Research Review : Planning

By Anshuman Sahoo

Planning is an idea where you are given a description of the starting state, the goal state and a set of possible actions that an agent can take. And in this framework, problem solving involved ***searching*** for a sequence of actions to go from the start state to the goal state.

Most of the work on planning is based on a state transition model where the system consists of a number of states, and transitions that change one state to another. The most convenient representation of this idea is in the form of graphs.

The simplest language for formalizing actions is the **STRIPS** language in which state variables have a binary domain and three sets of state variables, namely precondition, add list and delete list. This representation proved very influential in shaping modern programmable logic syntax for planning.

Petri nets are a model of state transition systems where transitions occur independently. In some ways, STRIPS models can be represented by restricted Petri nets. In Petri nets, the state variables are called places. Each place can hold 0 or more tokens, ie, the place is a state variable with a natural numbers domain. Transitions are described as predecessor places and a set of successor places.

PDDL/ADL is a generalization of STRIPS. The differences are mainly that the precondition can be an arbitrary boolean combination of atomic facts about the state variables. Unlike the add and delete lists representing unconditional assignments, statements in PDDL/ADL may be conditional (IF/THEN).

Explicit state space search became the most predominant approach to planning in the 1980s-90s. These include uninformed search algorithms like breadth-first and depth search along with systematic heuristic algorithms with optimality guarantees like A* with variants like IDA*, WA*. Others like best first dont have optimality guarantees. However, use of these methods are highly dependent on the size of the search space. Incomplete unsystematic search algorithms like stochastic search algorithms are useful in certain situations as well.

Symmetry reduction methods were developed to decrease the size of search space by recognizing symmetries in the state space graph. **Partial order reduction** methods try to decrease number of search steps by recognizing independence of actions/transitions. **Heuristics** usually involve different forms of relaxation of the problem.

Sources:

1) https://users.ics.aalto.fi/rintanen/planning.html
2) B. Bonet and H. Geffner, Planning as Heuristic Search, Artificial Intelligence Journal, 129(1-2), pp. 5-33, 2001.