algorithms such as TWEAK, SNLP, UCPOP, and GRAPHPLAN can all be viewed as special-purpose theorem provers aimed at planning problems. However, recent improvements in the performance of propositional satisfiability methods (Cook and Mitchell 1997) call this whole endeavor in doubt. Initial results for compiling bounded length planning problems to SAT were unremarkable (Kautz and Selman 1992), but recent experiments (Kautz and Selman 1996) suggest that compilation to SAT might yield the world’s fastest STRIPS-style planner.

The compiler takes a planning problem as input, guesses a plan length, and generates a propositional logic formula, which, if satisfied, implies the existence of a solution plan; a symbol table records the correspondence between propositional variables and the planning instance. The simplifier uses fast (linear time) techniques such as unit clause propagation and pure literal elimination (for example, Van Gelder and Tsuji [1996]) to shrink the CNF formula. The solver uses systematic or stochastic methods to find a satisfying assignment that the decoder translates (using the symbol table) into a solution plan. If the solver finds that the formula is unsatisfiable, then the compiler generates a new encoding reflecting a longer plan length.

REFEREBCES

Penberthy J S, Weld D S. UCPOP: A Sound, Complete, Partial Order Planner for ADL[J]. Kr, 1992, 92: 103-114.

Weld D S. Recent advances in AI planning[J]. AI magazine, 1999, 20(2): 93.